

RK312X
Technical Reference Manual

Rockchip Confidential

Revision 0.1
JUN 2014

Revision History

Date	Revision	Description
2014-6-30	0.1	Initial Release

Rockchip Confidential

Table of Content

Table of Content	3
Figure Index	22
Table Index	27
Chapter 1 Introduction.....	30
1.1 Features	30
1.1.1 MicroProcessor	30
1.1.2 Memory Organization.....	30
1.1.3 Internal Memory	30
1.1.4 External Memory or Storage device	31
1.1.5 System Component	32
1.1.6 Video CODEC.....	34
1.1.7 JPEG CODEC	35
1.1.8 Image Enhancement.....	35
1.1.9 Image Enhancement(IEP module)	36
1.1.10 Graphics Engine	37
1.1.11 Video IN/OUT	38
1.1.12 HDMI.....	39
1.1.13 LVDS	39
1.1.14 MIPI DPHY.....	39
1.1.15 Audio Interface	40
1.1.16 Connectivity	40
1.1.17 Others	43
1.2 Block Diagram	44
1.3 Pin Description	45
1.3.1 RK3128 power/ground IO descriptions	45
1.3.2 RK3128 function IO descriptions.....	48
1.3.3 IO pin name descriptions.....	59
1.3.4 RK3128 IO Type.....	64
1.4 Package information	65
1.4.1 LQFP176 Dimension	66
1.4.2 TFBGA313 Dimension	67
1.4.3 LQFP176 Ball Map	70
1.4.4 TFBGA313 Ball Map	72
1.4.5 LQFP176 Ball Pin Number Order(Tablet)	74
1.4.6 RK3126 function IO package descriptions	77
1.5 Electrical Specification	78
1.5.1 Absolute Maximum Ratings.....	78
1.5.2 Recommended Operating Conditions	78
1.5.3 DC Characteristics	79
1.5.4 Recommended Operating Frequency	83
1.5.5 Electrical Characteristics for General IO	86
1.5.6 Electrical Characteristics for PLL	87
1.5.7 Electrical Characteristics for SAR-ADC	87
1.5.8 Electrical Characteristics for USB OTG/Host2.0 Interface	88
1.5.9 Electrical Characteristics for HDMI	89
1.5.10 Electrical Characteristics for DDR IO.....	89
1.5.11 Electrical Characteristics for eFuse	89
1.5.12 Electrical Characteristics for MIPI.....	90

1.5.13 Electrical Characteristics for LVDS	90
1.5.14 Electrical Characteristics for VDAC	90
1.6 Hardware Guideline	91
1.6.1 Reference design for RK3128 oscillator PCB connection.....	91
1.6.2 Reference design for PLL PCB connection	92
1.6.3 Reference design for USB OTG/Host2.0 connection.....	92
1.6.4 Reference design for HDMI Tx PHY connection.....	94
1.6.5 Reference design for Audio Codec connection	95
1.6.6 Reference design for GPS connection	96
1.6.7 RK3128 Power on reset descriptions	96
Chapter 2 System Overview.....	97
2.1 Address Mapping.....	97
2.2 System Boot.....	97
2.3 System Interrupt connection.....	99
2.4 System DMA hardware request connection.....	102
Chapter 3 Clock and reset unit	103
3.1 Overview	103
3.2 Block Diagram	103
3.3 System Clock Solution	103
3.4 System Reset Solution	108
3.5 Function Description	108
3.6 PLL Introduction	109
3.6.1 Overview	109
3.6.2 Block diagram	109
3.7 Register Description	110
3.7.1 Registers Summary	110
3.7.2 Detail Register Description	113
3.8 Timing Diagram	161
3.9 Application Notes	161
3.9.1 PLL usage	161
3.9.2 PLL frequency change and lock check.....	163
3.9.3 Fractional divider usage	163
3.9.4 Global software reset.....	164
Chapter 4 System Debug.....	165
4.1 Overview	165
4.1.1 Features	165
4.1.2 Debug components address map	165
4.2 Block Diagram	165
4.3 Function description	165
4.3.1 DAP	165

4.4 Register description	166
4.4.1 DAP APB-AP register summary	166
4.4.2 DAP APB-AP Detailed Register Description	166
4.4.3 DAP-ROM register summary	169
4.4.4 DAP-ROM Detailed Register Description	169
4.5 Interface description.....	171
4.5.1 DAP SWJ-DP interface.....	171
4.5.2 DAP SW-DP interface	172
Chapter 5 General register file(GRF)	173
5.1 Overview	173
5.1.1 Features	173
5.2 GRF Register Description	173
5.2.1 Register Summary	173
5.2.2 Detailed Register Description	176
Chapter 6 Embedded Processor (Cortex-A7).....	251
6.1 Overview	251
6.2 Block Diagram	252
6.3 Function description	252
6.4 Register description.....	252
Chapter 7 Interconnect	253
7.1 Overview	253
7.2 Axi interconnect.....	253
7.2.1 Integration description.....	253
7.2.2 Function description	255
7.2.3 Memory Scheduler	256
7.2.4 Register Description(main interconnect)	257
7.3 Peri AXI interconnect	263
7.3.1 Integration description.....	263
7.3.2 Function description	264
7.3.3 Register description	264
7.4 Application Notes	267
7.4.1 Idle request to flush the transaction flow	267
7.4.2 QoS and more	267
Chapter 8 DMA Controller (DMAC)	270
8.1 Overview	270
8.2 Block Diagram	270
8.3 Function Description	271
8.3.1 Introduction	271
8.3.2 Operating states	272
8.4 Register Description	272
8.4.1 Register summary	272

8.4.2 Detail Register Description	275
8.5 Timing Diagram	288
8.6 Interface Description	289
8.7 Application Notes	290
8.7.1 Using the APB slave interfaces	290
8.7.2 Security usage.....	290
8.7.3 Programming restrictions	294
8.7.4 Unaligned transfers may be corrupted.....	295
8.7.5 Interrupt shares between channel.	295
8.7.6 Instruction sets.....	296
8.7.7 Assembler directives.....	297
8.7.8 MFIFO usage	297
Chapter 9 Generic Interrupt Controller (GIC).....	298
9.1 Overview	298
9.2 Block Diagram	298
9.3 Function Description	298
9.3.1 The Distributor	299
9.3.2 CPU interface	300
9.3.3 Interrupt handling state machine.....	300
9.4 Register Description	302
9.4.1 GIC Distributor interface register summary	302
9.4.2 GIC Distributor interface detail register description	303
9.4.3 GIC CPU interface register summary	308
9.4.4 GIC CPU interface detail register description	309
9.5 Interface Description	312
9.6 Application Notes	312
9.6.1 General handling of interrupts.....	312
9.6.2 Interrupt prioritization	312
9.6.3 The effect of the Security Extensions on interrupt handling ..	314
Chapter 10 DMC (Dynamic Memory Interface).....	316
10.1 Overview	316
10.2 Block Diagram	317
10.3 Function Description	318
10.3.1 Protocol controller(PCTL)	318
10.3.2 DDR PHY.....	319
10.4 Register Description.....	320
10.4.1 Registers Summary	320
10.4.2 Detail Registers Description	326
10.5 Interface Description	389
10.6 Application Notes	390
10.6.1 State transition of PCTL.....	390
10.6.2 Initialization	392

10.6.3 Low Power Operation	393
10.6.4 TX DLLs	395
10.6.5 RX DLLs	396
10.6.6 High Speed IO Drive Strength	396
10.6.7 PHY Low Speed Mode (200MHz)	397
10.6.8 Per bit de-skew tuning	398
10.6.9 DDR PHY Calibration	398
Chapter 11 Crypto.....	400
11.1 Overview	400
11.2 Block Diagram	400
11.3 Register description	401
11.3.1 Register Summary.....	401
11.3.2 Detail Register Description.....	403
11.4 Application Note	422
11.4.1 Reset a port	422
11.4.2 Overall Performance	422
11.4.3 Usage	422
Chapter 12 NandC(Nand Flash Controller).....	424
12.1 Overview	424
12.2 Block Diagram	425
12.3 Function Description	426
12.3.1 AHB Interface	426
12.3.2 Flash Type/Flash Interface	426
12.3.3 Linked List Pointer Mode (LLP).....	426
12.3.4 BCH Encoder/BCH Decoder	426
12.3.5 Randomizer	426
12.3.6 Delay Line Controller	426
12.4 Register Description.....	427
12.4.1 Internal Address Mapping	427
12.4.2 Registers Summary	427
12.4.3 Detail Register Description.....	439
12.5 Interface Description	538
12.6 Application Notes	540
12.6.1 Clock Description.....	540
12.6.2 BCHST/BCHLOC/BCHDE/SPARE Application	540
12.6.3 Bus Mode Application	543
12.6.4 BchPage Application.....	545
12.6.5 PageSize/SpareSize Application	545
12.6.6 Randomizer Application	546
12.6.7 DLL Application	546
12.6.8 NandC Interrupt Application.....	547
12.6.9 LLP Application.....	547
Chapter 13 Mobile Storage Host Controller.....	550
13.1 Overview	550
13.2 Block Diagram	551

13.3 Function Description	552
13.3.1 Bus Interface Unit	552
13.3.2 Card Interface Unit	556
13.4 Register Description.....	569
13.4.1 Register Summary.....	569
13.4.2 Detail Register Description.....	571
13.5 Interface Description	595
13.5.1 SDMMC IOMUX	595
13.5.2 SDIO IOMUX.....	595
13.5.3 EMMC IOMUX.....	596
13.6 Application Notes	596
13.6.1 Card-Detect and Write-Protect Mechanism	596
13.6.2 SD/MMC Termination Requirement.....	597
13.6.3 Software/Hardware Restriction	598
13.6.4 Programming Sequence.....	600
13.6.5 Voltage Switching	611
13.6.6 Back-End Power	616
13.6.7 DDR Operation.....	616
13.6.8 H/W Reset Operation	618
13.6.9 FBE Scenarios.....	619
13.6.10 Variable Delay/Clock Generation	620
13.6.11 Variable Delay Tuning.....	622
13.6.12 Package Command	623
13.6.13 Card Detection Method	624
13.6.14 SDMMC IOMUX With JTAG.....	625
Chapter 14 Power Management Unit (PMU)	626
14.1 Overview	626
14.1.1 Features	626
14.2 Block Diagram	626
14.2.1 power domain partition	626
14.2.2 PMU block diagram	627
14.3 Power Switch Timing Requirement	627
14.4 Function Description	627
14.4.1 Normal Mode	627
14.4.2 Low Power Mode	628
14.4.3 Wakeup source	628
14.5 Register Description.....	629
14.5.1 Register Summary.....	629
14.5.2 Detail Register Description.....	629
14.6 Timing Diagram	639
14.6.1 Each domain power switch timing	639
14.6.2 External wakeup PAD timing	639
14.7 Application Notes	640
14.7.1 Recommend configurations for power mode.	640
14.7.2 System Register.....	641

14.7.3 Configuration Constraint.....	641
Chapter 15 Embedded SRAM	642
15.1 Overview	642
15.1.1 Features supported.....	642
15.1.2 Features not supported	642
15.2 Block Diagram	642
15.3 Function Description	642
15.3.1 AXI slave interface	642
15.3.2 Embedded SRAM access path.....	642
15.3.3 Remap.....	643
Chapter 16 GPU (Graphics Process Unit)	644
16.1 Overview	644
16.2 Block Diagram	644
16.3 Function description.....	646
16.4 Register description	646
16.5 Timing Diagram	646
16.6 Interface description.....	647
16.7 Application Notes	647
Chapter 17 VPU_Combo	649
17.1 Overview	649
17.1.1 Features	649
17.2 Block Diagram	650
17.3 Function Description	651
17.3.1 HEVC Standard	651
17.3.2 HEVC Coding Tools	652
17.3.3 MMU.....	654
17.3.4 HEVC Working Mode	655
17.3.5 H.264 decoder	655
17.4 Register description	657
17.4.1 VDP_HEVC Register Summary	658
17.4.2 VDP_HEVC Detail Register Description	660
17.4.3 VDP_HEVC MMU Register Summary	675
17.4.4 VDP_HEVC MMU Detail Register Description	675
17.4.5 VDP_HEVC Pref_cache Register Summary	678
17.4.6 VDP_HEVC Pref_cache Detail Register Description	679
17.4.7 VEPU Register Summary.....	682
17.4.8 VEPU Detail Register Description.....	688
17.4.9 VDPU Register Summary	722
17.4.10 VDPU Detail Register Description	727
17.4.11 VPU MMU Register Summary	779
17.4.12 VPU MMU Detail Register Description	779
17.4.13 VDPU Pref_cacheRegister Summary	782
17.4.14 VDPU_Pref_cache Detail Register Description.....	783

17.5 Application Notes	786
Chapter 18 Video Output Processor (VOP).....	787
18.1 Overview	787
18.1.1 Features	787
18.2 Block Diagram	789
18.3 Function description.....	789
18.3.1 Data Format	789
18.3.2 Data path.....	790
18.3.3 Virtual display.....	791
18.3.4 Scaling	791
18.3.5 De-flicker (Interlace vertical filtering)	793
18.3.6 Overlay.....	794
18.3.7 Scaler.....	797
18.3.8 Gamma Correction	798
18.3.9 Replication and dither down	798
18.3.10 TV ENCODE	800
18.3.11 VDAC.....	802
18.4 Register description	803
18.4.1 Register Summary.....	803
18.4.2 Detail Register Description.....	807
18.5 Timing Diagram	855
18.6 Interface Description	857
18.6.1 VOP Outputs.....	857
18.6.2 IOMUXdescription	857
18.7 Application Notes	857
18.7.1 DMA transfer mode.....	857
18.7.2 HWC data load.....	858
18.7.3 IEP direct path	859
18.7.4 GAMMA LUT.....	860
18.7.5 DMA control (QoS/Hurry/Outstanding)	861
18.7.6 Interrupt.....	861
18.7.7 RGB display mode	862
18.7.8 Immediately control register	863
18.7.9 Output Polarity Control.....	863
18.7.10 Scaler	864
Chapter 19 RGA1_plus.....	868
19.1 Overview	868
19.1.1 Features	868
19.2 Block Diagram	869
19.3 Function Description	870
19.3.1 Data Format	870
19.3.2 Dithering.....	871
19.3.3 Scaling	871
19.3.4 Rotation.....	871
19.3.5 Bitmap Block Transfer	872

19.3.6 Alpha Blending	873
19.3.7 Raster Operation (ROP)	875
19.3.8 Line/Point Drawing	875
19.3.9 Blur/sharpness	875
19.4 Register Description	875
19.4.1 Register Summary	875
19.4.2 Detail Register Description	878
19.5 Programming Guide	896
19.5.1 Register Partition	896
19.5.2 Command Modes	896
19.5.3 Command Sync	897
Chapter 20 Image Enhancement Processor (IEP)	899
20.1 Overview	899
20.2 Block Diagram	900
20.3 Function description	901
20.3.1 Deinterlace	901
20.3.2 Noise reduction	901
20.3.3 Enhancement	901
20.3.4 Format conversion	901
20.3.5 Shadow registers	902
20.3.6 VOP direct path	902
20.4 Register description	902
20.4.1 Register Summary	902
20.4.2 Detail Register Description	904
20.5 Application Notes	930
20.5.1 VOP path disabled configure flow	930
20.5.2 VOP path enabled configure flow	930
20.5.3 VOP path turn on flow	930
20.5.4 VOP path turn off flow	930
Chapter 21 Camera Interface (CIF)	931
21.1 Overview	931
21.2 Block Diagram	931
21.3 Function description	931
21.3.1 Support Vsync high active or low active	932
21.3.2 Support href high active or low active	932
21.3.3 Support CCIR656 (NTSC and PAL)	933
21.3.4 Support Raw data(8-bit) or JPEG	933
21.4 Register description	934
21.4.1 Register Summary	934
21.4.2 Detail Register Description	934
21.5 Interface description	940
21.6 Application Notes	941
Chapter 22 EBC	942
22.1 Overview	942

22.1.1 Features	942
22.2 Block Diagram	942
22.3 Function description.....	942
22.3.1 Data Format	942
22.3.2 Waveform generation mode	943
22.3.3 Window display mode	944
22.3.4 Frame control	945
22.3.5 Image translation phase.....	946
22.3.6 IO description	946
22.4 Register Description.....	950
22.4.1 Register summary	950
22.4.2 Detail Register Description.....	951
Chapter 23 LVDS.....	959
23.1 Overview	959
23.1.1 Features	959
23.2 Block Diagram	960
23.3 Function description.....	960
23.3.1 Video Data Processing.....	960
23.3.2 LVDSFORMAT	961
23.4 Register Description.....	962
23.4.1 Register Summary	962
23.4.2 Detail Register Description.....	963
23.5 Interface Description	965
23.5.1 Video Input Source.....	965
23.6 ApplicationNotes	965
23.6.1 LVDS mode	965
23.6.2 TTL mode	966
Chapter 24 HDMI Tx	967
24.1 Overview	967
24.1.1 Features	967
24.2 Block Diagram	968
24.3 Function description.....	968
24.3.1 Video Data Processing.....	968
24.3.2 Audio Data Processing.....	970
24.3.3 DDC	973
24.3.4 EDID	973
24.3.5 HDCP.....	973
24.3.6 Hot Plug Detect.....	973
24.3.7 TMDS encoder	973
24.3.8 CEC.....	973
24.3.9 Optional Lipsync.....	973
24.4 Register Description.....	974
24.4.1 Register Summary	974
24.4.2 Detail Register Description.....	976

24.5 Interface Description	1021
24.5.1 Video Input Source	1021
24.5.2 Audio Input Source	1021
24.6 ApplicationNotes	1021
24.6.1 Initial Operation	1022
24.6.2 Hot Plug Detection	1022
24.6.3 Reading EDID	1022
24.6.4 Audio input configuration	1024
24.6.5 Video input configuration	1025
24.6.6 Low Power Mode	1025
24.6.7 CEC OPERATION	1026
24.6.8 HDCP OPERATION	1027
24.6.9 NOMAL OPERATION EXAMPLE AT 1080P	1030
24.7 Electrical Specification	1031
24.7.1 CONTROL SIGNAL - DDC	1032
24.7.2 CONTROL SIGNAL - HPD	1032
Chapter 25 USB Host 2.0	1034
25.1 Overview	1034
25.1.1 Features	1034
25.2 Block Diagram	1034
25.3 USB Host2.0 Controller	1034
25.4 USB Host2.0 PHY	1034
25.5 Register Description	1034
25.6 Interface description	1035
25.7 Application Note	1035
25.7.1 Reset a port	1035
25.7.2 Relative GRF Registers	1035
Chapter 26 USB OTG 2.0	1036
26.1 Overview	1036
26.1.1 Features	1036
26.2 Block Diagram	1036
26.2.1 USB OTG 2.0 Controller Function	1036
26.2.2 USB OTG 2.0 PHY Function	1037
26.2.3 UTMI Interface	1037
26.3 USB OTG2.0 Controller	1038
26.3.1 Host Architecture	1040
26.3.2 Device Architecture	1041
26.3.3 FIFO Mapping	1041
26.4 USB OTG2.0 PHY	1043
26.4.1 Block Diagram	1044
26.5 UART BYPASS FUNCITON	1047
26.6 Register Description	1049

26.6.1 Register Summary	1049
26.6.2 Detail Register Description	1052
26.7 Interface description	1167
26.8 Application Note	1168
26.8.1 Resume from Suspend Mode	1168
26.8.2 Reset a port	1168
26.8.3 Relative GRF Registers	1169
Chapter 27 SPDIF Transmitter	1171
27.1 Overview	1171
27.2 Block Diagram	1172
27.3 Function Description	1172
27.3.1 Frame Format	1172
27.3.2 Sub-frame Format	1173
27.3.3 Channel Coding	1174
27.3.4 Preamble	1174
27.4 Register description	1175
27.4.1 Register Summary	1175
27.4.2 Detail Register Description	1176
27.5 Interface description	1180
27.6 Application Notes	1181
Chapter 28 I2S 2-channel.....	1182
28.1 Overview	1182
28.1.1 Features	1182
28.2 Block Diagram	1183
28.3 Function description	1183
28.3.1 I2S normal mode	1184
28.3.2 I2S left justified mode	1185
28.3.3 I2S right justified mode	1185
28.3.4 PCM early mode	1185
28.3.5 PCM late1 mode	1186
28.3.6 PCM late2 mode	1186
28.3.7 PCM late3 mode	1186
28.4 Register description	1187
28.4.1 Register Summary	1187
28.4.2 Detail Register Description	1187
28.5 Interface description	1196
28.6 Application Notes	1197
Chapter 29 Audio Codec	1199
29.1 Overview	1199
29.2 Block Diagram	1199
29.3 Electrical Specification	1199

29.4 Function description.....	1201
29.4.1 Digital Interface	1201
29.4.2 Analog Interface.....	1204
29.4.3 Interface Relationship	1206
29.5 Register description	1206
29.5.1 Register Summary.....	1206
29.5.2 Detail Register Description.....	1207
29.6 Application Note.....	1221
Chapter 30 Global Positioning System (GPS)	1222
30.1 Overview	1222
30.2 Block Diagram	1222
30.3 Register Summary.....	1222
30.3.1 Base band register summary.....	1223
30.3.2 ACC operation register summary	1223
30.4 Interface Description	1224
30.5 Application note	1225
Chapter 31 SPI	1226
31.1 Overview	1226
31.1.1 Features	1226
31.2 Block Diagram	1226
31.3 Function description.....	1228
31.4 Register Description.....	1229
31.4.1 Registers Summary	1230
31.4.2 Detail Register Description.....	1230
31.5 Interface description.....	1239
31.6 Application Notes	1241
Chapter 32 UART	1244
32.1 Overview	1244
32.1.1 Features	1244
32.2 Block Diagram	1244
32.3 Function description.....	1245
32.4 Register Description.....	1248
32.4.1 Registers Summary	1248
32.4.2 Detail Register Description.....	1249
32.5 Interface description.....	1264
32.6 Application Notes	1265
32.6.1 None FIFO Mode Transfer Flow	1265
32.6.2 FIFO Mode Transfer Flow	1265

32.6.3 Baud Rate Calculation	1266
Chapter 33 I2C Interface.....	1268
33.1 Overview	1268
33.1.1 Features	1268
33.2 Block Diagram	1268
33.3 Function description.....	1269
33.3.1 Initialization	1269
33.3.2 Master Mode Programming	1269
33.4 Register Description.....	1272
33.4.1 Registers Summary	1272
33.4.2 Detail Register Description.....	1272
33.5 Interface description.....	1278
33.6 Application Notes	1278
Chapter 34 General-Purpose I/O Ports(GPIO)	1282
34.1 Overview	1282
34.1.1 Features	1282
34.2 Block Diagram	1282
34.3 Function description.....	1283
34.3.1 Operation.....	1283
34.3.2 Programming	1284
34.4 Register Description.....	1285
34.4.1 Registers Summary	1285
34.4.2 Detail Register Description.....	1285
34.5 Interface description.....	1288
34.6 Application Notes	1289
Chapter 35 Timer.....	1290
35.1 Overview	1290
35.1.1 Features	1290
35.2 Block Diagram	1290
35.3 Function description.....	1290
35.3.1 Timer clock.....	1290
35.3.2 Programming sequence.....	1290
35.3.3 Loading a timer countvalue.....	1291
35.3.4 Timer mode selection.....	1291
35.4 Register Description.....	1291
35.4.1 Registers Summary	1291
35.4.2 Detail Register Description.....	1293
35.5 Application Notes	1294
Chapter 36 Pulse Width Modulation(PWM).....	1295
36.1 Overview	1295

36.1.1 Features	1295
36.2 Block Diagram	1296
36.2.1 PWM APB Slave Interface	1296
36.2.2 PWM Channels	1296
36.3 Function description.....	1296
36.3.1 Reference mode	1296
36.3.2 Continuous Mode.....	1297
36.3.3 One-shot Mode	1297
36.4 Register description	1298
36.4.1 Register Summary.....	1298
36.4.2 Detail Register Description.....	1299
36.5 Interface Description	1316
36.6 Application Notes	1316
36.6.1 PWM Reference Mode Standard Usage Flow	1316
36.6.2 PWM One-shotMode/Continuous Standard Usage Flow	1317
36.6.3 Low-power mode.....	1317
36.6.4 Other notes	1317
Chapter 37 WatchDog	1318
37.1 Overview	1318
37.2 Block Diagram	1318
37.3 Function description.....	1319
37.3.1 Operation	1319
37.3.2 Programming sequence.....	1320
37.4 Register Description.....	1320
37.4.1 Registers Summary	1320
37.4.2 Detail Register Description.....	1321
37.5 Application Notes	1323
Chapter 38 SAR-ADC	1324
38.1 Overview	1324
38.1.1 Features	1324
38.2 Block Diagram	1324
38.3 Function description.....	1324
38.4 Register Description.....	1325
38.4.1 Registers Summary	1325
38.4.2 Detail Register Description.....	1325
38.5 Timing Diagram	1326
38.6 Application Notes	1327
Chapter 39 eFuse	1328
39.1 Overview	1328
39.1.1 Features	1328

39.2 Block Diagram	1328
39.3 Function description.....	1328
39.4 Register Description.....	1329
39.4.1 Registers Summary	1330
39.4.2 Detail Register Description.....	1330
39.5 Timing Diagram	1331
39.6 Application Notes	1334
39.6.1 Operating and Burning Conditions.....	1334
39.6.2 Signal Capacitance Requirements	1334
39.6.3 Standby and Active Current	1334
Chapter 40 Process-Voltage-Temperature Monitor(PVTM)	1335
40.1 Overview	1335
40.2 Block Diagram	1335
Chapter 41 Serial Flash Controller(SFC).....	1336
41.1 Overview	1336
41.1.1 Features	1336
41.2 Block Diagram	1336
41.3 Function description.....	1336
41.3.1 SFC ahb slave.....	1336
41.4 Register Description.....	1337
41.4.1 Register Summary	1337
41.4.2 Detail Register Description.....	1338
41.5 Interface Description	1342
41.6 Application Notes	1343
41.6.1 AHB Slave write flash flow	1343
41.6.2 AHB Slave read flash flow	1344
41.6.3 AHB dma transfer flow	1345
41.6.4 Software notice.....	1345
Chapter 42 TSP(Transport Stream Processing Module)	1346
42.1 Overview	1346
42.2 Block Diagram	1346
42.3 Function Description	1347
42.3.1 TS Stream of TS_IN Interface	1347
42.3.2 TS output of TS Out Interface	1349
42.3.3 Demux and descrambling	1349
42.4 Register Description.....	1350
42.4.1 Register Summary	1350
42.4.2 Detail Register Description.....	1352
42.5 Interface Description	1386
42.6 Application Notes	1387

42.6.1 Overall Operation Sequence	1387
42.6.2 TS Source	1387
42.6.3 TS Synchronous Operation.....	1388
42.6.4 Descrambling Operation	1389
42.6.5 Demux Operation	1389
42.6.6 TS Out Interface	1392
42.6.7 PVR	1392
42.6.8 PCR extraction	1392
Chapter 43 Smart Card Controller.....	1394
43.1 Overview	1394
43.2 Block Diagram	1395
43.2.1 APB Interface	1395
43.2.2 Configuration Registers	1395
43.2.3 Controller	1395
43.2.4 Receive FIFO	1395
43.2.5 Transmit FIFO	1396
43.2.6 Clock Generator	1396
43.3 Function Description	1396
43.3.1 Smart Card Insertion	1396
43.3.2 Automatic operating voltage class selection	1396
43.3.3 Activation of Contacts and Cold Reset Sequence	1397
43.3.4 Execution of Transaction.....	1397
43.3.5 Warm Reset.....	1397
43.3.6 Deactivation of Contacts.....	1398
43.4 Register description	1399
43.4.1 Register Summary	1399
43.4.2 Detail Register Description	1400
43.5 Interface Description	1420
43.6 Application Notes	1421
43.6.1 SCR Clock	1421
Chapter 44 GMAC Ethernet Interface.....	1423
44.1 Overview	1423
44.1.1 Features	1423
44.2 Block Diagram	1424
44.3 Function Description	1425
44.3.1 Frame Structure	1425
44.3.2 RMII Interface timing diagram.....	1425
44.3.3 RGMII interface.....	1427
44.3.4 Management Interface	1428
44.3.5 Power Management Block	1429
44.3.6 MAC Management Counters	1430
44.4 Register description	1430
44.4.1 Register Summary	1430
44.4.2 Detail Register Description	1433
44.5 Interface Description	1484

44.6 Application Notes	1485
44.6.1 Descriptors.....	1485
44.6.2 Receive Descriptor.....	1486
44.6.3 Transmit Descriptor	1490
44.6.4 Programming Guide	1493
44.6.5 Clock Architecture	1496
44.6.6 Remote Wake-Up Frame Filter Register.....	1498
44.6.7 System Consideration During Power-Down	1499
44.6.8 GRF Register Summary	1499
Chapter 45 MIPI Controller	1501
45.1 Overview	1501
45.2 Block Diagram	1501
45.3 Function Description	1502
45.3.1 DPI interface function	1502
45.3.2 APB Slave Generic Interface.....	1504
45.3.3 Transmission of Commands in Video Mode	1506
45.4 Register Description.....	1510
45.4.1 Register Summary.....	1511
45.4.2 Detail Register Description.....	1512
45.5 Application Notes	1533
Chapter 46 MIPI D-PHY	1535
46.1 Overview	1535
46.2 Block Diagram	1535
46.3 Function Description	1537
46.4 Register Description.....	1537
46.4.1 Register Summary.....	1538
46.4.2 Detail Register Description.....	1539
46.5 Interface Timing.....	1566
46.6 Application Notes	1567
46.6.1 Low power mode	1567
46.6.2 Programmable PLL IN DSI TX.....	1567
46.6.3 LVDS mode	1568
46.6.4 Other mipi_phy grf	1568
46.7 ELECTRICAL SPECIFICATIONS	1569
46.7.1 DC SPECIFICATIONS	1569
46.7.2 AC specifications	1570
Chapter 47 I2S 8-channel.....	1573
47.1 Overview	1573
47.1.1 Features	1573
47.2 Block Diagram	1574
47.3 Function description.....	1574

47.3.1 i2s normal mode	1575
47.3.2 i2s left justified mode	1576
47.3.3 i2s right justified mode	1576
47.3.4 PCM early mode	1577
47.3.5 PCM late1 mode	1577
47.3.6 PCM late2 mode	1577
47.3.7 PCM late3 mode	1578
47.4 Register Description.....	1578
47.4.1 Registers Summary	1578
47.4.2 Detail Register Description.....	1579
47.5 Application Notes	1588

Rockchip Confidential

Figure Index

Fig.1-1RKaudi Block Diagram	45
Fig.1-2RK3126 LQFP176 Package Top View	66
Fig.1-3RK3126 LQFP176 Package Side View	66
Fig.1-4 RK3126 LQFP176 Dimension	67
Fig.1-5RK3128 TFBGA316 Package Top View	67
Fig.1-6RK3128 TFBGA316 Package Side View	68
Fig.1-7RK3128 TFBGA316 Package Bottom View	68
Fig.1-8RK3128 TFBGA316 Package Dimension	69
Fig.1-9LQFP176 Ball Map	71
Fig.1-10TFBGA316 Ball Map	73
Fig.1-11 External Reference Circuit for 24MHzOscillators	92
Fig.1-12RK3128 USB OTG/Host2.0 differential lines requirement	93
Fig.1-13RK3128 USB OTG/Host2.0 ground plane guide	93
Fig.1-14RK3128 USB OTG/Host2.0 component placement	94
Fig.1-15RK3128 HDMI interface reference connection	94
Fig.1-16RK3128 HDMI CEC interface reference connection	95
Fig.1-17RK3128 HDMI ESD interface reference connection	95
Fig.1-18RK3128 Audio Codec interface reference connection	96
Fig.1-19 RK3128 reset signals sequence	96
Fig. 2-1 RKaudi Address Mapping	97
Fig. 2-2 RKaudi boot procedure flow	99
Fig. 3-1 CRU Architecture	103
Fig. 3-2Chip Clock Architecture Diagram 1	104
Fig. 3-3Chip Clock Architecture Diagram 2	105
Fig. 3-4Chip Clock Architecture Diagram 3	106
Fig. 3-5Reset Architecture Diagram	108
Fig. 3-6PLL Block Diagram	109
Fig. 3-7Chip Power On Reset Timing Diagram	161
Fig.4-1Debug system structure	165
Fig.4-2 DAP SWJ interface	171
Fig.4-3SW-DP acknowledgement timing	172
Fig.6-1MP Subsystem architecture	252
Fig.7-1Block Diagram	253
Fig.7-2Connectivity	256
Fig. 8-1 Block diagram of dmac	271
Fig. 8-2 DMAC operation states	272
Fig. 8-3 DMAC0 request and acknowledge timing	289
Fig. 9-1Block diagram of GIC	298
Fig. 9-2 GIC Interrupt handling state machine	301
Fig. 11-1 Protocol controller architecture	317
Fig. 10-2 PHY controller architecture	318
Fig. 11-3 Protocol controller architecture	319
Fig. 11-1 Crypto Architecture	400
Fig. 12-1 NandC Block Diagram	425
Fig. 12-2 NandC Address Assignment	544
Fig. 12-3 NandC DataFormat	545
Fig. 12-4 NandC LLP Data Format	548
Fig. 13-1 Host Controller Block Diagram	551
Fig. 13-2 SD/MMC Card-Detect Signal	556
Fig. 13-3 Host Controller Command Path State Machine	558
Fig. 13-4 Host Controller Data Transmit State Machine	561
Fig. 13-5 Host Controller Data Receive State Machine	563

Fig. 13-6 SD/MMC Card-Detect and Write-Protect	596
Fig. 13-7 SD/MMC Card Termination	597
Fig. 13-8 Host Controller Initialization Sequence	600
Fig. 13-9 Voltage Switching Command Flow Diagram.....	612
Fig. 13-10 ACMD41 Argument.....	613
Fig. 13-11 ACMD41 Response(R3)	613
Fig. 13-12 Voltage Switch Normal Scenario	614
Fig. 13-13 Voltage Switch Error Scenario	615
Fig. 13-14 CASES for eMMC 4.5 START bit	618
Fig. 13-15 Clock Generation Unit	620
Fig. 13-16 Card Detection Method 2	624
Fig. 13-17 Card Detection Method 4	625
Fig.14-1Power Domain Partition	626
Fig.14-2PMU Block Diagram.....	627
Fig.14-3 Each Domain Power Switch Timing	639
Fig.14-4 External Wakeup Source PAD Timing	640
Fig. 15-1 Embedded SRAM block diagram	642
Fig. 16-1 GPU block diagram.....	644
Fig. 16-2 GPU_ACLK generate block diagram.....	646
Fig. 16-3 GPU interrupt connection	647
Fig.17-1VPU Combo in SOC	650
Fig.17-2VPU Combo Block Diagram.....	651
Fig.17-4structure of two-level page table	655
Fig.17-5 Dataflow of HW performs entropy decoding in video decoder.....	656
Fig.17-6 Dataflow of SW performs entropy decoding in video decoder	657
Fig. 19-1 RGA Block Diagram	869
Fig. 19-2 RGA Input Data Format	870
Fig. 19-3 RGA Dither effect.....	871
Fig. 19-4 RGA Rotation AA effect	872
Fig. 19-5 RGA Gradient Fill	873
Fig. 19-6 RGA Alpha blanding	874
Fig. 19-7 RGA Line drawing	875
Fig. 19-8 HDMI TX Software Main Sequence Diagram	896
Fig. 19-9 RGA command line and command counter.....	897
Fig. 19-10 RGA command sync generation	897
Fig. 20-1 IEP block diagram	900
Fig. 21-1CIF block diagram.....	931
Fig. 21-2 Timing diagram for CIF when vsync low active	932
Fig. 21-3Timing diagram for CIF when vsync high active	932
Fig. 21-4Timing diagram for CIF when href high active	932
Fig. 21-5Timing diagram for CIF when href low active	932
Fig. 21-6Timing diagram for CIF when Y data first.....	932
Fig. 21-7Timing diagram for CIF when U data first	933
Fig. 21-8CCIR656 timing	933
Fig. 21-9Raw Data or JPEG Timing	933
Fig.22-1EBC Block Diagram	942
Fig.22-2 EBC Block Diagram	943
Fig.22-3 EBC LUT structure.....	944
Fig.22-4 EBC window display	945
Fig.22-5 EBC single frame display	945
Fig.22-6 EBC multi-frame display	945
Fig.22-7 EBC display phase.....	946
Fig.22-8 EBC source driver timing 1	948
Fig.22-9 EBC source driver timing 2	948

Fig.22-10 EBC gate driver timing 1	949
Fig.22-11 EBC gate driver timing 2	950
Fig.23-1LVDS TX Block Diagram	960
Fig.23-2LVDS TX InterfaceTiming	961
Fig. 24-1 HDMI TX Block Diagram.....	968
Fig. 24-2 HDMI Video Data Processing.....	968
Fig. 24-3HDMI Video Processing Timing.....	969
Fig. 24-4HDMI Audio Data Processing Diagram.....	971
Fig. 24-5HDMI Audio Clock Regeneration Model.....	972
Fig. 25-1 USB HOST 2.0 Architecture	1034
Fig. 26-1 USB OTG 2.0 Architecture	1036
Fig. 26-2 UTMI interface – Transmit timing for a data packet	1038
Fig. 26-3 UTMI interface – Receive timing for a data packet	1038
Fig. 26-4 USB OTG2.0 Controller Architecture.....	1039
Fig. 26-5 DFIFO single-port synchronous SRAM interface	1040
Fig. 26-6 USB OTG 2.0 Controller host mode FIFO address mapping.....	1042
Fig. 26-7 USB OTG 2.0 Controller device mode FIFO address mapping....	1043
Fig. 26-8 usb phy architecture	1044
Fig. 26-9 UART Application	1048
Fig. 26-10 UART Timing Sequence	1048
Fig. 26-11 Resume Timing Sequence.....	1168
Fig. 26-12 Reset a port when receiving.....	1169
Fig. 26-13 Reset a port when transmitting	1169
Fig.27-1SPDIF transmitter Block Diagram	1172
Fig.27-2SPDIF Frame Format	1173
Fig.27-3SPDIF Sub-frame Format	1173
Fig.27-4SPDIF Channel Coding.....	1174
Fig.27-5SPDIF Preamble	1175
Fig.27-6 SPDIF transmitter operation flow chart	1181
Fig.28-1I2S/PCM1/2 controller (2 channel) Block Diagram	1183
Fig.28-2I2S transmitter-master & receiver-slave condition	1184
Fig.28-3I2S transmitter-slave & receiver-master condition	1184
Fig.28-4I2S normal mode timing format.....	1185
Fig.28-5I2S left justified mode timing format	1185
Fig.28-6I2S right justified mode timing format	1185
Fig.28-7PCM early mode timing format.....	1185
Fig.28-8PCM late1 mode timing format.....	1186
Fig.28-9PCM late2 mode timing format.....	1186
Fig.28-10PCM late3 mode timing format.....	1187
Fig.28-11I2S/PCM1/2 controller transmit operation flow chart	1197
Fig.28-12I2S/PCM1/2 controller receive operation flow chart	1198
Fig.29-1 Audio Codec Block Diagram.....	1199
Fig.29-2 Left Justified Mode (assuming n-bit word length).....	1202
Fig.29-3 Right Justified Mode (assuming n-bit word length).....	1202
Fig.29-4 I2S Mode (assuming n-bit word length)	1203
Fig.29-5 DSP/PCM Mode A (assuming n-bit word length).....	1203
Fig.29-6 DSP/PCM Mode B (assuming n-bit word length).....	1204
Fig.29-7MicroPhone Input.....	1204
Fig.29-8 ADC Channels Relationship.....	1206
Fig.29-9 DAC Channels Relationship.....	1206
Fig. 30-1 RKAudi GPS block diagram	1222
Fig.31-1SPI Controller Block diagram	1227
Fig.31-2SPI Master and Slave Interconnection.....	1228
Fig.31-3SPI Format (SCPH=0 SCPOL=0)	1229

Fig.31-4SPI Format (SCPH=0 SCPOL=1)	1229
Fig.31-5SPI Format (SCPH=1 SCPOL=0)	1229
Fig.31-6SPI Format (SCPH=1 SCPOL=1)	1229
Fig.31-7SPI Master transfer flow diagram	1242
Fig.31-8SPI Slave transfer flow diagram	1243
Fig. 32-1UART Architecture.....	1244
Fig. 32-2UART Serial protocol	1245
Fig. 32-3UART baud rate	1246
Fig. 32-4UART Auto flow control block diagram.....	1247
Fig. 32-5UART AUTO RTS TIMING	1247
Fig. 32-6UART AUTO CTS TIMING	1248
Fig. 32-7UART none fifo mode.....	1265
Fig. 32-8UART fifo mode	1266
Fig. 32-9UART clock generation.....	1267
Fig.33-1I2C architecture	1268
Fig.33-2I2C DATA Validity.....	1271
Fig.33-3I2C Start and stop conditions	1271
Fig.33-4I2C Acknowledge	1271
Fig.33-5I2C byte transfer	1272
Fig.33-6I2C Flow chat for transmit only mode	1279
Fig.33-7I2C Flow chat for receive only mode	1280
Fig.33-8I2C Flow chat for mix mode	1281
Fig.34-1GPIO block diagram	1282
Fig.34-2 GPIO Interrupt RTL Block Diagram	1284
Fig. 35-1Timers Block Diagram	1290
Fig. 35-2Timer Usage Flow	1291
Fig. 35-3Timing between timer_en and timer_clk.....	1294
Fig.36-1PWM architecture.....	1296
Fig.36-2PWM Reference Mode	1296
Fig.36-3PWM Left-aligned Output Mode	1297
Fig.36-4PWM Center-aligned Output Mode	1297
Fig. 36-5PWM Center-aligned Output Mode	1297
Fig. 37-1 WDT block diagram	1318
Fig. 37-2 WDT Operation Flow.....	1320
Fig.38-1RKAudioSAR-ADC block diagram	1324
Fig.38-2SAR-ADC timing diagram in single-sample conversion mode.....	1326
Fig. 39-1RKaudi eFuse block diagram	1328
Fig. 39-2RKaudi efuse timing diagram in program mode	1332
Fig. 39-3RKaudi eFuse timing diagram in read mode.....	1333
Fig.41-1 SFC architecture	1336
Fig.41-2 idle cycles.....	1337
Fig.41-3spi mode	1337
Fig.41-4 slave mode write	1343
Fig.41-5 slave mode read	1344
Fig.41-6 master mode flow	1345
Fig. 42-1 TSP architecture	1347
Fig. 42-2 Sync/Valid Serial Mode with Msb-Lsb Bit Ordering.....	1348
Fig. 42-3 Sync/valid Parallel Mode	1348
Fig. 42-4 Sync/Burst Parallel Mode.....	1348
Fig. 42-5 Nosync/Valid Parallel Mode	1349
Fig.43-1 SCR Block Diagram	1395
Fig.43-2 Activation, Cold Reset and ATR	1397
Fig.43-3 Warm Reset and ATR	1398
Fig.43-4 Deactivation Sequence	1399

Fig. 44-1 GMAC architecture	1424
Fig. 44-2 MAC Frame structure.....	1425
Fig. 44-3 RMII transmission bit ordering	1426
Fig. 44-4 Start of MII and RMII transmission in 100-Mbps mode	1426
Fig. 44-5 End of MII and RMII Transmission in 100-Mbps Mode	1426
Fig. 44-6 Start of MII and RMII Transmission in 10-Mbps Mode	1427
Fig. 44-7 End of MII and RMII Transmission in 10-Mbps Mode.....	1427
Fig. 44-8 RMII receive bit ordering	1427
Fig. 44-9 MDIO frame structure.....	1428
Fig. 44-10 Descriptor Ring and Chain Structure	1486
Fig. 44-11 Rx/Tx Descriptors definition.....	1486
Fig. 44-12 RMII clock architecture when clock source from CRU	1497
Fig. 44-13 RMII clock architecture when clock source from external OSC.	1497
Fig. 44-14 RGMII clock architecture when clock source from CRU.....	1498
Fig. 44-15 Wake-Up Frame Filter Register	1498
Fig. 45-1MIPI Controller architecture.....	1502
Fig. 45-224 bpp APB Pixel to Byte Organization	1505
Fig. 45-318 bpp APB Pixel to Byte Organization	1506
Fig. 45-416 bpp APB Pixel to Byte Organization	1506
Fig. 45-512 bpp APB Pixel to Byte Organization	1506
Fig. 45-68 bpp APB Pixel to Byte Organization	1506
Fig. 45-7Command Transmission Periods within the Image Area	1507
Fig. 45-8 Location of outvact_lpcmd_time and invact_lpcmd_time in the Image Area	1509
Fig.46-1MIPI D-PHY simplified Block diagram with master to slave	1536
Fig.46-2MIPI D-PHY V1.0 detailed block diagram	1536
Fig.46-3HS-TX PPI Timing	1566
Fig.46-4LPDT TX PPI Timing.....	1567
Fig.46-5LPDT RX PPI Timing	1567
Fig.47-1 I2S/PCM controller (8 channel) Block Diagram	1574
Fig.47-2 I2S transmitter-master & receiver-slave condition	1575
Fig.47-3 I2S transmitter-slave& receiver-master condition	1575
Fig.47-4 I2S normal mode timing format	1576
Fig.47-5 I2S left justified mode timing format	1576
Fig.47-6 I2S right justified modetiming format	1576
Fig.47-7 PCM early modetiming format.....	1577
Fig.47-8 PCM late1 modetiming format.....	1577
Fig.47-9 PCM late2 modetiming format.....	1578
Fig.47-10 PCM late3 modetiming format.....	1578
Fig.47-11 I2S/PCM controller (8 channel) transmit operation flow chart ..	1588
Fig.47-12 I2S/PCM controller (8 channel) receive operation flow chart....	1589

Table Index

Table 1-1 RK3128 Power/Ground IO informations.....	45
Table 1-2 RK3128 IO descriptions	48
Table 1-3 RK3128 IO function description list	59
Table 1-4 RK3128 IO Type List	64
Table 1-5 RK3126 LQFP176 Pin Number Order Information.....	74
Table 1-6 RK3128 absolute maximum ratings	78
Table 1-7 RK3128 recommended operating conditions	78
Table 1-8 RK3128 DC Characteristics.....	79
Table 1-9 Recommended operating frequency for PLL and oscillator domain	83
Table 1-10 Recommended operating frequency for A9 core.....	83
Table 1-11 Recommended operating frequency for PD_CPU domain	83
Table 1-12 Recommended operating frequency for PD_PERI domain	84
Table 1-13 Recommended operating frequency for PD_VIO domain	85
Table 1-14 Recommended operating frequency PD_GPU domain.....	85
Table 1-15 Recommended operating frequency for PD_VIDEO domain.....	86
Table 1-16 RK3128 Electrical Characteristics for Digital General IO	86
Table 1-17 RK3128 Electrical Characteristics for PLL	87
Table 1-18 RK3128 Electrical Characteristics for SAR-ADC	87
Table 1-19 RK3128 Electrical Characteristics for USB OTG/Host2.0 Interface	88
Table 1-20 RK3128 Electrical Characteristics for HDMI	89
Table 1-21 RK3128 Electrical Characteristics for DDR IO	89
Table 1-22 RK3128 Electrical Characteristics for eFuse	89
Table 1-23 RK3128 Electrical Characteristics for MIPI HS transmitter	90
Table 1-24 RK3128 Electrical Characteristics for LVDS HS transmitter	90
Table 1-25 RK3128 Electrical Characteristics for VDAC	90
Table 2-1 RKAudi Interrupt connection list.....	100
Table 2-2 RKAudi DMAC_BUS Hardware request connection list	102
Table 3-1 Input clock description in clock architecture diagram	108
Table 4-1 SW-DP Interface Description	172
Table 7-1Master NIU	253
Table 7-2slave NIU	254
Table 7-3Clock and Power domain.....	254
Table 7-4 DDR configuration	257
Table 8-1DMAC Request Mapping Table.....	270
Table 8-2 DMAC Instruction sets.....	296
Table 10-1 DDR IO description	389
Table 10-2 DDR PHY TX DLLs Delay Step	395
Table 10-3 DDR PHY RX DQS Delay Step	396
Table 10-4 CK, CMD Signal Drive Strength Register.....	396
Table 10-5 DM, DQ Signal Drive Strength Register	397
Table 10-6 DM/DQ/DQS/CMD Driver output resistance with control bit	397
Table 10-7 DM/DQ/DQS RX ODT resistance with control bit	397
Table 10-8 Low Power DLL Setting	397
Table 10-9 per-bit de-skew tuning resolution.....	398
Table 12-1 NandC Address Mapping	427
Table 12-2 NandC Interface Description	538
Table 13-1 Bits in Interrupt Status Register.....	553
Table 13-2 Auto-Stop Generation.....	565
Table 13-3 Non-data Transfer Commands and Requirements.....	566
Table 13-4 IOMUX Settings for SDMMC.....	595
Table 13-5 IOMUX Settings for SDIO	595
Table 13-6 IOMUX Settings for eMMC	596

Table 13-7 Recommended Usage of use_hold_reg	598
Table 13-8 Command Settings for No-Data Command	603
Table 13-9 Command Setting for Single or Multiple-Block Read	605
Table 13-10 Command Settings for Single or Multiple-Block Write	606
Table 13-11 PBL and Watermark Levels	620
Table 13-12 Configuration for SDMMC Clock Generation	620
Table 13-13 Configuration for SDIO Clock Generation	621
Table 13-14 Configuration for eMMC Clock Generation	621
Table 13-15 Register for SDMMC Card Detection Method 3	624
Table 14-1RK3128 Power Domain and Voltage Domain Summary	626
Table 14-2 Power Switch Timing	627
Table 14-3 Low Power State.....	628
Table 14-4 Power Domain Status Summary in all Work Mode	640
Table 19-1 RGA 8bpp color palette LUT	873
Table 19-2 RGA Porter-Duff alpha factor	874
Table 19-3 RGA ROP Boolean operations.....	875
Table 22-1EBC Input Data Format.....	942
Table 22-2EBC output pins.....	946
Table 22-3EBC IOMUX.....	947
Table 22-4EBC source driver setting	948
Table 22-5EBC gate driver setting	950
Table 23-1 is LVDSFORMAT Input Data Format	961
Table 24-1HDMI Supported Input Video Formats.....	969
Table 24-2HDMI TX I2S 2Channel Audio Sampling Frequency at Each Video Format	971
Table 24-3HDMI TX I2S 8 Channel Audio Sampling Frequency at Each Video Format	971
Table 24-4HDMI SPDIF Sampling Frequency at Each Video Format	972
Table 24-5Normal Operating Conditions	1031
Table 24-6Absolute Maximum Conditions	1031
Table 24-7Transmitter DC Characteristics at TP1	1031
Table 24-8Transmitter AC Characteristics at TP1	1032
Table 24-9Programmable Output Resistance and Output Equalization Level	1032
Table 25-1 USB HOST 2.0 Interface Description	1035
Table 26-1 USB OTG 2.0 Interface Description	1167
Table 27-1 Input clock description in clock architecture diagram	1180
Table 29-1 Supported Data Formats in Different Modes	1201
Table 31-1SPI interface description in master mode	1240
Table 31-2SPI interface description in slave mode	1240
Table 32-1UART Interface Description	1264
Table 32-2UART baud rate configuration	1267
Table 33-1I2C Interface Description	1278
Table 34-1GPIO interface description.....	1288
Table 36-1PWM Interface Description	1316
Table 38-1RKAudi SAR-ADC timing parameters list	1327
Table 39-1RKAudi eFuse program timing parameters list	1332
Table 41-1 IOMUX Settings for SFC.....	1342
Table 42-1 TSP Interface Description.....	1386
Table 43-1IOMUX Setting	1420
Table 43-2BAUDTUNE register	1421
Table 44-1 RMII Interface Description	1484
Table 44-2 RGMII Interface Description	1484
Table 44-3 Receive Descriptor 0	1486
Table 44-4 Receive Descriptor 1	1488
Table 44-5 Receive Descriptor 2	1489
Table 44-6 Receive Descriptor 3	1490

Table 44-7 Transmit Descriptor 0	1490
Table 44-8 Transmit Descriptor 1	1492
Table 44-9 Transmit Descriptor 2	1493
Table 44-10 Transmit Descriptor 3	1493
Table 45-1Color table.....	1503
Table 46-1 HS Transmitter DC specifications.....	1569
Table 46-2 HS Transmitter DC specifications.....	1569
Table 46-3 LP Transmitter DC Specifications	1569
Table 46-4 HS receiver AC specifications.....	1570
Table 46-5 LP receiver AC specifications	1570
Table 46-6 LP Transmitter AC specifications.....	1570

Rockchip Confidential

Chapter 1 Introduction

RKaudi is a high performance application processor for low-end quad-core tablet, and TV-Box. Especially it is one High-integration and competitive Bom Cost SOC for H.265 1080p TV-Box.

Integrate quad-core Cortex-A7 with separately Neon and FPU coprocessor , also shared 256KB L2 Cache.

Mali400 MP2 GPU is embedded to support smoothly high-resolution (1080p) display and mainstream game.

Lots of high-performance interface to get very flexible solution, such as multi-pipe display with LVDS, MIPI-DSI, HDMI1.4a,VDAC .

Crypto hardware integrated for support security BOOT.

32bits DDR3/LPDDR2 and 16bits DDR3 provide high memory bandwidths for high-performance or low-cost application.

HEVC hardware integrated for support 1080p H.265 video.

1.1 Features

1.1.1 MicroProcessor

- Quad-core ARM Cortex-A7MPCore processor, a high-performance, low-power and cached application processor
- Full implementation of the ARM architecture v7-A instruction set, ARM Neon Advanced SIMD (single instruction, multiple data) support for accelerated media and signal processing computation
- Separately Integrated Neon and FPU per CPU
- 32KB/32KB L1 ICache/DCache per CPU
- Unified 256KB L2 Cache

1.1.2 Memory Organization

- Internal on-chip memory
 - 16KB BootRom
 - 8KB internal SRAM
- External off-chip memory^①
 - DDR3-800/DDR3L-800, 32bits data widths, 2 ranks, totally 4GB(max) address space, maximum address space for one rank is also 2GB.
 - LPDDR2-800, 32bits data width, 2 ranks, totally 4GB(max) address space, maximum address space for one rank is 2GB.
 - Async/Toggle/SyncNand Flash(include LBA Nand), 8bits data width,4 banks, 60bits ECC

1.1.3 Internal Memory

- Internal BootRom
 - Size : 16KB
 - Support system boot from the following device :
 - ◆ 8bits Async Nand Flash
 - ◆ 8bits toggle Nand Flash
 - ◆ SPI interface

- ◆ eMMC interface
- ◆ SDMMC interface
- Support system code download by the following interface:
 - ◆ USB OTG interface
- Internal SRAM
 - Size : 8KB

1.1.4 External Memory or Storage device

- Dynamic Memory Interface (DDR3/DDR3L/LPDDR2)
 - Compatible with JEDEC standard DDR3/DDR3L/LPDDR2 SDRAM
 - Data rates up to 800Mbps(400MHz) for DDR3/DDR3L/LPDDR2
 - Support up to 2 ranks (chip selects) for each channel, totally 4GB(max) address space.
 - 16bits/32bits data width is software programmable
 - 7 host ports with 64bits/128bits AXI bus interface for system access, AXI bus clock is asynchronous with DDR clock
 - Programmable timing parameters to support DDR3/DDR3L/LPDDR2 SDRAM from various vendor
 - Advanced command reordering and scheduling to maximize bus utilization
 - Low power modes, such as power-down and self-refresh for DDR3/LPDDR2 SDRAM; clock stop and deep power-down for LPDDR2 SDRAM
 - Compensation for board delays and variable latencies through programmable pipelines
 - Programmable output and ODT impedance with dynamic PVT compensation
- Nand Flash Interface
 - Support 8bits async/toggle/syncnandflash, up to 4 banks
 - Support LBA nandflash
 - 16bits, 24bits, 40bits, 60bits hardware ECC
 - For DDR nandflash, support DLL bypass and 1/4 or 1/8 clock adjust, maximum clock rate is 66.5MHz
 - For async/togglenandflash, support configurable interface timing , maximum data rate is 16bit/cycle
 - Embedded AHB master interface to do data transfer by DMA method
 - Also support data transfer by AHB slave interface together with external DMAC
- eMMC Interface
 - Compatible with standard iNAND interface
 - Support MMC4.41 protocol
 - Provide eMMC boot sequence to receive boot data from external eMMC device
 - Support FIFO over-run and under-run prevention by stopping card clock automatically
 - Support CRC generation and error detection
 - Embedded clock frequency division control to provide programmable baud rate
 - Support block size from 1 to 65535Bytes
 - 8bits data bus width

- SD/MMC Interface
 - Compatible with SD3.0, MMC ver4.41
 - Support FIFO over-run and under-run prevention by stopping card clock automatically
 - Support CRC generation and error detection
 - Embedded clock frequency division control to provide programmable baud rate
 - Support block size from 1 to 65535Bytes
 - Data bus width is 4bits

1.1.5 System Component

- CRU (clock & reset unit)
 - Support clock gating control for individual components inside RKaudi
 - One oscillator with 24MHz clock input and 4 embedded PLLs
 - Up to 2.0GHz clock output for all PLLs
 - Support global soft-reset control for whole SOC, also individual soft-reset for every components
- PMU(power management unit)
 - Multipleconfigurable work modes to save power by different frequency or automatical clock gating control or power domain on/off control
 - Lots of wakeup sources in different mode
 - 2 separate voltage domains
 - 3 separate power domains, which can be power up/down by software based on different application scenes
- Timer
 - 6 on-chip 64bits Timers in SoC with interrupt-based operation
 - Provide two operation modes: free-running and user-defined count
 - Support timer work state checkable
 - Fixed 24MHz clock input
- PWM
 - Four on-chip PWMs with interrupt-based operation
 - Programmable pre-scaled operation to bus clock and then further scaled
 - Embedded 32-bit timer/counter facility
 - Support capture mode
 - Support continuous mode or one-shot mode
 - Provides reference mode and output various duty-cycle waveform
- WatchDog
 - 32 bits watchdog counter width
 - Counter clock is from app bus clock
 - Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
 - WDT can perform two types of operations when timeout occurs:
 - ◆ Generate a system reset
 - ◆ First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
 - Programmable reset pulse length
 - Totally 16 defined-ranges of main timeout period

- Bus Architecture
 - 128bit/64-bit/32-bit multi-layer AXI/AHB/APB composite bus architecture
 - 5 embedded AXI interconnect
 - ◆ CPU interconnect with four 64-bits AXI masters, one 64-bits AXI slaves, one 32-bits AHB master and lots of 32-bits AHB/APB slaves
 - ◆ PERI interconnect with two 64-bits AXI masters, one 64-bits AXI slave, five 32-bits AHB masters and lots of 32-bits AHB/APB slaves
 - ◆ Display interconnect with three 128-bits AXI master, four 64-bits AXI masters and one 32-bits AHB slave
 - ◆ GPU interconnect with one 128-bits AXI master with point-to-point AXI-lite architecture and 32-bits APB slave
 - ◆ VCODEC interconnect also with two 64-bits AXI master and two 32-bits AHB slave, they are point-to-point AXI-lite architecture
 - Flexible different QoS solution to improve the utility of bus bandwidth
- Interrupt Controller
 - Support 3 PPI interrupt source and 74 SPI interrupt sources input from different components inside RKaudi
 - Support 16 software-triggered interrupts
 - Input interrupt level is fixed, only high-level sensitive
 - Two interrupt outputs (nFIQ and nIRQ) separately for each Cortex-A7, both are low-level sensitive
 - Support different interrupt priority for each interrupt source, and they are always software-programmable
- DMAC
 - Micro-code programming based DMA
 - The specific instruction set provides flexibility for programming DMA transfers
 - Linked list DMA function is supported to complete scatter-gather transfer
 - Support internal instruction cache
 - Embedded DMA manager thread
 - Support data transfer types with memory-to-memory, memory-to-peripheral, peripheral-to-memory
 - Signals the occurrence of various DMA events using the interrupt output signals
 - Mapping relationship between each channel and different interrupt outputs is software-programmable
 - One embedded DMA controller PERI_DMAM for peripheral system
 - PERI_DMAM features:
 - ◆ 8 channels totally
 - ◆ 16 hardware request from peripherals
 - ◆ 2 interrupt output
 - ◆ Not support trustzone technology
- Security system
 - Embedded encryption and decryption engine
 - ◆ Support AES 128/192/256 bits key mode, ECB/CBC/CTR chain mode, Slave/FIFO mode
 - ◆ Support DES/3DES (ECB and CBC chain mode), 3DES (EDE/ EEE key mode), Slave/FIFO mode
 - ◆ Support SHA1/SHA256/MD5 (with hardware padding) HASH

- ◆ function, FIFO mode only
- ◆ Support 160 bit Pseudo Random Number Generator (PRNG)
- ◆ Support PKA 512/1024/2048 bit Exp Modulator
- ◆ Support up to 150M clock frequency

1.1.6 Video CODEC

- Shared internal memory and bus interface for video decoder and encoder^②
- Embedded memory management unit(MMU)
- Video Decoder
 - Real-time video decoder of MPEG-1, MPEG-2, MPEG-4,H.263, H.264, H.265,VC-1, RV, VP6/VP8, Sorenson Spark, MVC
 - MMU Embedded
 - Supports frame timeout interrupt , frame finish interrupt and bitstream error interrupt
 - Error detection and concealment support for all video formats
 - Output data format is YUV420 semi-planar, and YUV400(monochrome) is also supported for H.264
 - H.265 up to MP Level 4.1 High Tier : 1080P@60fps (1920x1080)
 - H.264 up to HP level 5.1 : 1080p@60fps (1920x1088)^③
 - MPEG-4 up to ASP level 5 : 1080p@60fps (1920x1088)
 - MPEG-2 up to MP : 1080p@60fps (1920x1088)
 - MPEG-1 up to MP : 1080p@60fps (1920x1088)
 - H.263 : 576p@60fps(720x576)
 - Sorenson Spark : 1080p@60fps (1920x1088)
 - VC-1 up to AP level 3 : 1080p@30fps (1920x1088)
 - RV8/RV9/RV10 : 1080p@60fps (1920x1088)
 - VP6/VP8 : 1080p@60fps (1920x1088)
 - MVC : 2160p@24fps (3840x2160)
 - For H.264, image cropping not supported
 - For MPEG-4, GMC(global motion compensation)not supported
 - For VC-1, upscaling and range mapping are supported in image post-processor
 - For MPEG-4 SP/H.263/Sorenson spark, using a modified H.264 in-loop filter to implement deblocking filter in post-processor unit
- Video Encoder
 - Support video encoder for H.264 UP to HP@level4.1, MVC and VP8
 - Only support I and P slices, not B slices
 - Support error resilience based on constrained intra prediction and slices
 - Input data format:
 - ◆ YCbCr 4:2:0 planar
 - ◆ YCbCr 4:2:0 semi-planar
 - ◆ YCbYCr 4:2:2
 - ◆ CbYCrY 4:2:2 interleaved
 - ◆ RGB444 and BGR444
 - ◆ RGB555 and BGR555
 - ◆ RGB565 and BGR565
 - ◆ RGB888 and BRG888
 - ◆ RGB101010 and BRG101010
 - Image size is from 96x96 to 1920x1088(Full HD)
 - Maximum frame rate is up to 30fps@1920x1080^③
 - Bit rate supported is from 10Kbps to 20Mbps

1.1.7 JPEG CODEC

- JPEG decoder
 - Input JPEG file : YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 sampling formats
 - Output raw image : YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 semi-planar
 - Decoder size is from 48x48 to 8176x8176(66.8Mpixels)
 - Support JPEG ROI(region of image) decode
 - Maximum data rate^④ is up to 76million pixels per second
 - Embedded memory management unit(MMU)
- JPEG encoder
 - Input raw image :
 - ◆ YCbCr 4:2:0 planar
 - ◆ YCbCr 4:2:0 semi-planar
 - ◆ YCbYCr 4:2:2
 - ◆ CbYCrY 4:2:2 interleaved
 - ◆ RGB444 and BGR444
 - ◆ RGB555 and BGR555
 - ◆ RGB565 and BGR565
 - ◆ RGB888 and BRG888
 - ◆ RGB101010 and BRG101010
 - Output JPEG file : JFIF file format 1.02 or Non-progressive JPEG
 - Encoder image size up to 8192x8192(64million pixels) from 96x32
 - Maximum data rate^④ up to 90million pixels per second
 - Embedded memory management unit(MMU)

1.1.8 Image Enhancement

- Image pre-processor
 - Only used together with HD video encoder inside RKaudi, not support stand-alone mode
 - Provides RGB to YCbCr 4:2:0 color space conversion, compatible with BT601, BT709 or user defined coefficients
 - Provides YCbCr4:2:2 to YCbCr4:2:0 color space conversion
 - Support cropping operation from 8192x8192 to any supported encoding size
 - Support rotation with 90 or 270 degrees
- Video stabilization
 - Work in combined mode with HD video encoder inside RKaudi and stand-alone mode
 - Adaptive motion compensation filter
 - Support scene detection from video sequence, encodes key frame when scene change noticed
- Image Post-Processor(embedded inside video decoder)
 - Combined with HD video decoder and JPEG decoder, post-processor can read input data directly from decoder output to reduce bus bandwidth
 - Also work as a stand-alone mode, its input data is from image data stored in external memory
 - Input data format:
 - ◆ Any format generated by video decoder in combined mode

- ◆ YCbCr4:2:0 semi-planar
- ◆ YCbCr4:2:0 planar
- ◆ YCbYCr 4:2:2
- ◆ YCrYCb 4:2:2
- ◆ CbYCrY 4:2:2
- ◆ CrYCbY 4:2:2
- Output data format:
 - ◆ YCbCr4:2:0 semi-planar
 - ◆ YCbYCr 4:2:2
 - ◆ YCrYCb 4:2:2
 - ◆ CbYCrY 4:2:2
 - ◆ CrYCbY 4:2:2
 - ◆ Fully configurable ARGB channel lengths and locations inside 32bits, such as ARGB8888,RGB565,ARGB4444 etc.
- Input image size:
 - ◆ Combined mode: from 48x48 to 8176x8176 (66.8Mpixels)
 - ◆ Stand-alone mode: width from 48 to 8176, height from 48 to 8176, and maximum size limited to 16.7Mpixels
 - ◆ Step size is 16 pixels
- Output image size: from 16x16 to 1920x1088 (horizontal step size 8, vertical step size 2)
- Support image up-scaling:
 - ◆ Bicubic polynomial interpolation with a four-tap horizontal kernel and a two-tap vertical kernel
 - ◆ Arbitrary non-integer scaling ratio separately for both dimensions
 - ◆ Maximum output width is 3x input width
 - ◆ Maximum output height is 3x input height
- Support image down-scaling:
 - ◆ Arbitrary non-integer scaling ratio separately for both dimensions
 - ◆ Unlimited down-scaling ratio
- Support YUV to RGB color conversion, compatible with BT.601-5, BT.709 and user definable conversion coefficient
- Support dithering (2x2 ordered spatial dithering) for 4/5/6bit RGB channel precision
- Support programmable alpha channel and alpha blending operation with the following overlay input formats:
 - ◆ 8bit alpha +YUV444, big endian channel order with AYUV8888
 - ◆ 8bit alpha +24bit RGB, big endian channel order with ARGB8888
- Support deinterlacing with conditional spatial deinterlace filtering, only compatible with YUV420 input format
- Support RGB image contrast/brightness/color saturation adjustment
- Support image cropping & digital zoom only for JPEG or stand-alone mode
- Support picture in picture
- Support image rotation (horizontal flip, vertical flip, rotation 90, 180 or 270 degrees)

1.1.9 Image Enhancement(IEP module)

- Image formats support
 - Input data: XRGB/RGB565/YUV420/YUV422
 - Output data: ARGB/RGB565/YUV420/YUV422
 - ARGB/XRGB/RGB565/YUV swap
 - UV SP/P

- BT601_I/BT601_f/BT709_I/BT709_f color space conversion
- RGB dither up/down
- YUV up/down sampling
- Max source image resolution: 8192x8192
- Max scaled image resolution: 4096x4096
- YUV enhancement &denoise
 - Hue, Saturation, Brightness, Contrast adjustment
- RGB enhancement &denoise
 - Contrast enhancement
 - Color enhancement
 - Gamma adjustment
- High quality scale
 - Averaging filter down-scaling
 - Bi-cubic up-scaling
 - Arbitrary non-integer horizontal & vertical scaling ratio range from 1/16 to 16
- De-interlace
 - 3x5 Y motion detection matrix
 - Source width up to 1920
 - Configed high frequency de-interlace
 - I4O2 (Input 4 field,output 2 frame) /I4O1B/I4O1T/I2O1B/I2O1T mode
- Interface
 - Configed direct path to LCDC if source width no more than 1920
 - 32bit AHB bus slave
 - 64bit AXI bus master
 - Combined interrupt output

1.1.10 Graphics Engine

- 3D Graphics Engine :
 - High performance OpenGL ES1.1 and 2.0, OpenVG1.1 etc.
 - Embedded 4shader cores with shared hierarchical tiler
 - Separate vertex(geometry) and fragment(pixel) processing for maximum parallel throughput
 - Provide MMU and L2 Cache with 32KB size
 - Triangle rate : 30M triangles/s
 - Pixel rate: 300 pixels/s @ 150MHz
- 2D Graphics Engine(RGA module) :
 - Pixel rate: 300M pixel/s without scale, 150M pixel/s with bilinear scale, 66.5M pixel/s with bicubic scale.
 - Bit Blit with Strength Blit, Simple Blit and Filter Blit
 - Color fill with gradient fill, and pattern fill
 - Line drawing with anti-aliasing and specified width
 - High-performance stretch and shrink
 - Monochrome expansion for text rendering
 - ROP2, ROP3, ROP4 full alpha blending and transparency
 - Alpha blending modes including Java 2 Porter-Duff compositing blending rules , chroma key, and pattern mask
 - 8K x 8K raster 2D coordinate system
 - Arbitrary degrees rotation with anti-aliasing on every 2D primitive
 - Programmable bicubic filter to support image scaling

- Blending, scaling and rotation are supported in one pass for stretch blit
- Source formats :
 - ◆ ABGR8888, XBGR888, ARGB8888, XRGB888
 - ◆ RGB888, RGB565
 - ◆ RGBA5551, RGBA4444
 - ◆ YUV420 planar, YUV420 semi-planar
 - ◆ YUV422 planar, YUV422 semi-planar
 - ◆ BPP8, BPP4, BPP2, BPP1
- Destination formats :
 - ◆ ABGR8888, XBGR888, ARGB8888, XRGB888
 - ◆ RGB888, RGB565
 - ◆ RGBA5551, RGBA4444
 - ◆ YUV420 planar, YUV420 semi-planar only in filter and pre-scale mode
 - ◆ YUV422 planar, YUV422 semi-planar only in filter and pre-scale mode

1.1.11 Video IN/OUT

- Camera Interface
 - Support up to 5M pixels
 - 8bits CCIR656(PAL/NTSC) interface
 - 8bits raw data interface
 - YUV422 data input format with adjustable YUV sequence
 - YUV422,YUV420 output format with separately Y and UV space
 - Support picture in picture (PIP)
 - Support image crop with arbitrary windows
- Display Interface
 - Support displaying the same source on HDMI and PAD simultaneously (not support HDMI+vertical PAD)
 - Support LCD or TFT interfaces up to 1920x1080
 - Support HDMI 1.4 output up to 1080p@30fps
 - Parallel RGB LCD Interface :
 - RGB888(24bits),RGB666(18bits) ,RGB565(15bits)
 - Serial RGB LCD Interface: 2x12-bit, 3x8-bit(RGB delta support), 3x8-bit + dummy
 - MCU LCD interface: i-8080(up to 24-bit RGB), Hold/Auto/Bypass modes
 - TV Interface: ITU-R BT.656(8-bit, 480i/576i/1080i),TV encoder 10bit out for DAC, RGB888+1080i for HDMI, Parallel RGB HDMI interface:24-bit(RGB888 YCbCr444)
 - Max output resolution 1920x1080 for HDMI, 480i/576i for CVBS
 - 4 display layers :
 - ◆ One background layer with programmable 24bits color
 - ◆ One video layer (win0)
 - RGB888, ARGB888, RGB565, YCbCr422, YCbCr420, YCbCr444
 - maximum resolution is 1920x1080,support virtual display
 - 1/8 to 8 scaling up/down engine with arbitrary non-integer ratio
 - 256 level alpha blending(pre-multiplied alpha support)
 - Support transparency color key
 - De-flicker support for interlace output
 - Direct path support
 - YCbCr2RGB(rec601-mpeg/rec601-jpeg/rec709)
 - RGB2YCbCr(BT601/BT709)
 - ◆ One video layer (win1)

- RGB888, ARGB888, RGB565
- Support virtual display
- 256 level alpha blending (pre-multiplied alpha support)
- Support transparency color key
- Direct path support
- RGB2YCbCr(BT601/BT709)
- ◆ Hardware cursor(win3)
 - 8BPP (ARGB888 LUT)
 - Support two size: 32x32 and 64x64
 - 256 level alpha blending
 - Support hwc over panel at right and below side
- Win0 and Win1 layer overlay exchangeable
- 3 x 256 x 8 bits display LUTs
- Support replication(16bits to 24bits) and dithering(24bits to 16bits/18bits) operation
- Blank and blank display
- Scaler
 - ◆ Output for LVDS/RGB (max up to 1024x768)

1.1.12 HDMI

- HDMI version 1.4a, HDCP revision 1.4 and DVI version 1.0 compliant transmitter
- Supports DTV from 480i to 1080i/p HD resolution
- Supports 3D function defined in HDMI 1.4 spec
- Supports data rate from 25MHz, 1.65bps up to 3.4Gbps over a Single channel HDMI
- TMDS Tx Drivers with programmable output swing, resister values and pre-emphasis
- Digital video interface supports a pixel size of 24, 30, 36, 48bits color depth in RGB
- S/PDIF output supports PCM, Dolby Digital, DTS digital audio transmission (32-192kHz Fs) using IEC60958 and IEC 61937
- Multiphase 4MHz fixed bandwidth PLL with low jitter
- HDCP encryption and decryption engine contains all the necessary logic to encrypt the incoming audio and video data
- Support HDMI LipSync if needed as addon feature
- Lower power operation with optimal power management feature
- The EDID and CEC function are also supported by Innosilicon HDMI Transmitter Controller
- Optional Monitor Detection supported through Hot Plug

1.1.13 LVDS

- 135MHz clock support
- 28:4 data sub_channel compression at data rates up to 945 Mbps per channel
- Support VGA, SVGA, XGA and single pixel SXGA
- PLL requires no external components
- Comply with the Standard TIA/EIA-644-A LVDS standard
- Support alternative LVDS output or LVTTL output

1.1.14 MIPI DPHY

- Embedded 1 MIPI DPHY for TX

- Support 4 data lane
- Support 1080p @ 60fps output

1.1.15 Audio Interface

- I2S/PCM with 8ch
 - Up to 8 channels (4xTX, 2xRX)
 - Audio resolution from 16bits to 32bits
 - Sample rate up to 192KHz
 - Provides master and slave work mode, software configurable
 - Support 3 I2S formats (normal, left-justified, right-justified)
 - Support 4 PCM formats(early, late1, late2, late3)
 - I2S and PCM mode cannot be used at the same time
- I2S/PCM with 2ch
 - Up to 2 channels (2xTX, 2xRX)
 - Audio resolution from 16bits to 32bits
 - Sample rate up to 192KHz
 - Provides master and slave work mode, software configurable
 - Support 3 I2S formats (normal , left-justified , right-justified)
 - Support 4 PCM formats(early , late1 , late2 , late3)
 - I2S and PCM cannot be used at the same time
- SPDIF
 - Support two 16-bit audio data store together in one 32-bit wide location
 - Support biphase format stereo audio data output
 - Support 16 to 31 bit audio data left or right justified in 32-bit wide sample data buffer
 - Support 16, 20, 24 bits audio data transfer in linear PCM mode
 - Support non-linear PCM transfer
- Audio Codec
 - 18 to 24 bit High Order Sigma-Delta modulation for DAC for >93 dB SNR configurable
 - 16 to 18 bit High Order Sigma-Delta modulation for ADC for >90 dB SNR configurable
 - Digital interpolation and decimation filter integrated
 - Line-in, Microphone in and Speaker out Interface
 - On-Chip Analog Post Filter and digital filters
 - Single-ended or differential Input and Output
 - Sampling Rate of 8kHz/12kHz/16kHz/ 24kHz/32kHz /48kHz/44.1K/96KHz
 - Support 16ohm to 32ohm Head Phone and Speaker Phone Output
 - Mono, Stereochannel supported
 - Optional Fractional PLL available that support 6Mhz to 20Mhz clock input to any clockoutput that meets 8kHz/12kHz/16kHz/ 24kHz/32kHz /48kHz/44.1K/96KHz and 128 time oversampling ratio.

1.1.16 Connectivity

- SDIO interface
 - Compatible with SDIO 3.0 protocol
 - 4bits data bus widths
- High-speed ADC stream interface

- Support single-channel 8bits/10bits interface
- DMA-based and interrupt-based operation
- Support 8bits TS stream interface
- TS interface
 - Supports two TS input channels and one TS output channel.
 - Supports 4 TS Input Mode: sync/valid mode in the case of serial TS input; nosync/valid mode, sync/valid, sync/burst mode in the case of parallel TS input.
 - Supports serial and parallel output mode with PCR adjustment, and lsb-msb or msb-lsb bit ordering can be chosen in the serial output mode.
 - Supports 2 TS sources: demodulators and local memory.
 - Supports 2 Built-in PTIs(Programmable Transport Interface) to process TS simultaneously, and Each PTI supports:
 - ◆ 64 PID filters.
 - ◆ TS descrambling with 16 sets of Control Word under CSA v2.0 standard, up to 104Mbps
 - ◆ 16 PES/ES filters with PTS/DTS extraction and ES start code detection.
 - ◆ 4/8 PCR extraction channels
 - ◆ 64 Section filters with CRC check, and three interrupt mode: stop per unit, full-stop, recycle mode with version number check
 - ◆ PID done and error interrupts for each channel
 - ◆ PCR/DTS/PTS extraction interrupt for each channel
 - Supports 1 PVR(Personal Video Recording) output channel.
 - 1 built-in multi-channel DMA Controller.
- Smart Card
 - support card activation and deactivation
 - support cold/warm reset
 - support Answer to Reset (ATR) response reception
 - support T0 for asynchronous half-duplex character transmission
 - support T1 for asynchronous half-duplex block transmission
 - support automatic operating voltage class selection
 - support adjustable clock rate and bit (baud) rate
 - support configurable automatic byte repetition
- GPS Interface
 - Single chip, integrate GPS bb with cpu
 - 32 DMA channels for AHB master access
 - Complete 1-band, C/A, and NMEA-0183 compatibility
 - Support reference frequencies 16.368MHz
 - High sensitivity for indoor fixes
 - Low power consumption
 - Low cost with smaller size
 - Multi modes support both standalone GPS and A_GPS
- GMAC 10/100/1000M Ethernet Controller
 - Supports 10/100/1000-Mbps data transfer rates with the RGMII interfaces
 - Supports 10/100-Mbps data transfer rates with the RMII interfaces
 - Supports both full-duplex and half-duplex operation
 - ◆ Supports CSMA/CD Protocol for half-duplex operation

- ◆ Supports packet bursting and frame extension in 1000 Mbps half-duplex operation
 - ◆ Supports IEEE 802.3x flow control for full-duplex operation
 - ◆ Optional forwarding of received pause control frames to the user application in full-duplex operation
 - ◆ Back-pressure support for half-duplex operation
 - ◆ Automatic transmission of zero-quanta pause frame on deassertion of flow control input in full-duplex operation
 - Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
 - Automatic CRC and pad generation controllable on a per-frame basis
 - Options for Automatic Pad/CRC Stripping on receive frames
 - Programmable InterFrameGap (40-96 bit times in steps of 8)
 - Supports a variety of flexible address filtering modes
 - Separate 32-bit status returned for transmission and reception packets
 - Supports IEEE 802.1Q VLAN tag detection for reception frames
 - Support detection of LAN wake-up frames and AMD Magic Packet frames
 - Support checksum off-load for received IPv4 and TCP packets encapsulated by the Ethernet frame
 - Support checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams
 - Comprehensive status reporting for normal operation and transfers with errors
 - Automatic generation of PAUSE frame control or backpressure signal to the GMAC core based on Receive FIFO-fill (threshold configurable) level
 - Handles automatic retransmission of Collision frames for transmission
 - Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- SPI Controller
 - Support serial-master and serial-slave mode, software-configurable
 - DMA-based or interrupt-based operation
 - Embedded two 32x16bits FIFO for TX and RX operation respectively
 - Support 2 chip-selects output in serial-master mode
 - UartController
 - 3 on-chip uart controller inside RKaudi
 - DMA-based or interrupt-based operation
 - UART0 Embedded two 64Bytes FIFO for TX and RX operation respectively
 - UART1/UART2 Embedded two 32Bytes FIFO for TX and RX operation respectively
 - Support 5bit,6bit,7bit,8bit serial data transmit or receive
 - Standard asynchronous communication bits such as start,stop and parity
 - Support different input clock for uart operation to get up to 4Mbps or other special baud rate
 - Support non-integer clock divides for baud clock generation
 - Support auto flow control mode
 - I2C controller
 - 4 on-chip I2C controller in RKaudi
 - Multi-master I2C operation
 - Support 7bits and 10bits address mode

- Software programmable clock frequency and transfer rate up to 400Kbit/s in the fast mode
- Serial 8bits oriented and bidirectional data transfers can be made at up to 100Kbit/s in the standard mode
- GPIO
 - 4 groups of GPIO (GPIO0~GPIO3) , 32 GPIOs per group in GPIO0~GPIO3, totally have 128 GPIOs
 - All of GPIOs can be used to generate interrupt to Cortex-A9
 - All of pullup GPIOs are software-programmable for pullup resistor or not
 - All of pulldown GPIOs are software-programmable for pulldown resistor or not
 - All of GPIOs are always in input direction in default after power-on-reset
- USB Host2.0
 - Embedded 2 USB Host2.0 interfaces
 - Compatible with USB Host2.0 specification
 - Supports high-speed(480Mbps), full-speed(12Mbps) and low-speed(1.5Mbps) mode
 - Provides 16 host mode channels
 - Support periodic out channel in host mode
- USB OTG2.0
 - Compatible with USB OTG2.0 specification
 - Supports high-speed(480Mbps), full-speed(12Mbps) and low-speed(1.5Mbps) mode
 - Support up to 9 device mode endpoints in addition to control endpoint 0
 - Support up to 6 device mode IN endpoints including control endpoint 0
 - Endpoints 1/3/5/7 can be used only as data IN endpoint
 - Endpoints 2/4/6 can be used only as data OUT endpoint
 - Endpoints 8/9 can be used as data OUT and IN endpoint
 - Provides 9 host mode channels

1.1.17 Others

- SAR-ADC(Successive Approximation Register)
 - 3-channel single-ended 10-bit SAR analog-to-digital converter
 - Sample rate Fs is 200KHz
 - SAR-ADC clock must be large than 11*Fs, recommend is 11*Fs
 - DNL is less than ± 1 LSB , INL is less than ± 2.0 LSB
 - Power supply is 3.3V ($\pm 10\%$) for analog interface, power dissipation is less than 900uW
- eFuse
 - Two high-density electrical Fuse is integrated: 512bits (64x8)
 - Support standby mode
 - Programming condition : VP must be 2.5V($\pm 10\%$)
 - Program time is 2us.
 - Read condition : VP must be 0V or Floating.
 - Provide inactive mode, VP must be 0V or Floating in this mode.
- Operation Temperature Range
 - -40°C to +85°C
- Operation Voltage Range

- Core supply: 1.1V ($\pm 10\%$)
- IO supply : 3.3V or 2.5V or 1.8V ($\pm 10\%$)
- Process
 - SMIC40nm LL
- Package Type
 - BGA316 (body: 14mm x 14mm ; ball size : 0.3mm ; ball pitch : 0.65mm)
 - LQFP176L (body: 20mm x 20mm)
- Power
 - TBA

Notes : ^①: DDR3/LPDDR2/LPDDR3 are not used simultaneously as well as async and sync ddrnand flash

^②: In RKaudi, Video decoder and encoder are not used simultaneously because of shared internal buffer

^③: Actual maximum frame rate will depend on the clock frequency and system bus performance

^④: Actual maximum data rate will depend on the clock frequency and JPEG compression rate

1.2 Block Diagram

The following diagram shows the basic block diagram for RKaudi.

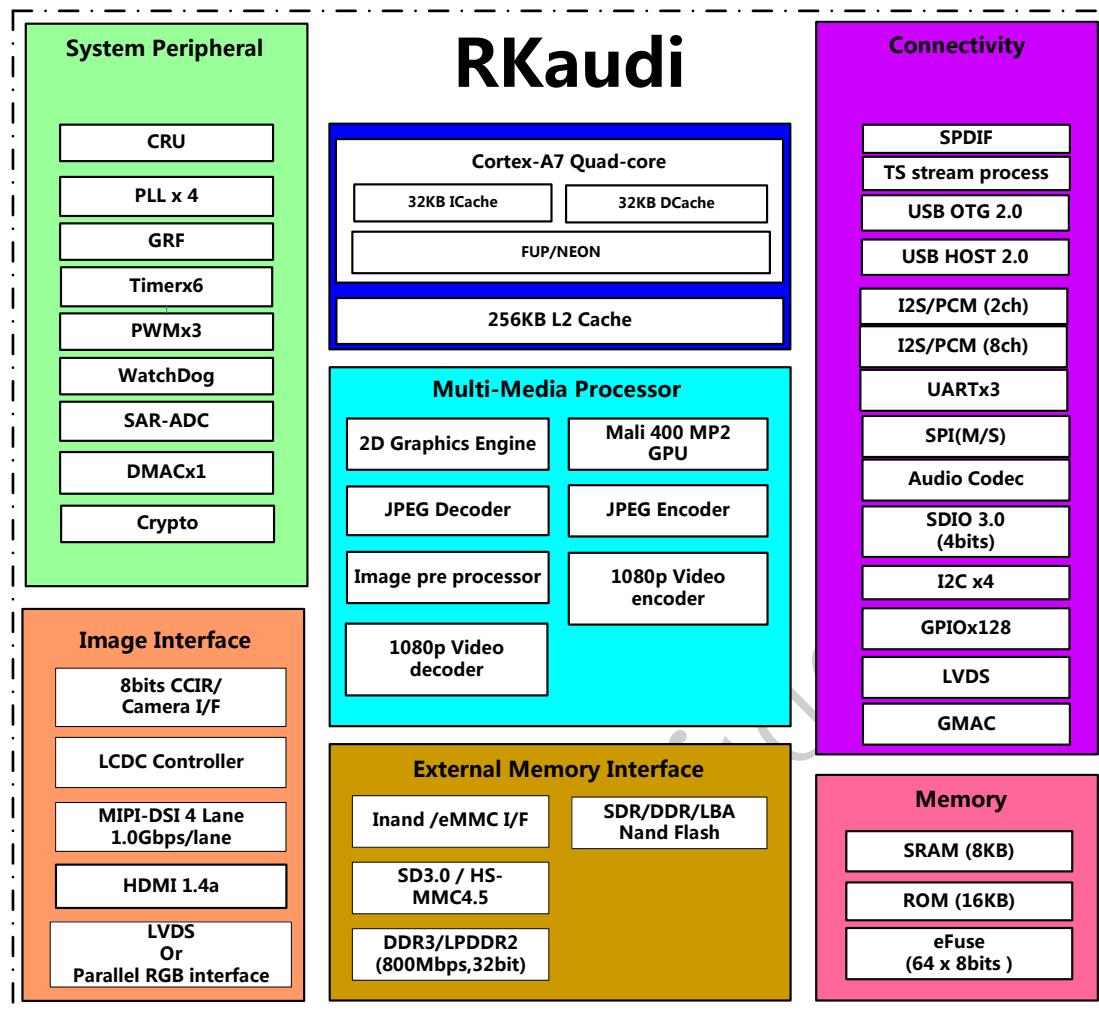


Fig.1-1RKaudi Block Diagram

1.3 Pin Description

In this chapter, the pin description will be divided into two parts, one is all power/ground descriptions in Table 1-1, include analog power/ground, another is all the function signals descriptions in Table 1-2, also include analog power/ground.

1.3.1 RK3128 power/ground IO descriptions

Table 1-1 RK3128 Power/Ground IO informations

Group	Ball #	Min(V)	Typ(V)	Max(V)	Descriptions
GND	B14, C3,C7,C10,C13, G3,G13, H8,H9,H10,H11,H12,H13, J8,J9,J10,J11,J12,J13, K3,K8,K9,K10,K11,K12,K13, L8,L9,L10,L11,L12,L13,L14 M2,M8,M9,M10,M11,M12,M13, N7,N8,N9,N10,N11,N12,N13,	N/A	N/A	N/A	Internal Core Ground and Digital IO Ground

	P7,P8,V3,W6,W9,W12,W15				
AVDD	P12,P13,P14,N14,M14	1.08	1.2	1.32	Internal CPU Power (@ cpu frequency <= 1GHz)
CVDD	G7,K7,P10,J14,H14,G10	1.08	1.2	1.32	Internal Core Power
VCCIO1	N6	3	3.3	3.6	Digital GPIO Power
VCCIO2	T14	3	3.3	3.6	Digital GPIO Power
VCCIO3	K14	3	3.3	3.6	Digital GPIO Power
VCCIO4	G14	3	3.3	3.6	Digital GPIO Power
DDR_VDD	H7,J7,L7,M7,G12,G11,G9,G8	1.4 N/A	1.5 1.35	1.6 N/A	DDR3 Digital IO Power LVDDR3 Digital IO Power
A/GPLL_DV DD11	N3	0.99	1.1	1.21	ARM PLL Analog Power
C/DPLL_DV DD11	N4	0.99	1.1	1.21	DDR PLL Analog Power
PLL_VCCIO	N5	3	3.3	3.6	DDR PLL Analog Power
SAR_AVDD3 3	P9	2.97	3.3	3.67	SAR-ADC Analog Power
USB_DVDD 11	T11	0.99	1.1	1.21	USB OTG2.0/Host 2.0 Digital Power
USB_AVDD 33	T10	2.97	3.3	3.63	USB OTG2.0/Host 2.0 Analog Power
CODEC_AV DD	D13	2.97	3.3	3.63	Audio Codec Analog Power

CODEC_AV_SS	D14				Audio Codec Analog Ground
HDMI_DVD_D1V1_1	V4	0.99	1.1	1.21	HDMI Digital Power
HDMI_AVD_D33	V2	3.0	3.3	3.6	HDMI Analog Power
LVDS/MIPI_VDD11	R8	0.99	1.1	1.21	MIPI/LVDS Digital Power
LVDS/MIPI_VCC1	R7	2.97	3.3	3.63	MIPI/LVDS Analog Power
VDAC_AVD_D	T7	2.97	3.3	3.63	VDAC Analog Power
VDAC_AGN_D	T8				VDAC Analog Ground

1.3.2 RK3128 function IO descriptions

Table 1-2 RK3128 IO descriptions

Ball Name	Ball #	func1	func2	func3	func4	pad ^① type	Driving ^②	Pull up/ do wn	Reset State ^③	power suppl y ^⑤
Left Side ^④										
DDR_A0	D1									
DDR_A2	E2									
DDR_A5	E5									
DDR_A9	E1									
DDR_A13	G1									
DDR_A7	E4									
DDR_ODT1	G2									
DDR_RESETN	F6									
DDR_DQ10	F2									
DDR_DQ8	G6									
DDR_DQS1	H1									
DDR_DQS1_N	H2									
DDR_DQ14	K2									
DDR_DQ12	H3									
DDR_DQ15	K1									
DDR_DQ13	J2									
DDR_DQ9	G5									
DDR_DM1	H4									
DDR_DQ11	H6									
DDR_DQ26	G4									
DDR_DQ24	K4									

DDR_DQS3	L1									
DDR_DQS3_N	L2									
DDR_DQ30	L5									
DDR_DQ28	H5									
DDR_DQ31	L6									
DDR_DQ29	K6									
DDR_DQ25	L4									
DDR_DM3	L3									
DDR_DQ27	K5									
XOUT24M	N1									
XIN24M	N2									
GPIO2_C6/LCDC_D20/EBC_BO RDER0/GPS_SIGN/GMAC_TXD2	P5									
GPIO2_C7/LCDC_D21/EBC_BO RDER1/GPS_MAG/GMAC_TXD3	P4									
GPIO2_C5/LCDC_D19/EBC_SD SHR/I2C2_SCL/GMAC_RXD2	P3									
GPIO2_C4/LCDC_D18/EBC_GD RL/I2C2_SDA/GMAC_RXD3	T4									
GPIO2_C3/LCDC_D17/EBC_GD PWR0/GMAC_TXD0	P2									
GPIO2_C2/LCDC_D16/EBC_GD SP/GMAC_TXD1	P1									
GPIO2_C1/LCDC_D15/EBC_GD OE/GMAC_RXD0	R2									
GPIO2_C0/LCDC_D14/EBC_VC OM/GMAC_RXD1	T3									
GPIO2_D1/LCDC_D23/EBC_GD PWR2/GMAC_MDC	T2									
GPIO2_D0/LCDC_D22/EBC_GD	P6									

PWR1/GPS_CLK/GMAC_COL										
GPIO2_B7/LCDC_D13/EBC_SD CE5/GMAC_RXER	T1									
GPIO2_B6/LCDC_D12/EBC_SD CE4/GMAC_CLK	U4									
GPIO2_B5/LCDC_D11/EBC_SD CE3/GMAC_TXEN	U2									
GPIO2_B4/LCDC_D10/EBC_SD CE2/GMAC_MDIO	U3									
GPIO2_B3/LCDC_DEN/EBC_GD CLK/GMAC_RXCLK	U1									
GPIO2_B2/LCDC_VSYNC/EBC_SDOE/GMAC_CRS	V5									
GPIO2_B1/LCDC_HSYNC/EBC_SDLE/GMAC_TXCLK	T5									
GPIO2_B0/LCDC_CLK/EBC_SD CLK/GMAC_RXDV	U5									
HDMI_EXTR	W3									
HDMI_TX3N	W1									
HDMI_TX3P	Y1									
HDMI_TX0N	W2									
HDMI_TX0P	Y2									
HDMI_TX1N	W4									
HDMI_TX1P	Y4									
HDMI_TX2N	W5									
HDMI_TX2P	Y5									
VDAC_IOUTN	U7									
VDAC_IOUTP	V7									
VDAC_IREF	U8									

LVDS/MIPI_EXTR	V8										
LCDC_D9/LVDS_CLKN/EBC_SD CE1/MIPI_CLKN	W7										
LCDC_D8/LVDS_CLKP/EBC_SD CE0/MIPI_CLKP	Y7										
LCDC_D7/LVDS_TX3N/EBC_SD DO7/MIPI_D3N	Y8										
LCDC_D6/LVDS_TX3P/EBC_SD DO6/MIPI_D3P	W8										
LCDC_D5/LVDS_TX2N/EBC_SD DO5/MIPI_D2N	W10										
LCDC_D4/LVDS_TX2P/EBC_SD DO4/MIPI_D2P	Y10										
LCDC_D3/LVDS_TX1N/EBC_SD DO3/MIPI_D1N	W11										
LCDC_D2/LVDS_TX1P/EBC_SD DO2/MIPI_D1P	Y11										
LCDC_D1/LVDS_TX0N/EBC_SD DO1/MIPI_D0N	W13										
LCDC_D0/LVDS_TX0P/EBC_SD DO0/MIPI_D0P	Y13										
USB1_DP	W14										
USB1_DM	Y14										
USB_EXTR	V11										
USB0_VBUS	U11										
USB0_ID	R11										
USB0_DM	Y16										
USB0_DP	W16										
ADCIN0	P11										
ADCIN1	U10										

ADCIN2	V10									
EFUSE	R10									
EFUSE	R10									
CIF_D0/TS_D0	U17									
CIF_D1/TS_D1	V17									
CIF_D2/TS_D2	W17									
CIF_D3/TS_D3	V16									
CIF_D4/TS_D4	T13									
CIF_D5/TS_D5	R13									
CIF_D6/TS_D6	R14									
CIF_D7/TS_D7	T16									
CIF_VSYNC/TS_SYNC	U13									
CIF_CLKI/TS_VALID	U16									
CIF_HREF/TS_FAIL	V18									
GPIO3_C1/DRIVE_VBUS/PMIC_SLEEP	U14									
CIF_CLKO/TS_CLKO	V13									
GPIO0_D2/PWM0	U18									
CIF_PDN1(GPIO3_B3)	Y20									
GPIO0_D3/PWM1	T17									
GPIO0_D4/PWM2	V14									
GPIO3_D2/IR	W18									
GPIO3_D3/SPDIF	Y19									
GPIO2_D2/CARD_RST/UART0_TX	T18									
GPIO2_D3/CARD_CLK/UART0_RX	Y17									
GPIO1_C1/SDMMC0_DET	W19									

GPIO2_D4											
GPIO2_D5/CARD_DET/UART0_CTSN	W20										
GPIO1_C6/FLASH_CS2/EMMC_CMD	U19										
GPIO2_A5/FLASH_WP/EMMC_PWR	V19										
GPIO1_C7/FLASH_CS3/EMMC_RST	P19										
GPIO1_D0/FLASH_D0/EMMC_D0/SFC_SIO0	P16										
GPIO1_D2/FLASH_D2/EMMC_D2/SFC_SIO2	T19										
GPIO1_D1/FLASH_D1/EMMC_D1/SFC_SIO1	U20										
GPIO1_D3/FLASH_D3/EMMC_D3/SFC_SIO3	T20										
GPIO1_D4/FLASH_D4/EMMC_D4/SPI_RXD	P18										
GPIO1_D6/FLASH_D6/EMMC_D6/SPI_CSN0	N15										
GPIO1_D5/FLASH_D5/EMMC_D5/SPI_TXD	R19										
GPIO1_D7/FLASH_D7/EMMC_D7/SPI_CSN1	P17										
GPIO2_A0/FLASH_ALE/SPI_CLK	N16										
GPIO2_A1/FLASH_CLE	N17										
GPIO2_A2/FLASH_WRN/SFC_CSNO	P20										

GPIO2_A3/FLASH_RDN/SFC_C SN1	L15										
GPIO2_A4/FLASH_RDY/EMMC_ CMD/SFC_CLK	K17										
GPIO2_A6/FLASH_CS0	L16										
GPIO2_A7/FLASH_DQS/EMMC_ CLKO	N19										
GPIO0_C7/FLASH_CS1	L17										
TEST	H15										
GPIO1_B6/SDMMC0_PWR	N18										
NPOR	N20										
GPIO1_C5/SDMMC0_D3/JTAG_ TMS	M19										
GPIO1_C4/SDMMC0_D2/JTAG_ TCK	K16										
GPIO1_C3/SDMMC0_D1/UART2_ RX	K15										
GPIO1_C2/SDMMC0_D0/UART2_ TX	L18										
GPIO1_A7/SDMMC0_WP	L19										
GPIO0_B6/I2S_SDI/SPI_CSN0	K19										
GPIO3_D7/CIF_PDN0/TEST_CL KO	K18										
GPIO0_B5/I2S_SDO/SPI_RXD	L20										
GPIO0_B4/I2S_LRCK_TX	H16										
GPIO0_B3/I2S_LRCK_RX/SPI_ TXD	J19										
GPIO0_B1/I2S_SCLK/SPI_CLK	K20										
GPIO0_B0/I2S_MCLK	H17										
GPIO1_C0/SDMMC0_CLKO	H20										

GPIO1_B3/UART1_RTSN/SPI_C_S0	G18										
GPIO1_B2/UART1_RX/SPI_RXD	H19										
GPIO1_B1/UART1_TX/SPI_TXD	H18										
GPIO3_C0											
GPIO1_B0/UART1_CTSN/SPI_C_LK	G19										
GPIO1_A5/I2S_SD1/SDMMC1_D3	G17										
GPIO1_A4/I2S_SDO/SDMMC1_D2	G16										
GPIO1_A3/I2S_LRCK_TX	G15										
GPIO1_A2/I2S_LRCK_RX/SDMMC1_D1	E19										
GPIO1_A1/I2S_SCLK/SDMMC1_D0/PMIC_SLEEP	E18										
GPIO1_A0/I2S_MCLK/SDMMC1_CLKO/XIN_32K	E20										
GPIO1_B7/SDMMC0_CMD	D18										
GPIO0_A3/I2C1_SDA/SDMMC1_CMD	D17										
GPIO0_A1/I2C0_SDA	E17										
GPIO0_A2/I2C1_SCL	F19										
GPIO0_A0/I2C0_SCL	D20										
GPIO0_C4/HDMI_CEC	C19										
GPIO0_B7/HDMI_HPD	E13										
GPIO0_A6/HDMI_SCL/I2C3_SC_L	B20										
GPIO0_A7/HDMI_SDA/I2C3_SDA	F14										

GPIO3_C7	G20								
GPIO3_C6	C18								
GPIO3_C5	F13								
GPIO3_C4	D19								
GPIO3_C3									
GPIO3_C2									
GPIO1_B4/SPI_CS1	B19								
GPIO0_D5									
GPIO0_D6/SDMMC1_PWR	A20								
GPIO0_D7									
GPIO0_D1/UART2_CTSN	A19								
GPIO0_D0/UART2_RTSN/PMIC_SLEEP	C17								
GPIO0_C6									
GPIO0_C3									
GPIO0_C2									
GPIO0_C1/CARD_IO/UART0_RT_SN	P15								
GPIO0_C0									
GPIO3_D6									
GPIO3_D5									
GPIO3_D4									
GPIO3_D1									
CODEC_MICL	E14								
CODEC_AIL	C16								
CODEC_VCM	A17								
CODEC_MICBIAS	B18								
CODEC_AIR	D16								

CODEC_MICR	E16										
CODEC_AOL	B17										
CODEC_AOMS	A16										
CODEC_AOM	B16										
CODEC_HPDET	C14										
CODEC_AOR	B15										
DDR_DQ18	B9										
DDR_DQ16	A10										
DDR_DQS2	A11										
DDR_DQS2_N	B11										
DDR_DQ22	F10										
DDR_DQ20	E10										
DDR_DQ23	A13										
DDR_DQ21	B12										
DDR_DQ17	C11										
DDR_DM2	D11										
DDR_DQ19	E11										
DDR_DQ2	F11										
DDR_DQ0	A14										
DDR_DQS0	A8										
DDR_DQS0_N	B8										
DDR_DQ6	B13										
DDR_DQ4	B10										
DDR_DQ7	C8										
DDR_DQ5	D10										
DDR_DQ1	F8										
DDR_DM0	B7										
DDR_DQ3	A7										
DDR_A8	D8										

DDR_A6	F7								
DDR_A14	B5								
DDR_A15	A5								
DDR_A11	C5								
DDR_A1	A4								
DDR_A4	A2								
DDR_A12	B4								
DDR_BA1	E8								
DDR_BA0	D7								
DDR_A10	C4								
DDR_CKE	B3								
DDR_ODT0	B2								
DDR_CLK_N	C2								
DDR_CLK	B1								
DDR_RASN	E3								
DDR_CASN	E7								
DDR_CSN1	B6								
DDR_CSN0	A1								
DDR_WEN	D5								
DDR_BA2	D3								
DDR_A3	D2								

Notes :

- ① : Pad types : I = input , O = output , I/O = input/output (bidirectional) ,
AP = Analog Power , AG = Analog Ground, DP = Digital Power , DG = Digital Ground, A = Analog
- ②: Output Drive Unit is mA , only Digital IO have drive value
- ③: Reset state : I = input without any pull resistor ,O = output without any pull resistor ,
- ④: It is die location. For examples, "Left side" means that all the related IOs are always in left side of die
- ⑤: Power supply means that all the related IOs is in these IO power domain. If multiple powers is included, they are connected together in one IO power ring

1.3.3 IO pin name descriptions

This sub-chapter will focus on the detailed function description of every pins based on different interface.

Table 1-3 RK3128 IO function description list

Interface	Pin Name	Direction	Description
Misc	XIN24M	I	Clock input of 24MHz crystal
	XOUT24M	O	Clock output of 24MHz crystal
	NPOR	I	Power on reset for chip

Interface	Pin Name	Direction	Description
Debug	TCK	I	JTAG interface clock input/SWD interface clock input
	TMS	I/O	JTAG interface TMS input/SWD interface data out

Interface	Pin Name	Direction	Description
SD/MMC Host Controller	sdmmc_clkout	O	sdmmc card clock.
	sdmmc_cmd	I/O	sdmmc card command output and reponse input.
	sdmmc_data <i>i</i> (<i>i</i> =0~3)	I/O	sdmmc card data input and output.
	sdmmc_detect_n	I	sdmmc card detect signal, a 0 represents presence of card.
	sdmmc_write_prt	I	sdmmc card write protect signal, a 1 represents write is protected.
	sdmmc_rstn_out	O	sdmmc card reset signal
	sdmmc_pwr_en	O	sdmmc card power-enable control signal

Interface	Pin Name	Direction	Description
SDIO Host Controller	sdio_clkout	O	sdio card clock.
	sdio_cmd	I/O	sdio card command output and reponse input.
	sdio_data <i>i</i> (<i>i</i> =0~3)	I/O	sdio card data input and output.
	sdio_detect_n	I	sdio card detect signal, a 0 represents presence of card.
	sdio_write_prt	I	sdio card write protect signal, a 1 represents write is protected.
	sdio_pwr_en	O	sdio card power-enable control signal
	sdio_int_n	O	sdio card interrupt indication
	sdio_backend	O	the back-end power supply for embedded device

Interface	Pin Name	Direction	Description
eMMC	emmc_clkout	O	emmc card clock.

Interface	emmc_cmd	I/O	emmc card command output and reponse input.
	emmc_data i ($i=0\sim 7$)	I/O	emmc card data input and output.
	emmc_pwr_en	O	emmc card power-enable control signal
	emmc_rstn_out	O	emmc card reset signal

Interface	Pin Name	Direction	Description
DMC	CLK	O	Active-high clock signal to the memory device.
	CLK_N	O	Active-low clock signal to the memory device.
	CKE	O	Active-high clock enable signal to the memory device
	CSNi ($i=0,1$)	O	Active-low chip select signal to the memory device. ATThere are two chip select.
	RASN	O	Active-low row address strobe to the memory device.
	CASN	O	Active-low column address strobe to the memory device.
	WEN	O	Active-low write enable strobe to the memory device.
	BAi($i=0,1,2$)	O	Bank address signal to the memory device.
	Ai($i=0\sim 15$)	O	Address signal to the memory device.
	DQi($i=0\sim 31$)	I/O	Bidirectional data line to the memory device.
	DQS0 DQS1 DQS2	I/O	Active-high bidirectional data strobes to the memory device.
	DQS0_N DQS1_N DQS2_N	I/O	Active-low bidirectional data strobes to the memory device.
	DMi($i=0\sim 3$)	O	Active-low data mask signal to the memory device.
	ODTi($i=0,1$)	O	On-Die Termination output signal for two chip select.
	RESETN	O	DDR3 reset signal to the memory device

Interface	Pin Name	Direction	Description
NandC	flash_wp	O	Flash write-protected signal
	flash_ale	O	Flash address latch enable signal
	flash_cle	O	Flash command latch enable signal
	flash_wrn	O	Flash write enable and clock signal
	flash_rdn	O	Flash read enable and write/read signal

	flash_data[i](i=0~7)	I/O	8bits of flash data inputs/outputs signal
	flash_dqs	I/O	Flash data strobe signal
	flash_rdy	I	Flash ready/busy signal
	flash_csn(i=0~3)	O	Flash chip enable signal for chip i, i=0~3

Interface	Pin Name	Direction	Description
I2S/PCM Controller (8 channel)	i2s_clk	O	I2S/PCM clock source
	i2s_sclk	I/O	I2S/PCM serial clock
	i2s_lrck_rx	I/O	I2S/PCM left & right channel signal for receiving serial data, synchronous left & right channel in I2S mode and the beginning of a group of left & right channels in PCM mode
	i2s_sdi	I	I2S/PCM serial data input
	i2s_sdo	O	I2S/PCM serial data output
	i2s_lrck_tx	I/O	I2S/PCM left & right channel signal for transmitting serial data, synchronous left & right channel in I2S mode and the beginning of a group of left & right channels in PCM mode

Interface	Pin Name	Direction	Description
SPI Controller	spi_clk	I/O	spi serial clock
	spi_csny (y=0,1)	I/O	spi chip select signal,low active
	spi_txd	O	spi serial data output
	spi_rxd	I	spi serial data input

Interface	Pin Name	Direction	Description
LCDC	lc当地_dclk	O	LCDC RGB interface display clock out, MCU i80 interface RS signal
	lc当地_vsync	O	LCDC RGB interface vertical sync pulse, MCU i80 interface CSN signal
	lc当地_hsync	O	LCDC RGB interface horizontal sync pulse, MCU i80 interface WEN signal
	lc当地_den	O	LCDC RGB interface data enable, MCU i80 interface REN signal
	lc当地_data[23:0]	I/O	LCDC data output/input

Interface	Pin Name	Direction	Description
Camera IF	cif_clkin	I	Camera interface input pixel clock
	cif_clkout	O	Camera interface output work clock

	cif_vsync	I	Camera interface vertical sync signal
	cif_href	I	Camera interface horizontal sync signal
	cif_data[7:0]	I	Camera interface 8-bit input pixel data

Interface	Pin Name	Direction	Description
GPS	gps_sign	I	GPS sign data input
	gps_mag	I	GPS mag data input
	gps_clk	I	GPS rf clock input

Interface	Pin Name	Direction	Description
PWM	Pwm2	O	Pulse Width Modulation output
	pwm1	O	Pulse Width Modulation output
	pwm0	O	Pulse Width Modulation output

Interface	Pin Name	Direction	Description
I2C	i2c0_sda	I/O	I2C0 data
	i2c0_scl	I/O	I2C0 clock
	i2c1_sda	I/O	I2C1 data
	i2c1_scl	I/O	I2C1 clock
	i2c2_sda	I/O	I2C2 data
	i2c2_scl	I/O	I2C2 clock
	i2c3_sda	I/O	I2C3 data
	i2c3_scl	I/O	I2C3 clock

Interface	Pin Name	Direction	Description
UART	uart0_sin	I	UART0 serial data input
	uart0_sout	O	UART0 serial data output
	uart0_cts_n	I	UART0 clear to send
	uart0_rts_n	O	UART0 request to send
	uart1_sin	I	UART1 serial data input
	uart1_sout	O	UART1 serial data output
	uart1_cts_n	O	UART1 clear to send
	uart1_rts_n	I	UART1 request to send
	uart2_sin	I	UART2 serial data input
	uart2_sout	O	UART2 serial data output

Interface	Pin Name	Direction	Description
USB OTG2.0 /HOST 2.0	USB0PP	I/O	USB OTG 2.0 Data signal DP
	USB0PN	I/O	USB OTG 2.0 Data signal DM
	VBUS_0	N/A	USB OTG 2.0 5V power supply pin
	USB0ID	I	USB OTG 2.0 ID indicator
	otg_drv_vbus	O	USB OTG 2.0 drive VBUS
	USB1PP	I/O	USB HOST 2.0 Data signal DP
	USB1PN	I/O	USB HOST 2.0 Data signal DM
	VBUS_1	N/A	USB HOST 2.0 5V power supply pin

USB1ID	I	USB HOST 2.0 ID indicator
USBRBIAS	N/A	45 Ohm Reference external resistance

Interface	Pin Name	Direction	Description
LVDS	Dn_m	I	Transmit parallel data in, n=1~4,m=0~6
	OEN	I	Output enable pin (Active Low)
	PDN	I	Transmitter power down enable pin(Active Low)
	CK_REF	I	Input clock
	DSn	I	Output loading selection, n=0~1
	PD_PLL	I	PLL Power down enable pin(Active High)
	PDN_CBG	I	CBG Power down enable pin (Active Low)
	XRES	I	Connected to external 12Kohm through bonding pad
	PADP_n	O	Transmit serial data out, n=1~4
	PADN_n	O	Transmit serial data out(Negative), n=1~4
	CLKP	O	Output clock
	CLKN	O	Output clock(Negative)
	Tn	I	TTL data input, n=1~10

Interface	Pin Name	Direction	Description
Audio Codec	MICL	I	Left channel microphone PGA positive input
	LINEL	I	Left channel line-in input
	VCM	I	Decoupling for voltage reference
	VREF_MIC	O	Microphone bias voltage output
	LINER	I	Right channel line-in input
	MICR	I	Right channel microphone PGA positive input
	VOUTL	O	Left channel DAC driver amplifier output
	VOUTR	O	Right channel DAC driver amplifier output
	AOMS	I	Headphone virtual ground feedback
	AOM	O	Headphone virtual ground output
	HPDET		Headphone jack detection

Interface	Pin Name	Direction	Description
HDMI	EXTR	O	Connect 2.0Kohm resistor to ground to generate reference current

TX3N	O	TMDS negative clock line
TX3P	O	TMDS positive clock line
TX0N	O	TMDS channel 0 negative data line
TX0P	O	TMDS channel 0 positive data line
TX1N	O	TMDS channel 1 negative data line
TX1P	O	TMDS channel 1 positive data line
TX2N	O	TMDS channel 2 negative data line
TX2P	O	TMDS channel 2 positive data line

Interface	Pin Name	Direction	Description
SAR-ADC	SARADC_AIN[i] (i=0~2)	N/A	SAR-ADC input signal for 3 channel

Interface	Pin Name	Direction	Description
eFuse	EFUSE_VP	N/A	eFuse program and sense power

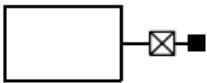
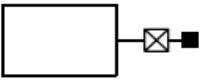
Interface	Pin Name	Direction	Description
MIPI	XRES	I	Connected to external 12Komh through bonding pad
	PADP_n	O	Transmit serial data out, n=1~4
	PADN_n	O	Transmit serial data out(Negative), n=1~4
	CLKP	O	Output clock
	CLKN	O	Output clock(Negative)

Interface	Pin Name	Direction	Description
MIPI	IOUT_N	O	Negative Output for DAC
	IOUT_P	O	Positive Output for DAC
	IREF	O	Reference Current. Output Current when using External Reference Resistor or Input Reference Current when using external current source

1.3.4 RK3128 IO Type

The following list shows IO type except DDR IO and all of Power/Ground IO .

Table 1-4 RK3128 IO Type List

Type	Diagram	Description	Pin Name
A		Analog IO Cell with IO voltage	EFUSE_VP
B		Dedicated Power supply to Internal Macro with IO voltage	SARADC_AIN[2:0]

C		Crystal Oscillator with internal register	XIN24M/XOUT24M
D		CMOS 3-state output pad with controllable input and controllable pulldown	Part of digital GPIO (PBCDxRNC)
E		CMOS 3-state output pad with controllable input and controllable pullup	Part of digital GPIO (PBCUxRNC)
F		controllable input pad with controllable pulldown	Part of digital GPIO (PICDRNC)
G		controllable input pad with controllable pullup	Part of digital GPIO (PICURNC)

1.4 Package information

RK3128 has two type of package
One is LQFP176(RK3126)

(body: 20mm x 20mm ; pin pitch : 0.4mm)

The other is TFBGA316(RK3128)

(body: 14mm x 14mm ; ball size : 0.3mm ; ball pitch : 0.65mm)

1.4.1 LQFP176 Dimension

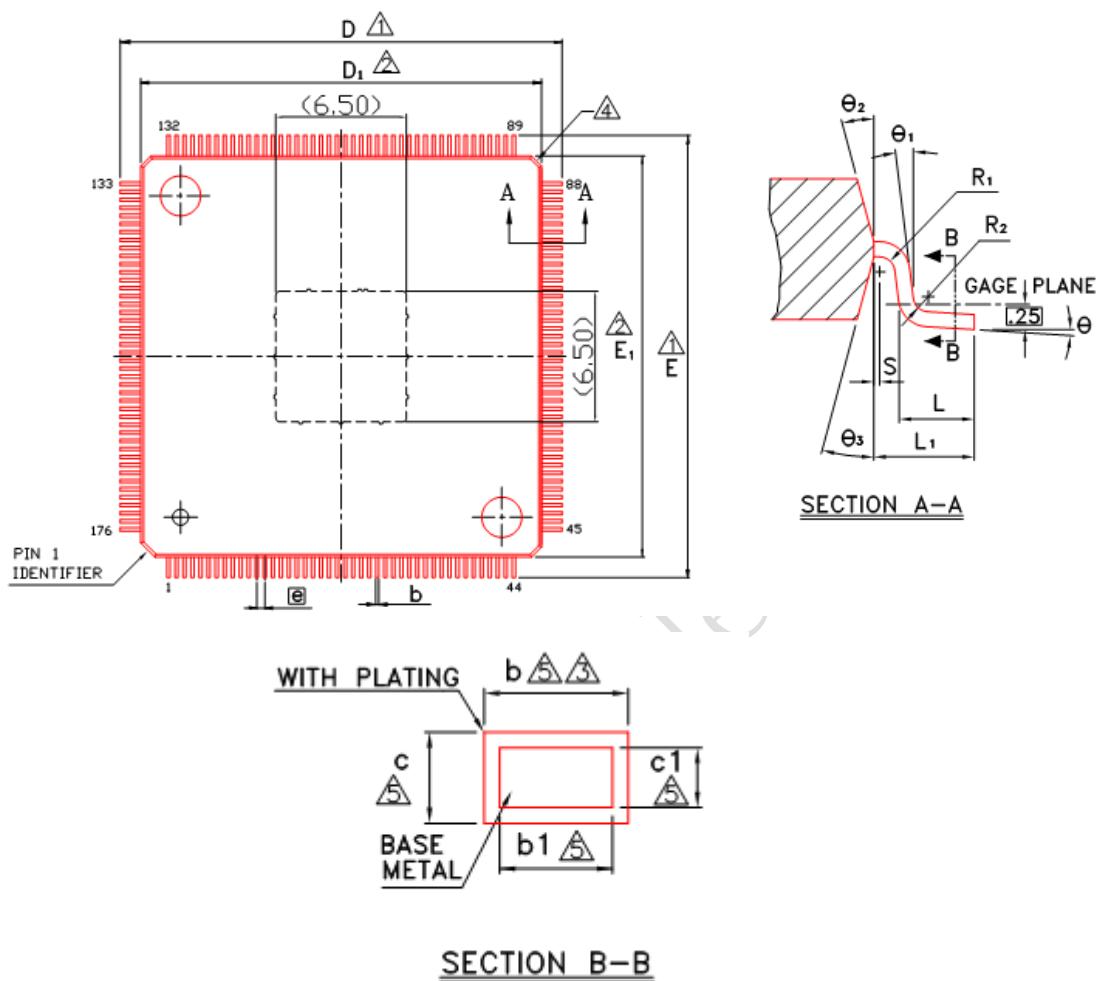


Fig.1-2RK3126 LQFP176 Package Top View



Fig.1-3RK3126 LQFP176 Package Side View

Symbol	Dimension in mm			Dimension in inch		
	Min	Nom	Max	Min	Nom	Max
A	—	—	1.60	—	—	0.063
A ₁	0.05	—	0.15	0.002	—	0.006
A ₂	1.35	1.40	1.45	0.053	0.055	0.057
b	0.13	0.18	0.23	0.005	0.007	0.009
b ₁	0.13	0.16	0.19	0.005	0.006	0.007
c	0.09	—	0.20	0.004	—	0.008
c ₁	0.09	0.12	0.16	0.004	0.005	0.006
D	21.60	22.00	22.40	0.850	0.866	0.882
D ₁	—	20.00	—	—	0.787	—
E	21.60	22.00	22.40	0.850	0.866	0.882
E ₁	—	20.00	—	—	0.787	—
e	0.40 BSC			0.016 BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
L ₁	1.00 REF			0.039 REF		
R ₁	0.08	—	—	0.003	—	—
R ₂	0.08	—	—	0.003	—	—
S	0.20	—	—	0.008	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ ₁	0°	—	—	0°	—	—
θ ₂	11°	12°	13°	11°	12°	13°
θ ₃	11°	12°	13°	11°	12°	13°
ccc	0.08			0.003		

Fig.1-4 RK3126 LQFP176 Dimension

1.4.2 TFBGA313 Dimension

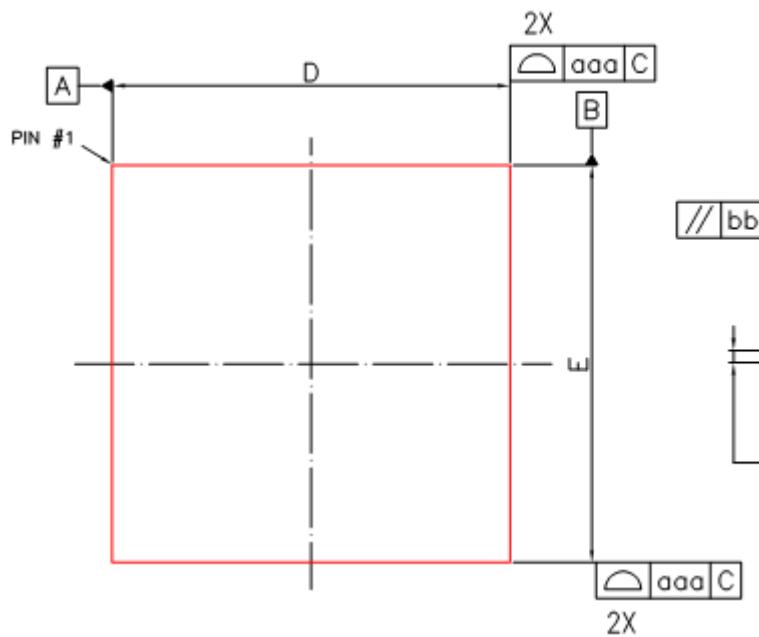


Fig.1-5RK3128 TFBGA316 Package Top View

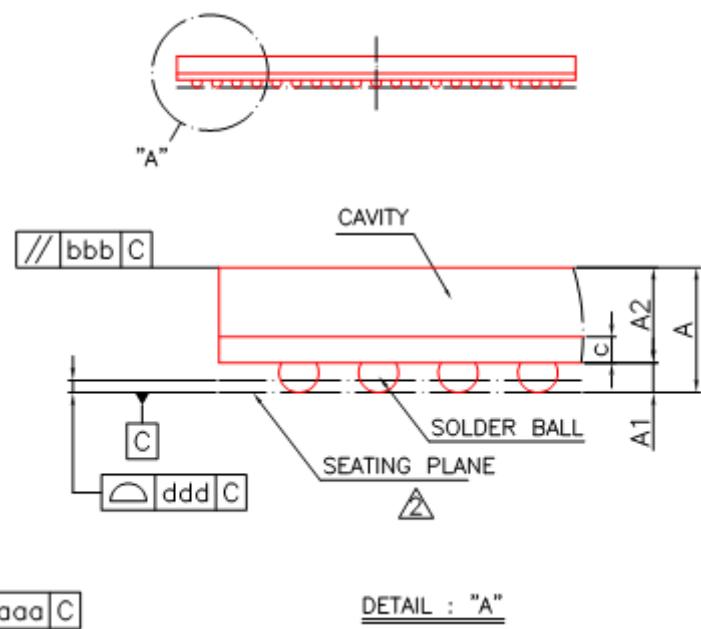


Fig.1-6RK3128 TFBGA316 Package Side View

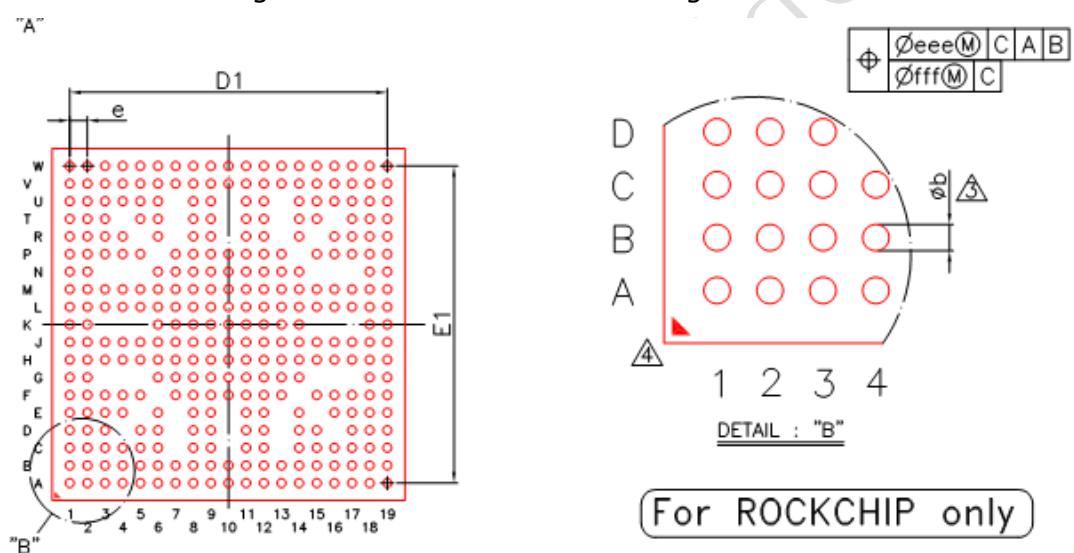


Fig.1-7RK3128 TFBGA316 Package Bottom View

symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	1.200	—	—	0.047
A1	0.160	0.210	0.260	0.006	0.008	0.010
A2	0.840	0.890	0.940	0.033	0.035	0.037
c	0.150	0.190	0.230	0.006	0.007	0.009
D	13.900	14.000	14.100	0.547	0.551	0.555
E	13.900	14.000	14.100	0.547	0.551	0.555
D1	—	12.350	—	—	0.486	—
E1	—	12.350	—	—	0.486	—
e	—	0.650	—	—	0.026	—
b	0.250	0.300	0.350	0.010	0.012	0.014
aaa		0.150			0.006	
bbb		0.200			0.008	
ddd		0.080			0.003	
eee		0.150			0.006	
fff		0.080			0.003	
N		316			316	
MD/ME		20/20			20/20	

Fig.1-8RK3128 TFBGA316 Package Dimension

1.4.3 LQFP176 Ball Map

Pin Number	Pin Name	Pin Number	
		DDR_VDD	176
1	DDR_A0	DDR_A3	175
2	DDR_A2	DDR_BA2	174
3	DDR_A5	DDR_WIEN	173
4	DDR_A9	DDR_CSNO	172
5	DDR_A13	DDR_CSNM	171
6	DDR_A7	DDR_CASN	170
7	DDR_ODT1	DDR_RASN	169
8	DDR_DQ10	DDR_VDD	168
9	DDR_DQ8	DDR_CLK	167
10	DDR_VDD	DDR_CLK_N	166
11	DDR_DQS1	DDR_ODT0	165
12	DDR_DDS1_N	DDR_CKE	164
13	CVDD	DDR_A10	163
14	DDR_DQ14	DDR_BA0	162
15	DDR_DQ12	DDR_BA1	161
16	DDR_DQ15	CVDD	160
17	DDR_DQ13	DDR_A12	159
18	DDR_VDD	DDR_A4	158
19	DDR_DQ9	DDR_A1	157
20	DDR_DM1	DDR_A11	156
21	DDR_DQ11	DDR_A15	155
22	C/DPLL_DVDD12	DDR_A14	154
		DDR_A6	153
		DDR_A8	152
		DDR_DQ3	151
		DDR_DM0	150
		DDR_DQ1	149
		DDR_VDD	148
		DDR_DQ6	147
		CVDD	143
		DDR_DQSO_N	142
		DDR_DQSO	141
		DDR_VDD	140
		DDR_DQ0	139
		DDR_DQ2	138
		CODEC_AOR	137
		CODEC_AN0	136
		CODEC_AOL	135
		CODEC_MIGR	134
		CODEC_YCM	133
		CVDD	132
		I2C0_SCL/GPIO0_A0	131
		I2C0_SDA/GPIO0_A1	130
		I2C1_SCL/GPIO0_A2	129
		I2C1_SDA/GPIO0_A3	128
		SDMMC0_CMD/GPIO1_B7	127
		VCCIO	126
		I2S_MCLK/GPIO1_A0	125
		I2S_SCLK/GPIO1_A1	124
		I2S_LRCK_RX/GPS_CLK/GPIO1_A2	123
		I2S_SDO/GPS_MAG/GPIO1_A4	122
		I2S_SD/GPS_SIGN/GPIO1_A5	121
		SPI_CLK/UART1_CTSN/GPIO1_B0	120
		SPI_TxD/UART1_TX/GPIO1_B1	119
		SPI_RxD/UART1_RX/GPIO1_B2	118
		SPI_CSNO/UART1_RTSN/GPIO1_B3	117
		SDMMC0_CLK0/GPIO1_C0	116
		VCCIO	115
		SDMMC0_DET/GPIO1_C1	114
		SDMMC0_D0/GPIO1_C2	113
		SDMMC0_D1/GPIO1_C3	112
		SDMMC0_D2/GPIO1_C4	111

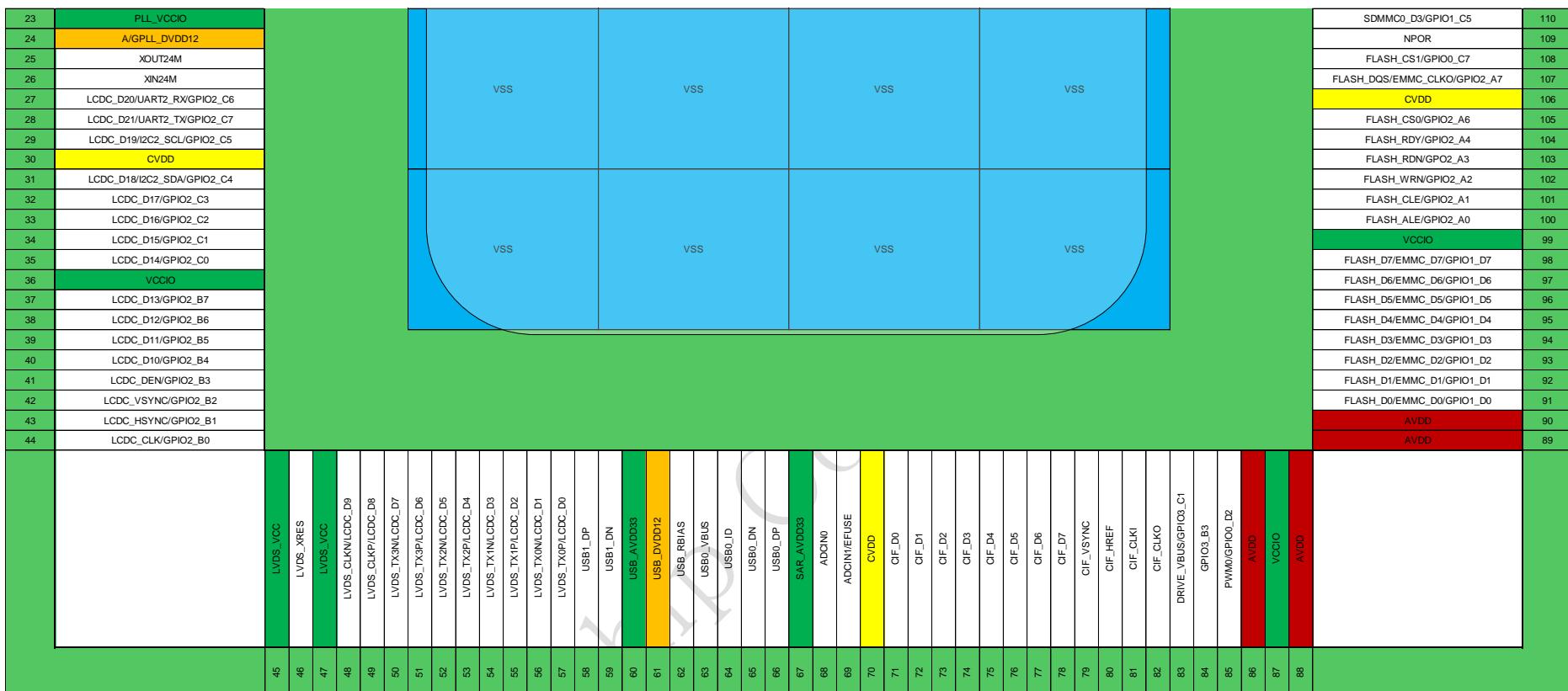


Fig.1-9LQFP176 Ball Map

1.4.4 TFBGA313 Ball Map

	316	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A	DDR_CSN_0	DDR_A4	NP	DDR_A1	DDR_A15	NP	DDR_DQ3	DDR_DQS_0	NP	DDR_DQ1_6	DDR_DQS_2	NP	DDR_DQ2_3	DDR_DQ0	NP	CODEC_A_OMS	CODEC_V_CM	NP	GPIO0_D1/UART2_CTSN	GPIO0_D6/SDMMC1_PWR	A	
B	DDR_CLK	DDR_ODT_0	DDR_CKE	DDR_A12	DDR_A14	DDR_CSN_1	DDR_DM0	DDR_DQS_0_N	DDR_DQ1_8	DDR_DQ4	DDR_DQS_2_N	DDR_DQ2_1	DDR_DQ6	VSS22	CODEC_A_OR	CODEC_A_OM	CODEC_MICBIAS	GPIO1_B4/SPI_CS1	GPIO0_A6/HDMI_SCL/I2C3_SCL	B		
C	NP	DDR_CLK_N	VSS54	DDR_A10	DDR_A11	NP	VSS1	DDR_DQ7	NP	VSS2	DDR_DQ1_7	NP	VSS7	CODEC_H_PDET	NP	CODEC_AI_L	GPIO0_D0/UART2_RT_SN/PMIC_SLEEP	GPIO3_C6	GPIO0_C4/HDMI_CEC	NP	C	
D	DDR_A0	DDR_A3	DDR_BA2	NP	DDR_WEN	NP	DDR_BA0	DDR_A8	NP	DDR_DQ5	DDR_DM2	NP	CODEC_A_VDD	CODEC_A_VSS	NP	CODEC_AI_R	GPIO0_A3/I2C1_SDA/SDMMC1_CMD	GPIO1_B7/SDMMC0_CMD	GPIO3_C4	GPIO0_A0/I2C0_SCL	D	
E	DDR_A9	DDR_A2	DDR_RAS_N	DDR_A7	DDR_A5	NP	DDR_CAS_N	DDR_BA1	NP	DDR_DQ1_9	NP	GPIO0_B7/HDMI_HPD	CODEC_MICL	NP	CODEC_MICR	GPIO0_A1/I2C0_SDA	GPIO1_A1/I2S_SCLK/SDMMC1_D0/PMIC_S	GPIO0_A2/I2S_LRCK_RX/SDMM_C1_D1	GPIO1_A0/I2S_MCLK/SDMMC1_CLKO/XIN_	E		
F	NP	DDR_DQ1_0	NP	NP	NP	DDR_RES_ETN	DDR_A6	DDR_DQ1	NP	DDR_DQ2_2	DDR_DQ2	NP	GPIO3_C5	GPIO0_A7/HDMI_SDA/I2C3_SDA	NP	NP	NP	NP	NP	F		
G	DDR_A13	DDR_ODT_1	VSS3	DDR_DQ2_6	DDR_DQ9	DDR_DQ8	CVDD1	DDR_VDD_5	DDR_VDD_6	CVDD6	DDR_VDD_7	DDR_VDD_8	VSS6	VCCIO4	GPIO1_A3/I2S_LRCK_TX	GPIO1_A4/I2S_SDO/S_DMMC1_D2	GPIO1_A5/I2S_SD/S_DMMC1_D3	GPIO1_B3/UART1_RT_SN/SPI_CL_K	GPIO1_B0/UART1_CT_SN/SPI_CLK	GPIO3_C7	G	
H	DDR_DQS_1	DDR_DQS_1_N	DDR_DQ1_2	DDR_DM1	DDR_DQ2_8	DDR_DQ1_1	DDR_VDD_4	VSS15	VSS16	VSS17	VSS18	VSS19	VSS20	CVDD5	TEST	GPIO0_B4/I2S_LRCK_TX	GPIO0_B0/I2S_MCLK	GPIO1_B1/UART1_TX/SPI_TXD	GPIO1_B2/UART1_RX/SPI_RXD	GPIO1_C0/SDMMC0_CLKO	H	

J	NP	DDR_DQ1_3	NP	NP	NP	NP	DDR_VDD_3	VSS23	VSS24	VSS25	VSS26	VSS27	VSS28	CVDD4	NP	NP	NP	NP	GPIO0_B3/I2S_LRCK_RX/SPI_TX_D	NP	J	
K	DDR_DQ1_5	DDR_DQ1_4	VSS4	DDR_DQ2_4	DDR_DQ2_7	DDR_DQ2_9	CVDD2	VSS21	VSS29	VSS30	VSS46	VSS31	VSS32	VCCIO3	GPIO1_C3/SDMMC0_D1/UART2_RX	GPIO1_C4/SDMMC0_D2/JTAG_TCK	GPIO2_A4/FLASH_RDY/EMMC_CSD/TEST_CLK_O	GPIO3_D7/CIF_PDN0/TEST_CLK_O	GPIO0_B6/I2S_SDIN/PI_CSNO	GPIO0_B1/I2S_SCLK/SPI_CLK	K	
L	DDR_DQS_3	DDR_DQS_3_N	DDR_DM3	DDR_DQ2_5	DDR_DQ3_0	DDR_DQ3_1	DDR_VDD_2	VSS33	VSS34	VSS35	VSS36	VSS37	VSS38	VSS39	GPIO2_A3/FLASH_RD_N/SFC_CS_N1	GPIO2_A6/FLASH_CS0	GPIO0_C7/FLASH_CS1	GPIO1_C2/SDMMC0_D0/UART2_TX	GPIO1_A7/SDMMC0_WP	GPIO1_B5/SDMMC0_D3/JTAG_TMS	L	
M	NP	VSS5	NP	NP	NP	NP	DDR_VDD_1	VSS40	VSS41	VSS42	VSS43	VSS44	VSS45	AVDD5	NP	NP	NP	NP	GPIO1_C5/SDMMC0_D3/JTAG_TMS	NP	M	
N	XOUT24M	XIN24M	A/GPLL_DV_DD11	C/DPLL_DV_DD11	PLL_VCC10	VCCIO1	VSS47	VSS48	VSS49	VSS50	VSS51	VSS52	VSS53	AVDD4	GPIO1_D6/FLASH_D6/E/EMMC_D6_E/SPI_CSN0	GPIO2_A0/FLASH_AL_E/SPI_CLK	GPIO2_A1/FLASH_CL_E	GPIO1_B6/SDMMC0_PWR	GPIO2_A7/FLASH_DQ_S/EMMC_C_LKO	NPOR	N	
P	GPIO2_C2/LCDC_D16/WR0/GMA_P/GMAC_T	GPIO2_C3/LCDC_D17/EBC_GDS/WR0/GMA	GPIO2_C5/LCDC_D19/EBC_GDP/HR/IC2C_S	GPIO2_C7/LCDC_D21/EBC_SDS/DER1/GPS	GPIO2_C6/LCDC_D20/EBC_BOR/DERO/GPS	GPIO2_D0/LCDC_D22/EBC_GDP/WR1/GPS	VSS13	VSS8	SAR_AVD_D33	CVDD3	ADCIN0	AVDD1	AVDD2	AVDD3	GPIO1_D0/FLASH_D0/EMMC_D0_SFC_SIO0	GPIO1_D0/FLASH_D0/EMMC_D7_SFC_SIO1	GPIO1_D0/FLASH_D0/EMMC_D4_SFC_SIO2	GPIO1_D7/FLASH_D7/EMMC_D0_SFC_SIO0	GPIO1_D4/FLASH_D4/EMMC_D7_SFC_SIO1	GPIO1_C7/FLASH_W3/EMMC_R_ST_SFC_SIO2	P	
R	GPIO2_C1/LCDC_D15/EBC_GDO/E/GMAC_R	NP	NP	NP	NP	NP	LVDS/MIPI_VCC1	LVDS/MIPI_VDD11	NP	EFUSE	USB0_ID	NP	CIF_D5_TS_D5	CIF_D6_TS_D6	NP	NP	NP	NP	NP	GPIO1_D5/FLASH_D5/EMMC_D5_SPL_TXD	NP	R
T	GPIO2_B7/LCDC_D13/EBC_SDC_E5/GMAC	GPIO2_D1/LCDC_D23/WR2/GMA	GPIO2_C0/LCDC_D23/EBC_GDP_M/GMAC	GPIO2_C4/LCDC_D14/EBC_VCO_M/GMAC	GPIO2_B1/LCDC_HS_YNC/EBC_L2/IC2C_SD	GPIO2_B0/LCDC_D18/YNC/EBC_SDLE/GMA	NP	VDAC_AV_DD	VDAC_AG_ND	NP	USB_AVDD33	USB_DVD_D11	NP	CIF_D4_TS_D4	VCCIO2	NP	CIF_D7_TS_D7	GPIO0_D3/CARD_RS_T/UART0_TX	GPIO2_D2/FLASH_D2/EMMC_D3_SFC_SIO2	GPIO1_D2/FLASH_D2/EMMC_D2_SFC_SIO3	GPIO1_D3/FLASH_D3/EMMC_D2_SFC_SIO3	T
U	GPIO2_B3/LCDC_DE/N/EBG_GD_E3/GMAC	GPIO2_B5/LCDC_D11/EBC_SDC_E3/GMAC	GPIO2_B4/LCDC_D10/EBC_SDC_E2/GMAC	GPIO2_B6/LCDC_D12/EBC_SDC_E4/GMAC	GPIO2_B0/LCDC_CLK_LK/GMAC	NP	VDAC_IOU_TN	VDAC_IREF	NP	ADCIN1	USB0_VBUS	NP	CIF_VSYN_C_TS_SYN_C	GPIO3_C1/DRIVE_VB_US/PMIC_SLEEP	NP	CIF_CLK_TS_VALID	CIF_D0_TS_D0	GPIO0_D2/PWM0	GPIO1_C6/FLASH_CS2/EMMC_C1_M	GPIO1_D1/FLASH_D1/EMMC_D1_SFC_SIO1	GPIO1_D3/FLASH_D3/EMMC_D1_SFC_SIO1	U
V	NP	HDMI_AVD_D33	VSS14	HDMI_DVD_D1V1_1	GPIO2_B2/LCDC_VSY_N/EBG_S_DOE/GMA	NP	VDAC_IOU_TP	LVDS/MIPI_EXTR	NP	ADCIN2	USB_EXTR	NP	CIF_CLK_TS_CLKO	GPIO0_D4/PWM2	NP	CIF_D3_TS_D3	CIF_D1_TS_D1	CIF_HREF_TS_FAIL	GPIO2_A5/FLASH_W_P/EMMC_P	NP	WR	V
W	HDMI_TX3_N	HDMI_TX0_N	HDMI_EXT_R	HDMI_TX1_N	HDMI_TX2_N	VSS9	LCDC_D9/LVDS_CLK_N/EBG_SD_CE1/MIPI_D06/MIPI	LCDC_D6/LVDS_CLK_N/EBG_SD_D05/MIPI	VSS10	LCDC_D5/LVDS_CLK_N/EBG_SD_D03/MIPI	LCDC_D3/LVDS_CLK_N/EBG_SD_D01/MIPI	VSS12	LCDC_D1/LVDS_CLK_N/EBG_SD_D00/MIPI	USB1_DP	VSS11	USB0_DP	CIF_D2_TS_D2	GPIO3_D2/IR	GPIO1_C1/SDMMC0_DET	GPIO1_D5/CARD_DE_T/UART0_CTSN	W	
Y	HDMI_TX3_P	HDMI_TX0_P	NP	HDMI_TX1_P	HDMI_TX2_P	NP	LCDC_D8/LVDS_CLK_P/EBG_SD_CE0/MIPI_D07/MIPI	LCDC_D7/LVDS_CLK_P/EBG_SD_D04/MIPI	NP	LCDC_D4/LVDS_CLK_P/EBG_SD_D02/MIPI	LCDC_D2/LVDS_CLK_P/EBG_SD_D00/MIPI	NP	LCDC_D0/LVDS_CLK_P/EBG_SD_D00/MIPI	USB1_DM	NP	USB0_DM	GPIO2_D3/CARD_CLK/UART0_RX	GPIO3_D3/SPDIF	GPIO3_D3/CIF_PDN1/GPIO3_B3	Y		
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20																						

Fig.1-10TFBGA316 Ball Map

1.4.5 LQFP176 Ball Pin Number Order(Tablet)

Table 1-5 RK3126 LQFP176 Pin Number Order Information

Pin Index	Pin name	Pin Index	Pin name
1	DDR_A0	89	AVDD3
2	DDR_A2	89	AVDD3
3	DDR_A5	89	AVDD3
4	DDR_A9	89	AVDD3
5	DDR_A13	90	AVDD4
6	DDR_A7	90	AVDD4
7	DDR_ODT1	90	AVDD4
8	DDR_DQ10	90	AVDD4
9	DDR_DQ8	91	AVDD5
10	DDR_VDD	91	AVDD5
10	DDR_VDD	91	AVDD5
10	DDR_VDD	92	GPIO1_C6/FLASH_CS2/ EMMC_CMD
11	DDR_DQS1	93	GPIO1_D0/FLASH_D0/E MMC_D0/SFC_SIO0
12	DDR_DQS1_N	95	GPIO1_D2/FLASH_D2/E MMC_D2/SFC_SIO2
13	CVDD	94	GPIO1_D1/FLASH_D1/E MMC_D1/SFC_SIO1
13	CVDD	96	GPIO1_D3/FLASH_D3/E MMC_D3/SFC_SIO3
13	CVDD	97	GPIO1_D4/FLASH_D4/E MMC_D4/SPI_RXD
14	DDR_DQ14	99	GPIO1_D6/FLASH_D6/E MMC_D6/SPI_CSN0
15	DDR_DQ12	98	GPIO1_D5/FLASH_D5/E MMC_D5/SPI_TXD
16	DDR_DQ15	100	GPIO1_D7/FLASH_D7/E MMC_D7/SPI_CSN1
17	DDR_DQ13	101	VCCIO3
18	DDR_VDD	101	VCCIO3
18	DDR_VDD	101	VCCIO3
18	DDR_VDD	101	VCCIO3
19	DDR_DQ9	102	GPIO2_A0/FLASH_ALE/ SPI_CLK
20	DDR_DM1	103	GPIO2_A1/FLASH_CLE
21	DDR_DQ11	104	GPIO2_A2/FLASH_WRN/ SFC_CSN0
22	C/DPLL_DVDD12	105	GPIO2_A3/FLASH_RDN/ SFC_CSN1
22	C/DPLL_DVDD12	106	GPIO2_A4/FLASH_RDY/ EMMC_CMD/SFC_CLK
23	PLL_VCCIO	107	GPIO2_A6/FLASH_CS0
23	PLL_VCCIO	108	CVDD4
23	PLL_VCCIO	109	GPIO2_A7/FLASH_DQS/ EMMC_CLKO

23	PLL_VCCIO	110	NPOR
24	A/GPLL_DVDD12	111	GPIO1_C5/SDMMC0_D3/JTAG_TMS
24	A/GPLL_DVDD12	112	GPIO1_C4/SDMMC0_D2/JTAG_TCK
25	XOUT24M	113	GPIO1_C3/SDMMC0_D1/UART2_RX
26	XIN24M	114	GPIO1_C2/SDMMC0_D0/UART2_TX
27	XVSS	115	VCCIO4
28	GPIO2_C5/LCDC_D19/EBC_SDSHR/I2C2_SCL/GMAC_RXD2	115	VCCIO4
29	CVDD2	115	VCCIO4
29	CVDD2	115	VCCIO4
29	CVDD2	116	GPIO1_C0/SDMMC0_CLKO
29	CVDD2	117	GPIO1_B3/UART1_RTSN/SPI_CSNO
30	GPIO2_C4/LCDC_D18/EBC_GDRL/I2C2_SDA/GMAC_RXD3	118	GPIO1_B2/UART1_RX/SPI_RXD
31	GPIO2_C3/LCDC_D17/EBC_GDPWR0/GMAC_TXD0	119	GPIO1_B1/UART1_TX/SPI_TXD
32	GPIO2_C2/LCDC_D16/EBC_GDSP/GMAC_RXD1	120	GPIO1_B0/UART1_CTSN/SPI_CLK
33	GPIO2_C1/LCDC_D15/EBC_GDOE/GMAC_RXD0	121	GPIO1_A5/I2S_SDI/SDMMC1_D3
34	GPIO2_C0/LCDC_D14/EBC_VCOM/GMAC_RXD1	122	GPIO1_A4/I2S_SDO/SDMMC1_D2
35	VCCIO1	123	GPIO1_A2/I2S_LRCK_RX/SDMMC1_D1
35	VCCIO1	124	GPIO1_A1/I2S_SCLK/SDMMC1_D0/PMIC_SLEEP
35	VCCIO1	125	GPIO1_A0/I2S_MCLK/SDMMC1_CLKO/XIN_32K
35	VCCIO1	126	CVDD5
36	GPIO2_B7/LCDC_D13/EBC_SDCE5/GMAC_RXER	126	CVDD5
37	GPIO2_B6/LCDC_D12/EBC_SDCE4/GMAC_CLK	127	GPIO1_B7/SDMMC0_CMD
38	GPIO2_B5/LCDC_D11/EBC_SDCE3/GMAC_TXEN	128	GPIO0_A3/I2C1_SDA/SDMMC1_CMD

39	GPIO2_B4/LCDC_D10/ EBC_SDCE2/GMAC_MD IO	130	GPIO0_A1/I2C0_SDA
40	GPIO2_B3/LCDC_DEN/ EBC_GDCLK/GMAC_RX CLK	129	GPIO0_A2/I2C1_SCL
41	GPIO2_B2/LCDC_VSYN C/EBC_SDOE/GMAC_C RS	131	GPIO0_A0/I2C0_SCL
42	GPIO2_B1/LCDC_HSYN C/EBC_SDLE/GMAC_TX CLK	132	CODEC_VCM
43	GPIO2_B0/LCDC_CLK/ EBC_SDCLK/GMAC_RX DV	133	CODEC_MICR
44	LVDS/MIPI_VCC1	134	CODEC_AVSS1
45	LVDS/MIPI_EXTR	134	CODEC_AVSS1
46	LVDS/MIPI_VCC2	134	CODEC_AVSS1
46	LVDS/MIPI_VCC2	135	CODEC AOL
46	LVDS/MIPI_VCC2	136	CODEC_AVDD
47	LCDC_D9/LVDS_CLKN/ EBC_SDCE1/MIPI_CLK N	136	CODEC_AVDD
48	LCDC_D8/LVDS_CLKP/ EBC_SDCE0/MIPI_CLK P	136	CODEC_AVDD
49	LCDC_D7/LVDS_TX3N/ EBC_SDDO7/MIPI_D3N	137	CODEC_AOR
50	LCDC_D6/LVDS_TX3P/ EBC_SDDO6/MIPI_D3P	138	CODEC_AVSS2
51	LCDC_D5/LVDS_TX2N/ EBC_SDDO5/MIPI_D2N	139	VSS
52	LCDC_D4/LVDS_TX2P/ EBC_SDDO4/MIPI_D2P	140	DDR_DQ2
53	LCDC_D3/LVDS_TX1N/ EBC_SDDO3/MIPI_D1N	141	DDR_DQ0
54	LCDC_D2/LVDS_TX1P/ EBC_SDDO2/MIPI_D1P	142	DDR_VDD3
55	LCDC_D1/LVDS_TX0N/ EBC_SDDO1/MIPI_D0N	142	DDR_VDD3
56	LCDC_D0/LVDS_TX0P/ EBC_SDDO0/MIPI_D0P	142	DDR_VDD3
57	USB1_DP	143	DDR_DQS0
58	USB1_DM	144	DDR_DQS0_N
59	USB_AVDD33	145	DDR_DQ6
60	USB_DVDD11	146	DDR_DQ4
60	USB_DVDD11	147	DDR_DQ7
61	USB_EXTR	148	DDR_DQ5
62	USB0_VBUS	149	DDR_VDD4
63	USB0_ID	149	DDR_VDD4
64	USB0_DM	149	DDR_VDD4

65	USB0_DP	150	DDR_DQ1
66	SAR_AVDD33	151	DDR_DM0
66	SAR_AVDD33	152	DDR_DQ3
66	SAR_AVDD33	153	DDR_A8
67	ADCIN0	154	DDR_A6
68	ADCIN2	155	DDR_A14
68	EFUSE	156	DDR_A15
68	EFUSE	157	DDR_A11
69	ADCIN6	158	DDR_A1
69	CVDD3	159	DDR_A4
69	CVDD3	160	DDR_A12
69	CVDD3	161	CVDD6
70	CIF_D0/TS_D0	161	CVDD6
71	CIF_D1/TS_D1	161	CVDD6
72	CIF_D2/TS_D2	161	CVDD6
73	CIF_D3/TS_D3	162	DDR_BA1
74	CIF_D4/TS_D4	163	DDR_BA0
75	CIF_D5/TS_D5	164	DDR_A10
76	CIF_D6/TS_D6	165	DDR_CKE
77	CIF_D7/TS_D7	166	DDR_ODT0
78	CIF_VSYNC/TS_SYNC	167	DDR_CLK_N
80	CIF_CLKI/TS_VALID	168	DDR_CLK
79	CIF_HREF/TS_FAIL	169	DDR_VDD6
82	GPIO3_C1/DRIVE_VBU S/PMIC_SLEEP	169	DDR_VDD6
81	CIF_CLKO/TS_CLKO	169	DDR_VDD6
84	GPIO0_D2/PWM0	170	DDR_RASN
83	CIF_PDN1/GPIO3_B3	171	DDR_CASN
85	GPIO0_D3/PWM1	172	DDR_CSN1
86	AVDD1	173	DDR_CSN0
86	AVDD1	174	DDR_WEN
86	AVDD1	175	DDR_BA2
86	AVDD1	176	DDR_A3
87	VCCIO2		
87	VCCIO2		
87	VCCIO2		
88	AVDD2		

1.4.6 RK3126 function IO package descriptions

Please refer to the chapter 1.3.2 for the detail RK3128 function IO package descriptions.

1.5 Electrical Specification

1.5.1 Absolute Maximum Ratings

Table 1-6 RK3128 absolute maximum ratings

Paramerters	Related Power Group	Max	Unit
DC supply voltage for Internal digital logic	AVDD,CVDD, USB_DVDD11,HDMI_DVDD1V1_1,LVDS/MIPI_DVDD11	1.21	V
DC supply voltage for Digital GPIO (except for SAR-ADC, PLL, USB, DDR IO)	VCCIO1,VCCIO2,VCCIO3,VCCIO4	3.6	V
DC supply voltage for DDR IO	DDR_VDD	1.95	V
DC supply voltage for Analog part of SAR-ADC	SAR_AVDD33	3.6	V
DC supply voltage for Analog part of PLL	PLL_VCCIO A/DPLL_DVDD11,C/GPLL_DVDD11	3.3 1.21	V
DC supply voltage for Analog part of USB OTG/Host2.0	USB_AVDD33	3.63	V
DC supply voltage for Analog part of HDMI	HDMI_AVDD33	3.63	V
DC supply voltage for Analog part of Acodec	CODEC_AVDD	3.63	V
DC supply voltage for Analog part of LVDS	LVDS/MIPI_VCC1	3.63	V
Analog Input voltage for SAR-ADC		2.75	V
Analog Input voltage for DP/DM/VBUS of USB OTG/Host2.0		5	V
Digital input voltage for input buffer of GPIO		3.6	V
Digital output voltage for output buffer of GPIO		3.6	V
Storage Temperature		150	°C

Absolute maximum ratings specify the values beyond which the device may be damaged permanently. Long-term exposure to absolute maximum ratings conditions may affect device reliability.

1.5.2 Recommended Operating Conditions

Table 1-7 RK3128 recommended operating conditions

Parameters	Symbol	Min	Typ	Max	Units
Internal digital logic Power	AVDD,CVDD, USB_DVDD11,HDMI_DVDD1V1_1,LVDS/MIPI_DVDD11	0.99	1.1	1.21	V
Digital GPIO Power(3.3V)	VCCIO1,VCCIO2,VCCIO3,VCCIO4	2.97	3.3	3.63	V
DDR IO (DDRIII mode) Power	DDR_VDD	1.425	1.5	1.575	V

DDR IO (LVDDRIII mode) Power	DDR_VDD	1.28	1.35	1.45	V
PLL Analog Power	PLL_VCCIO	2.97	3.3	3.63	V
PLL Analog Power	A/DPLL_DVDD11 ,C/GPLL_DVDD11	0.99	1.1	1.21	V
SAR-ADC Analog Power	SAR_AVDD33	2.97	3.3	3.63	V
SAR-ADC external reference Power	VREF	0.2* SAR_A VDD33		0.9* SAR_A VDD33	
USB OTG/Host2.0 Analog Power(3.3V)	USB_AVDD33	2.97	3.3	3.63	V
USB OTG/Host2.0 external resistor	REXT	40.5	45	49.5	Ohm
ACodec Analog Power	CODEC_AVDD	2.97	3.3	3.63	V
HDMI Analog Power	HDMI_AVDD33	2.97	3.3	3.63	V
LVDS/MIPI Analog Power	MIPI/LVDS_VCC1	2.97	3.3	3.63	V
VDAC Analog Power	ADDHV6	2.97	3.3	3.63	V
EFUSE programming voltage		N/A	2.5	N/A	V
PLL input clock frequency		N/A	24	N/A	MHz
Operating Temperature		-40	25	85	°C

1.5.3 DC Characteristics

Table 1-8 RK3128 DC Characteristics

Parameters	Symbol	Min	Typ	Max	Units
Digital GPIO @3.3V	Input Low Voltage	Vil	-0.3	0	V
	Input High Voltage	Vih	2	3.3	V
	Output Low Voltage	Vol	N/A	0	V
	Output High Voltage	Voh	2.4	3.3	V
	Threshold Point	Vt	1.21	1.42	V
	Schmitt trig Low to High threshold point	Vt+	1.36	1.6	V
	Schmitt trig High to Low threshold point	Vt-	0.93	1.09	V
	Pullup Resistor	Rpu	33	41	Kohm
	Pulldown Resistor	Rpd	33	42	Kohm
Digital	Input Low Voltage	Vil	-0.3	0	V
				0.63	

GPIO @1.8 V	Input High Voltage	Vih	1.17	1.8	2.1	V
	Output Low Voltage	Vol	N/A	0	0.45	V
	Output High Voltage	Voh	1.35	1.8	N/A	V
	Threshold Point	Vt	0.72	0.83	0.95	V
	Schmitt trig Low to High threshold point	Vt+	0.74	0.88	1.03	V
	Schmitt trig High to Low threshold point	Vt-	0.52	0.61	0.73	V
	Pullup Resistor	Rpu	67	93	152	Kohm
DDR IO @DD RIII mode	Pulldown Resistor	Rpd	64	92	170	Kohm
	Input High Voltage	Vih_ddr	VREFi + 0.125 (i=0~2)	1.8	VDDIO_DDRi + 0.3 (i=0~6)	V
	Input Low Voltage	Vil_ddr	-0.3	0	VREFi - 0.125 (i=0~2)	V
	Output High Voltage	Voh_ddr	VDDIO_DDRi - 0.28 (i=0~6)	1.8	N/A	V
	Output Low Voltage	Vol_ddr	N/A	0	0.28	V
DDR IO @LPD DR mode	Input termination resistance(ODT) to VDDIO_DDRi /2 (i=0~6)	Rtt	120 60 40	150 75 50	180 90 60	Ohm
	Input High Voltage	Vih_ddr	0.7*VDDIO_DDRi (i=0~6)	1.8	N/A	V
	Input Low Voltage	Vil_ddr	N/A	0	0.3*VDDIO_DDRi (i=0~6)	V
PLL	Input High Voltage	Vih_pll	0.8*DVID_iPLL (i=A,D,C,G)	DVDD_iPLL (i=A,D,CG)	DVDD_iPLL (i=A,D,CG)	V
	Input Low Voltage	Vil_pll	0	0	0.2*DVID_iPLL (i=A,D,CG)	V
LVDS	Output voltage high, Voa or	Voh	N/A	N/A	1100	mV

	Vob(Rload = 100om+-1%)					
	Output voltage low, Voa or Vob(Rload = 100om+-1%)	Vol	700	N/A	N/A	mV
	Output differential voltage (Rload = 100om+-1%)	Vod	250	N/A	400	mV
	Output offset voltage(Rload = 100om+-1%)	Vos	825	N/A	975	mv
	Change in Vod between '0' and '1'(Rload = 100om+-1%)	△Vod	N/A	N/A	25	mV
	change in Vos between '0' and '1'(Rload = 100om+-1%)	△Vos	N/A	N/A	25	mV
	Output impedance, single ended(Vcm= 1.0v and 1.4v)	Ro	40	N/A	100	ohm
	Ro mismatch between A & B (Vcm=1.0v and 1.4v)	ΔRo	N/A	N/A	10	%
HDMI	single-ended high level output voltage, VH(when sink <=165Mhz)	Voh	HDMI_AV DD33-10 mv	N/A	HDMI_AVD D33+10mv	mV
	single-ended high level output	Voh	HDMI_AV DD33- 200mv	N/A	HDMI_AVD D33+10mv	mV

	voltage, VH(when sink > 165Mhz)					
	single-ended low level output voltage, VL (when sink <= 165Mhz)	Vol	HDMI_AV DD33 - 600mv	N/A	HDMI_AVD D33-400mv	mV
	single-ended low level output voltage, VL (when sink > 165Mhz)	Vol	HDMI_AV DD33- 700mv	N/A	HDMI_AVD D33-400mv	mv
	single-ended output swing voltage, Vswing	Vswing	400	N/A	600	mV
	single-ended standby (off) output voltage,	Voff	HDMI_AV DD33 - 10mv	N/A	HDMI_AVD D33+10mv	mv
	single-ended standby (off) output current	Ioff	-10	N/A	10	uA
MIPI	HS TX static Common-mode voltage	Vcmtx	150	200	250	mV
	VCMTX mismatch when output is Differential- 1 or Differential- 0	$ \Delta V_{CMTX} $ (1,0)	N/A	N/A	5	mV
	HS transmit differential voltage	VODI	140	200	270	mV
	VOD mismatch when output is Differential- 1 or Differential- 0	$ \Delta V_{ODI} $	N/A	N/A	10	mv
	HS output high voltage	VOHHS	N/A	N/A	360	mV
	Single ended output	ZOS	40	50	62.5	ohm

	impedance					
	Single ended output impedance mismatch	ΔZ_{OS}	N/A	N/A	10	%

1.5.4 Recommended Operating Frequency

Table 1-9 Recommended operating frequency for PLL and oscillator domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
XIN Oscillator	1.1V , 25 °C	XIN24M	24	24	24	MHz
	1.21V , -40 °C		24	24	24	
	0.99V , 125 °C		24	24	24	
DDR PLL	1.1V , 25 °C	ddr_pll_clk	N/A	N/A	1050	MHz
	1.21V , -40 °C		N/A	N/A	1176	
	0.99V , 125 °C		N/A	N/A	950	
ARM PLL	1.1V , 25 °C	arm_pll_clk	N/A	N/A	1086	MHz
	1.21V , -40 °C		N/A	N/A	1176	
	0.99V , 125 °C		N/A	N/A	850	
CODEC PLL	1.1V , 25 °C	cocecc_pll_clk	N/A	N/A	880	MHz
	1.21V , -40 °C		N/A	N/A	1000	
	0.99V , 125 °C		N/A	N/A	770	
GENERAL PLL	1.1V , 25 °C	general_pll_clk	N/A	N/A	900	MHz
	1.21V , -40 °C		N/A	N/A	940	
	0.99V , 125 °C		N/A	N/A	780	

Table 1-10 Recommended operating frequency for A9 core

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
Cortex-A9	1.1V , 25 °C	CORE_SRC_CLK	N/A	N/A	970	MHz
	1.21V , -40 °C		N/A	N/A	1060	
	0.99V , 125 °C		N/A	N/A	790	
	1.1V , 25 °C	aclk_core_pre	N/A	N/A	490	MHz
	1.21V , -40 °C		N/A	N/A	520	
	0.99V , 125 °C		N/A	N/A	400	

Table 1-11 Recommended operating frequency for PD_CPU domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
CPU AXI interconnect	1.1V , 25 °C	CPU_ACLK	N/A	N/A	500	MHz
	1.21V , -40 °C		N/A	N/A	650	
	0.99V , 125 °C		N/A	N/A	300	
	1.1V , 25 °C		N/A	N/A	320	MHz
	CPU_HCLK					

	1.21V , -40 °C		N/A	N/A	470	
	0.99V , 125 °C		N/A	N/A	180	
	1.1V , 25 °C	CPU_PCLK	N/A	N/A	90	MHz
	1.21V , -40 °C		N/A	N/A	90	
	0.99V , 125 °C		N/A	N/A	80	
DMC	1.1V , 25 °C	DDR_PHY1X_CLK	N/A	N/A	900	MHz
	1.21V , -40 °C		N/A	N/A	760	
	0.99V , 125 °C		N/A	N/A	400	

Table 1-12 Recommended operating frequency for PD_PERI domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
PERI AXI interconnect	1.1V , 25 °C	PERI_ACLK	N/A	N/A	470	MHz
	1.21V , -40 °C		N/A	N/A	600	
	0.99V , 125 °C		N/A	N/A	300	
	1.1V , 25 °C	PERI_HCLK	N/A	N/A	180	MHz
	1.21V , -40 °C		N/A	N/A	200	
	0.99V , 125 °C		N/A	N/A	150	
	1.1V , 25 °C	PERI_PCLK	N/A	N/A	80	MHz
	1.21V , -40 °C		N/A	N/A	88	
	0.99V , 125 °C		N/A	N/A	75	
NANDC	1.1V , 25 °C	FLASH_HCLK	N/A	N/A	190	MHz
	1.21V , -40 °C		N/A	N/A	220	
	0.99V , 125 °C		N/A	N/A	150	
USB OTG	1.1V , 25 °C	UTMI_CLK_0/ UTMI_CLK_1	N/A	N/A	140	MHz
	1.21V , -40 °C		N/A	N/A	200	
	0.99V , 125 °C		N/A	N/A	76	
UART0/1/2	1.1V , 25 °C	UART0_CLK/ UART1_CLK/ UART2_CLK	N/A	N/A	100	MHz
	1.21V , -40 °C		N/A	N/A	100	
	0.99V , 125 °C		N/A	N/A	100	
SDMMC/SDIO	1.1V , 25 °C	MMC0_CLK/ SDIO_CLK	N/A	N/A	100	MHz
	1.21V , -40 °C		N/A	N/A	100	
	0.99V , 125 °C		N/A	N/A	100	
EMMC	1.1V , 25 °C	EMMC_CLK	N/A	N/A	100	MHz
	1.21V , -40 °C		N/A	N/A	100	
	0.99V , 125 °C		N/A	N/A	100	
GPS	1.1V , 25 °C	GPS_RFCLK	N/A	N/A	50	MHz
	1.21V , -40 °C		N/A	N/A	50	
	0.99V , 125 °C		N/A	N/A	50	

	1.1V , 25 °C	GPS_HCLK	N/A	N/A	300	MHz
	1.21V , -40 °C		N/A	N/A	650	
	0.99V , 125 °C		N/A	N/A	300	
I2S	1.1V , 25 °C	I2S_CLK	N/A	N/A	50	MHz
	1.21V , -40 °C		N/A	N/A	50	
	0.99V , 125 °C		N/A	N/A	50	
SPI0	1.1V , 25 °C	SPI0_CLK	N/A	N/A	50	MHz
	1.21V , -40 °C		N/A	N/A	50	
	0.99V , 125 °C		N/A	N/A	50	
SAR-ADC	1.1V , 25 °C	SARADC_CLK	N/A	N/A	12	MHz
	1.21V , -40 °C		N/A	N/A	12	
	0.99V , 125 °C		N/A	N/A	12	
Timer0/1	1.1V , 25 °C	TIMER0_CLK/ TIMER1_CLK	N/A	N/A	24	MHz
	1.21V , -40 °C		N/A	N/A	24	
	0.99V , 125 °C		N/A	N/A	24	
	1.1V , 25 °C	TIMER0_PCLK / TIMER1_PCLK	N/A	N/A	140	MHz
	1.21V , -40 °C		N/A	N/A	190	
	0.99V , 125 °C		N/A	N/A	75	

Table 1-13 Recommended operating frequency for PD_VIO domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
Display AXI interconnection	1.1V , 25 °C	DISP_ACLK	N/A	N/A	520	MHz
	1.21V , -40 °C		N/A	N/A	700	
	0.99V , 125 °C		N/A	N/A	300	
LCDC	1.1V , 25 °C	DISP_HCLK	N/A	N/A	360	MHz
	1.21V , -40 °C		N/A	N/A	500	
	0.99V , 125 °C		N/A	N/A	200	
LCDC	1.1V , 25 °C	LCDC_DCLK	N/A	N/A	200	MHz
	1.21V , -40 °C		N/A	N/A	230	
	0.99V , 125 °C		N/A	N/A	150	
CIF	1.1V , 25 °C	LCDC1_DCLK	N/A	N/A	200	MHz
	1.21V , -40 °C		N/A	N/A	240	
	0.99V , 125 °C		N/A	N/A	160	
CIF	1.1V , 25 °C	IO_CIF_CLKIN	N/A	N/A	100	MHz
	1.21V , -40 °C		N/A	N/A	100	
	0.99V , 125 °C		N/A	N/A	100	

Table 1-14 Recommended operating frequency PD_GPU domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
GPU	1.1V , 25 °C	GPU_ACLK	N/A	N/A	550	MHz
	1.21V , -40 °C		N/A	N/A	720	
	0.99V , 125 °C		N/A	N/A	300	

Table 1-15 Recommended operating frequency for PD_VIDEO domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
VIDEO	1.1V , 25 °C	VEPU_ACLK	N/A	N/A	490	MHz
	1.21V , -40 °C		N/A	N/A	650	
	0.99V , 125 °C		N/A	N/A	300	
	1.1V , 25 °C	hclk_vepu	N/A	N/A	200	MHz
	1.21V , -40 °C		N/A	N/A	225	
	0.99V , 125 °C		N/A	N/A	150	
	1.1V , 25 °C	VDPU_ACLK	N/A	N/A	420	MHz
	1.21V , -40 °C		N/A	N/A	610	
	0.99V , 125 °C		N/A	N/A	280	
hclk_vdpu	1.1V , 25 °C	hclk_vdpu	N/A	N/A	200	MHz
	1.21V , -40 °C		N/A	N/A	220	
	0.99V , 125 °C		N/A	N/A	150	

1.5.5 Electrical Characteristics for General IO

Table 1-16 RK3128 Electrical Characteristics for Digital General IO

Parameters	Symbol	Test condition	Min	Typ	Max	Units
Digital GPIO @3.3V	Input leakage current	Ii	Vin = 3.3V or 0V	-1	N/A	1 uA
	Tri-state output leakage current	IoZ	Vout = 3.3V or 0V	-1	N/A	1 uA
	High level input current	Iih	Vin = 3.3V, pulldown disabled	TBD	N/A	TBD uA
			Vin = 3.3V, pulldown enabled	TBD	TBD	TBD uA
	Low level input current	Iil	Vin = 0V, pullup disabled	TBD	N/A	TBD uA
			Vin = 0V, pullup enabled	TBD	TBD	TBD uA
Digital GPIO @1.8V	Input leakage current	Ii	Vin = 1.8V or 0V	-1	N/A	1 uA
	Tri-state output leakage current	IoZ	Vout = 1.8V or 0V	-1	N/A	1 uA
	High level input current	Iih	Vin = 1.8V, pulldown disabled	TBD	N/A	TBD uA
			Vin = 1.8V, pulldown enabled	TBD	TBD	TBD uA
	Low level input current	Iil	Vin = 0V, pullup disabled	TBD	N/A	TBD uA

		Vin = 0V, pullup enabled	TBD	TBD	TBD	uA
--	--	--------------------------	-----	-----	-----	----

1.5.6 Electrical Characteristics for PLL

Table 1-17 RK3128 Electrical Characteristics for PLL

Parameters	Symbol	Test condition	Min	Typ	Max	Units
PLL	Input clock frequency	Fin Fin = FREF @3.3V/1.1V①	1/10	24	800	MHz
	Comparison frequency	Fref FREF = Fin/REFDIV @3.3V/1.1V	1	N/A	40	MHz
	VCO operating range	Fvco Fvco = Fref * FBDIV① @3.3V/1.1V	400	N/A	1600	MHz
	Output clock frequency	Fout Fout = Fvco/POSTDIV① @3.3V/1.1V	1	N/A	1600	MHz
	Lock time②	Tlt @ 3.3V/1.1V, FREF=24M,REFDIV=1	N/A	41.7	62.5	us
	VDDHV Power consumption ③ (normal mode)	N/A Fvco = 1000MHz, @3.3V, 25 °C	N/A	1	1.2	mA
	VDD Power consumption (normal mode)	N/A @3.3V/1.1V, 25 °C	N/A	3	4	uW/MHz
	Power consumption (bypass mode)	N/A BYPASS=HIGH , PD=LOW , Fin = 24MHz, Fout = 24MHz, @3.3V/1.1V, 25 °C	N/A	N/A	N/A	uW
	Power consumption (power-down mode)	N/A PD=HIGH, @27 °C	N/A	10	N/A	uA

Notes :

①:REFDIV is the input divider value;
FBDIV is the feedback divider value;
POSTDIV is the output divider value

② Lock Time is 1000cycles of input clocks in typ, and 1500cycles of input clocks in max.

③ Current scale as (Fvco/1GHz)^{1.5}

1.5.7 Electrical Characteristics for SAR-ADC

Table 1-18 RK3128 Electrical Characteristics for SAR-ADC

Parameters	Symbol	Test condition	Min	Typ	Max	Units
ADC resolution			N/A	10	N/A	bits
Conversion speed	Fs		N/A	N/A	N/A	MSPS
Differential Non Linearity	DNL		N/A	N/A	N/A	LSB
Integral Non Linearity	INL		N/A	N/A	N/A	LSB
Gain Error	Egain		N/A	N/A	N/A	%FS
Offset Error	Eoffset		N/A	N/A	N/A	%FS
Input Range	CH[2:0]	3-channel	0.01*	N/A	0.99*	V

		single-ended input	SAR_AVDD33		SAR_AVDD33	
Input Resistance	RIN		N/A	N/A	N/A	KΩ
Input Capacitance	CIN		N/A	1	N/A	pF
Sampling Clock			N/A	200	N/A	KHz
Main Clock Frequency	CLK		N/A	2.2	N/A	MHz
Data Latency			N/A	11	N/A	Clock Cycle
SNR plus Distortion(Up to 5th harmonic)	SINAD	Fin=10K Fin=99K	N/A	61.49 60.58	N/A	dB
Spurious-Free Dynamic Range	SFDR	Fin=10K Fin=99K	N/A	66.29 67.14	N/A	dB
Second-Harmonic Distortion	2HD	Fin=10K Fin=99K	N/A	-72.64 -69.94	N/A	dB
Third-Harmonic Distortion	3HD	Fin=10K Fin=99K	N/A	-74.79 -68.85	N/A	dB
Effective Number of Bits	ENOB	Fin=10K Fin=99K	N/A	9.92 9.77	N/A	Bits
Positive Reference	VREF		0.2* SARADC_AVDD33		0.9* SARADC_AVDD33	V
Analog Supply Current(SARADC_VDDA)			N/A	N/A	200	uA
Digital Supply Current			N/A	N/A	50	uA
Reference Supply Current			N/A	N/A	50	uA
Power Down Current			N/A	N/A	N/A	uA
Power up time			N/A	N/A	N/A	1/Fs

1.5.8 Electrical Characteristics for USB OTG/Host2.0 Interface

Table 1-19 RK3128 Electrical Characteristics for USB OTG/Host2.0 Interface

Parameters	Test condition	Min	Typ	Max	Units
HS transmit,(quiescent supply current; Vin=0 or 1)	Current From USB_AVDD33	N/A	N/A	0.1	mA
	Current From USB_DVDD11	N/A	N/A	20	mA
Classic mode active(quiescent supply current; Vin=0 or 1)	Current From USB_AVDD33	N/A	N/A	0.5	mA
	Current From USB_DVDD11	N/A	N/A	0.5	mA
HS mode(CL=10pF) Active supply current	Current From USB_AVDD33	N/A	0.1	N/A	mA
	Current From USB_DVDD11	N/A	2.22	N/A	mA
FS transmit,(CL=50pF) Active supply current	Current From USB_AVDD33	N/A	10	30	mA
	Current From USB_DVDD11	N/A	5	10	mA
LS transmit(CL=50 to 350pF) Active supply current	Current From USB_AVDD33	N/A	2	25	mA
	Current From USB_DVDD11	N/A	2	5	mA
Suspend mode	Current From	N/A	N/A	50	uA

USB_AVDD33		N/A	N/A	5	uA
Current From USB_DVDD11					

1.5.9 Electrical Characteristics for HDMI

Table 1-20 RK3128 Electrical Characteristics for HDMI

Parameters	Symbol	Test condition	Min	Typ	Max	Units
rise time/fall time(20%-80%)	Tfall/Trise		75	N/A	0.4Tbit	ps
			15%	of full differential amplitude(Vswing*2)		ps
			25%	of full differential amplitude(Vswing*2)		ps
			N/A	N/A	0.15 Tbit	ps
			N/A	N/A	0.2 Tpixel	ps
			N/A	N/A	0.25 Tbit	ps
			40%	N/A	60%	

1.5.10 Electrical Characteristics for DDR IO

Table 1-21 RK3128 Electrical Characteristics for DDR IO

Parameters	Symbol	Test condition	Min	Typ	Max	Units
DDR IO @DDR3 mode	DDR IO power standby current, ODT OFF	@ 1.5V , 125°C	N/A	N/A	N/A	uA
	Input leakage current, SSTL mode, unterminated	@ 1.5V , 125°C	N/A	N/A	N/A	uA
DDR IO @LVDDR3 mode	Input leakage current	@ 1.35V , 125°C	N/A	N/A	N/A	uA
	DDR IO power quiescent current	@ 1.35V , 125°C	N/A	N/A	N/A	uA

1.5.11 Electrical Characteristics for eFuse

Table 1-22 RK3128 Electrical Characteristics for eFuse

	Parameters	Symbol	Test condition	Min	Typ	Max	Units
Active mode	read current	Iactive	STROBE high	N/A	2.53	N/A	mA
standby mode	standby current	Istandby		N/A	0.4	N/A	uA
power-down mode	power-down current	Ipdown		N/A	N/A	N/A	uA

Peak program current	Peak program current	Iprog		N/A	20.8	N/A	mA
----------------------	----------------------	-------	--	-----	------	-----	----

1.5.12 Electrical Characteristics for MIPI

Table 1-23 RK3128 Electrical Characteristics for MIPI HS transmitter

	Parameters	Symbol	Test condition	Min	Typ	Max	Units
	Common-mode variations above 450 MHz	$\Delta V_{CMTX}(HF)$		N/A	N/A	15	mVRMS
	Common-mode variations between 50MHz – 450MHz	$\Delta V_{CMTX}(LF)$		N/A	N/A	25	mVPEAK
	20%-80% rise time and fall time	TR and TF		N/A	N/A	0.3	UI
				150	N/A	N/A	ps

Note: 1.UI is equal to $1/(2*fh)$. See section 7.3 for the definition of fh.

1.5.13 Electrical Characteristics for LVDS

Table 1-24 RK3128 Electrical Characteristics for LVDS HS transmitter

	Parameters	Symbol	Test condition	Min	Typ	Max	Units
	Vod fall time, 20-80%	Tfall	$R_{load} = 100 \Omega \pm 1\%$	100	N/A	250	ps
	Vod rise time, 20-80%	Trise	$R_{load} = 100 \Omega \pm 1\%$	100	N/A	250	ps
	$ tpHLA - tpLHB $ or $ tpHLB - tpLHA $, Differential skew	Tskew1	Any differential pair on package	N/A	N/A	30	ps
	$ tpdiff[m] - tpdiff[n] $, Channel-to-channel skew	Tskew2	Any two signals on package	N/A	N/A	50	ps

1.5.14 Electrical Characteristics for VDAC

Table 1-25 RK3128 Electrical Characteristics for VDAC

Parameters	Symbol	Test condition	Min	Typ	Max	Units
Bandgap Voltage	Vbg		N/A	1.21	N/A	V
Reference Resistor		E96 series	N/A	1130	N/A	ohm
Reference Current			N/A	1.07	N/A	mA
Output Full Scale Current		Programmable through dacXgc5..0 word (external load of 37.50hm) Refer to	N/A	N/A	34	mA

		Operating Modes for details				
Resistive Load			N/A	37.5	N/A	Ohm
Offset Error			N/A	+/-1	N/A	%FS
Gain Error(DAC to DAC matching)			N/A	+/-2	N/A	%FS
Absolute Gain Error			N/A	+/-4	N/A	%FS
DNL		I _{fs} =34mA	N/A	+/-0.5	N/A	LSB
INL		I _{fs} =34mA	N/A	+/-1.0	N/A	LSB
Update Rate			1	N/A	300	MHz
Startup Time		From Complete shut-down to normal operation	N/A	3	4	Us
Cable sensing Cycle time		Details on Cable Sensing Cycle Timing Diagram	N/A	4.5	N/A	Clk cycles
SFDR	SFDR	F _{out} =5MHz, I _{fs} =34mA, R _L =37.5ohm, F _s =300MHz	N/A	58	N/A	dBc
		F _{out} =1MHz, I _{fs} =34mA, R _L =37.5ohm, F _s =300MHz	N/A	61	N/A	dBc
SINAD	SINAD	F _{out} =5MHz, I _{fs} =34mA, R _L =37.5ohm, F _s =300MHz	N/A	54	N/A	dBc
		F _{out} =1MHz, I _{fs} =34mA, R _L =37.5ohm, F _s =300MHz	N/A	57	N/A	dBc
High Voltage Analog Current(avddhv6.0)		I _{fs} =34mA	N/A	51	N/A	mA
Digital Current(dvdd)		F _s =300MHz	N/A	0.7	N/A	mA
Power down current		High Voltage Analog supply and digital supply	N/A	60	N/A	uA

1.6 Hardware Guideline

1.6.1 Reference design for RK3128 oscillator PCB connection

RK3128 only use one oscillator, and its typical clock frequency is 24MHz. The oscillator will provide input clock to four on-chip PLLs.

- External reference circuit for oscillators with 24MHz input

In the following diagram ,Rf is used to bias the inverter in the high gain region.

The recommend value is 1Mohm.

R_d is used to increase stability, low power consumption, suppress the gain in high frequency region and also reduce $-R_d$ of the oscillator. Thus, proper R_d cannot be too large to cease the loop oscillating.

C_1 and C_2 are deciding regard to the crystal or resonator CL specification. the value for R_f, R_d, C_1, C_2 must be adjusted a little to improve performance of oscillator based on real crystal model .

In RK3128, the crystal oscillator I/O cells have embedded internal resistor, so we need not add feedback resistor (R_f) as above description.

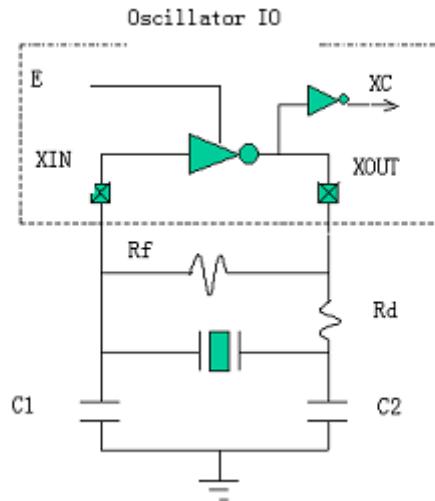


Fig.1-11 External Reference Circuit for 24MHzOscillators

1.6.2 Reference design for PLL PCB connection

The following reference design is suitable for PLL in RK3128.

For optimal jitter performance it is suggested to place external decoupling capacitors on the board between VDDHV-VSS(PLL_VSS1) and VDDPOST-VSS(PLL_VSS2) . VDDREF is typically connected to the global chip supply and does not require dedicated decoupling.

It is recommended to use at least one large capacitor (e.g. 4.7uF) capacitor for each separate supply. Additionally, a 100nF and 10nF capacitor may be placed in parallel since the lead inductance of the 4.7uF capacitor may be large.

Capacitors with minimal lead inductance should be selected. Ceramic type capacitors work well. The capacitors should be placed as close to the package pins as possible. No series impedance should be added anywhere on the board, and impedance to the voltage source should be minimized.

1.6.3 Reference design for USB OTG/Host2.0 connection

In RK3128 there are USB OTG and USB Host2.0 interface, and they share a common PHY.

- Decouple Capacitance

We should include decoupling and bypass capacitors at each power pin in the layout. These are shown schematically in Figure 1-9. Place these components as closely as possible to the power pins.

- Differential Lines

The differential lines should be routed together, minimizing the number of vias through which the signal lines are routed. Layout the differential pairs with

controlled impedance of 100 ohm differential.

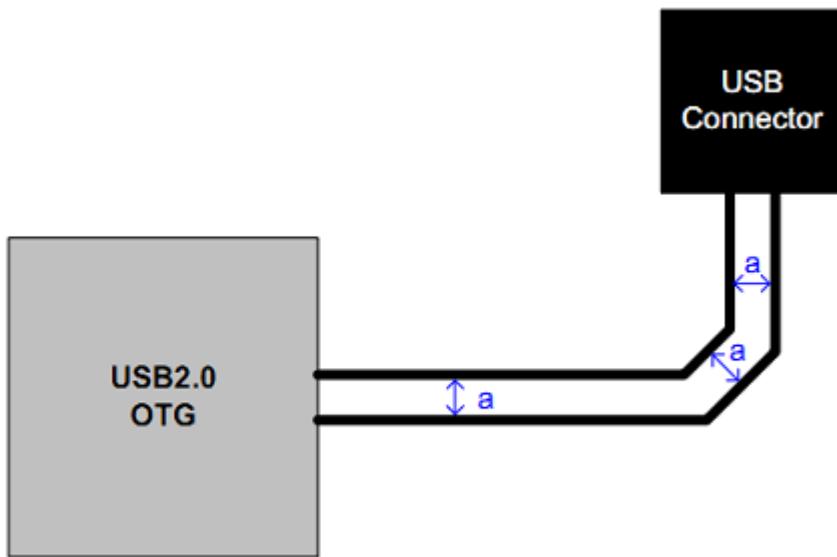


Fig.1-12RK3128 USB OTG/Host2.0 differential lines requirement.

If high-speed signals are routed on the Top layer, best results will be obtained if the Layer 2 is a Ground plane. Furthermore, there must have only one ground plane under high-speed signals in order to avoid the high-speed signals to cross another ground plane.

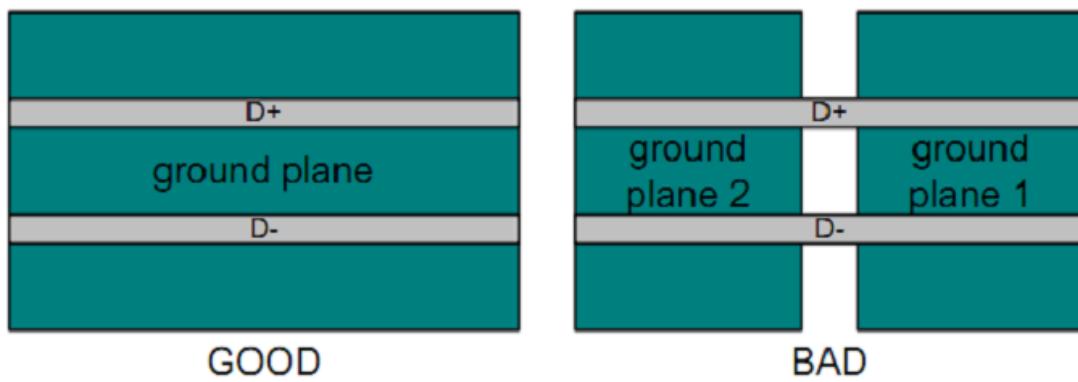


Fig.1-13RK3128 USB OTG/Host2.0 ground plane guide.

- Component Placement

It is very important to not create stubs on the high-speed lines, to avoid that, the placement of component should be the closed as possible from D+ and D- lines, like shown in the following figure.

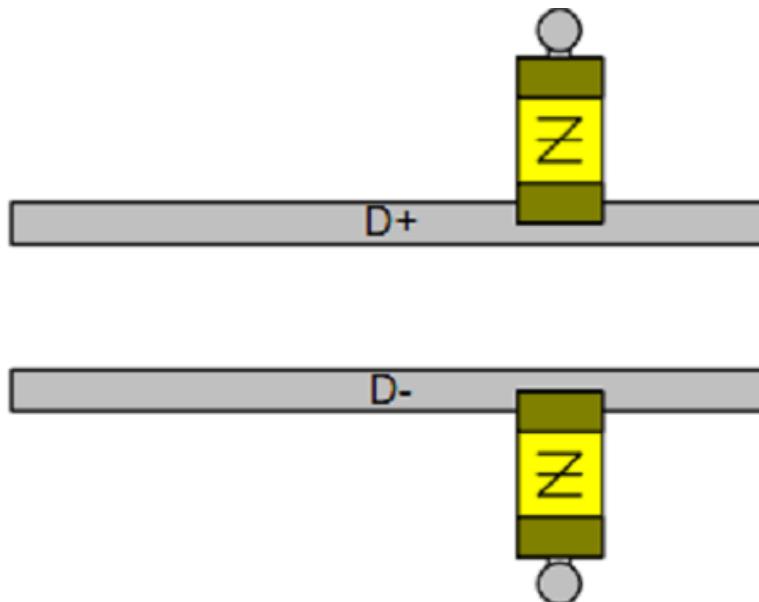


Fig.1-14RK3128 USB OTG/Host2.0 component placement.

1.6.4 Reference design for HDMI Tx PHY connection

In RK3128, the following diagram shows external PCB reference design for HDMI Tx PHY. It mainly introduces how to connect the TMDS channel, DDC channel, CEC channel and HPD signal of RK3128 HDMI Transmitter to the HDMI port type A.

- TMDS channel

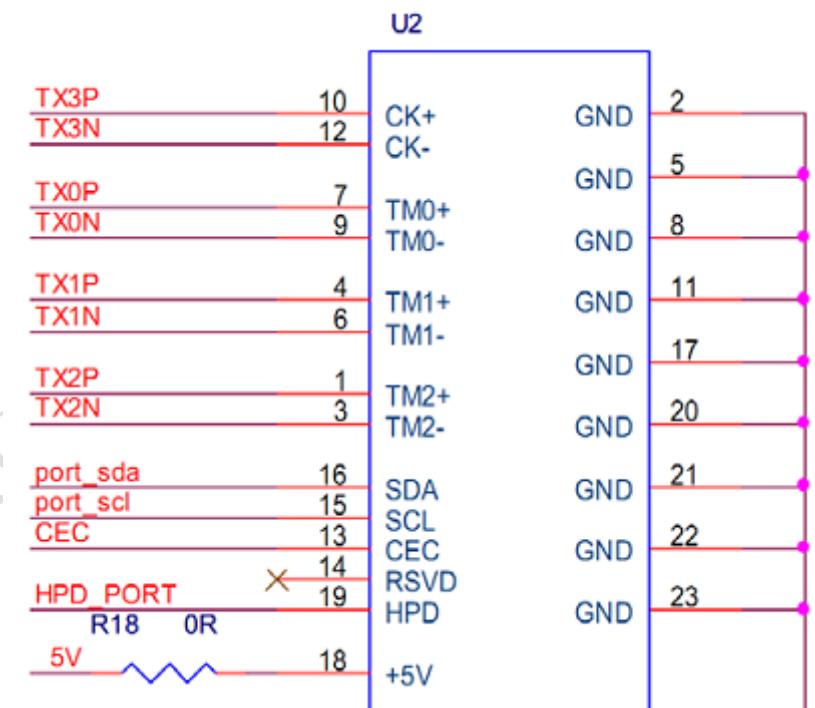


Fig.1-15RK3128 HDMI interface reference connection

- DDC channel

RK3128 can accept DDC_sda/DDC_scl 5V voltage input, it's no need to add additional Transmitter to transfer the DDC_sda/DDC_scl from 5V to 3.3V outside the chip.

- CEC channel

RK3128 can accept CEC 5V voltage input, it's no need to add additional Transmitter to transfer the CEC from 5V to 3.3V outside the chip.

- HPD

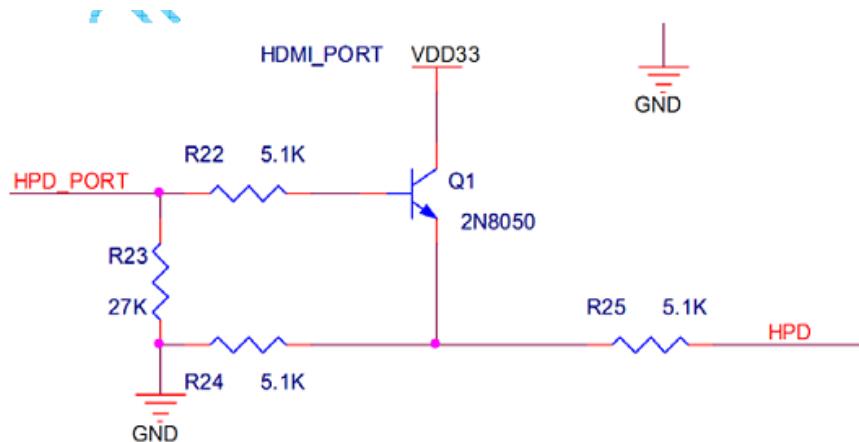


Fig.1-16RK3128 HDMI CEC interface reference connection

- ESD

If ESD suppression devices or common mode chokes are used, place them near the HDMI connector.

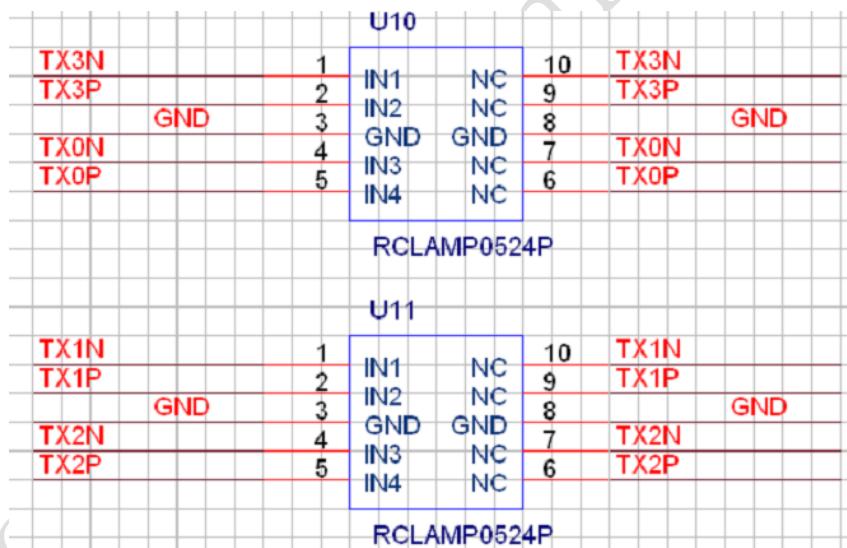


Fig.1-17RK3128 HDMI ESD interface reference connection

1.6.5 Reference design for Audio Codec connection

In RK3128, the following diagram shows external PCB reference design for Audio Codec.

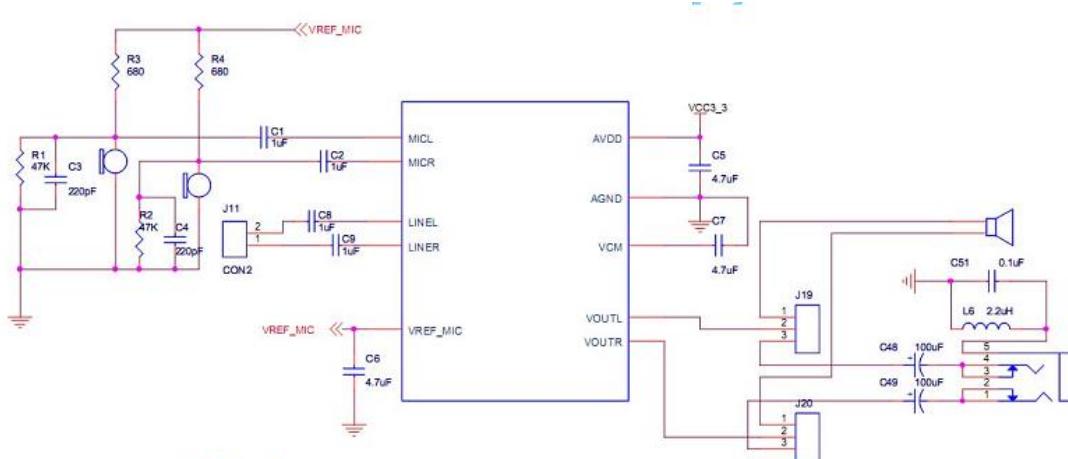


Fig.1-18RK3128 Audio Codec interface reference connection

As above diagram shows, the MICL and MICR are each connected with a MIC through a 1uf CAP, the LINEL and LINER have the same function as the MICL and MICR. The R1 and C3 are formed a filter for the MIC, and the R2, C4 have same function. The VREF_MIC is used for bias the MIC through a resistor. The resistor value should be changed according the MIC. The AVDD should be supplied by 3.3V. The CAP connected with AVDD should be placed as close as possible. The VCM is connected with GND through a 4.7uF CAP. The CAP should be placed as close as possible. The VOUTL and VOUTR could be connected with a speaker or an earphone. When connecting with a speaker, they could connect it directly. When connecting with an earphone, they should connect it through a 100uF CAP. The J19 and J20 are dip-switches, and you could select a speaker or an earphone as the output.

1.6.6 Reference design for GPS connection

Please refer to the GPS_HV5820_设计指南_V1.1.pdf for detail PCB layout guideline.

1.6.7 RK3128 Power on reset descriptions

NPOR is hardware reset signal from out-chip, which is filtered glitch to obtain signal sysrstn. To make PLLs work normally, the PLL reset signal (pllrstn) must maintain high for more than 1us, and PLLs start to lock when pllrstndeassert, and the PLL max lock time is 1500 PLL REFCLK cycles. And then the system will wait about 138us, and then deactivate reset signal chiprstn. The signal chiprstn is used to generate output clocks in CRU. After CRU start output clocks, the system waits again for 512cycles (21.3us) to deactivate signal rstn_pre, which is used to generate power on reset of all IP.

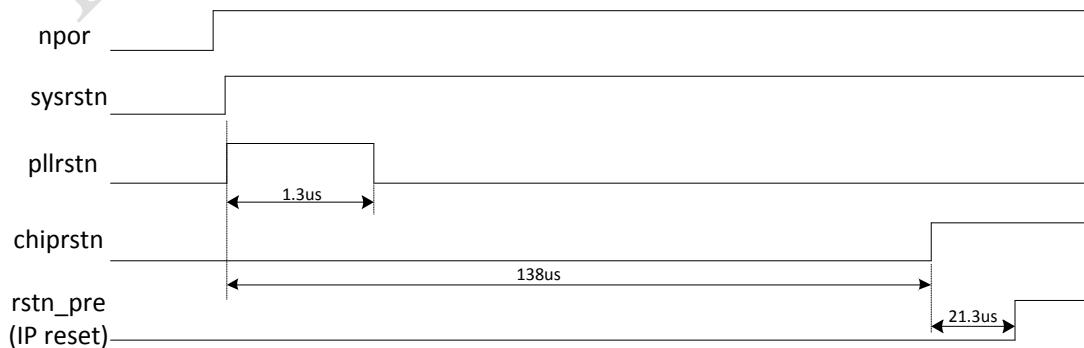


Fig.1-19 RK3128 reset signals sequence

Chapter 2 System Overview

2.1 Address Mapping

RKaudi support to boot from internal bootrom, which support remap function by software programming. Remap is controlled by GRF_SOC_CON0[12].

Addr	IP	Addr	IP	Addr	IP	Addr	IP
	Reserved		MIPI-ANA 20038000 16K		Reserved		Reserved
	Reserved		HDMI-ANA 20034000 16K		NANDC 16K		eFuse 16K
	Reserved		ACODEC-ANA 20030000 16K		GPS 1024K		GMAC 16K
1013e000	Reserved		DBG 20020000 64K		PERI BUS 1024k		GPIO3 16K
1013d000	Reserved 4K		Reserved 20010000 64K		AHB ARB1 784K		GPIO2 16K
1013c000	Reserved 8k		DDR_PHY 2000a000 24K		AHB ARB0 32K		GPIO1 16K
10138000	GIC 16K		GRF 20008000 8K		Reserved 64K		GPIO0 16K
10130000	Reserved 32K		DDR_PCTL 20004000 16K		I2S_2ch 16K		DMAC 16K
10128000	CPU BUS 32K		CRU 20000000 16K		eMMC 16K		SPI 16K
10118000	Reserved 64K				SDIO 16K		I2C0 16K
10114000	EBC 16k				SDMMC 16K		SARADC 16K
10112000	Reserved 8K				SDMMC 16K		UART2 16K
10110000	MIPI_ctrl 8K				SFC 32k		UART1 16K
1010e000	LCDCO 8K				TSP 16k		UART0 16K
1010c000	RGA 8K				SPDIF 16k		I2C3 16K
1010a000	CIF 8K				I2S_8ch 16k		I2C2 16K
10108000	IEP 8K				USB HOST OHCI 128K		I2C1 16K
10104000	VCODEC 16K				USB HOST ECHI 128K		PWM0 16K
10100000	ROM 16K				USB OTG 256K		WDT 16K
100fc000	crypto 16K						SCR 16K
100b0000	Reserved 304K						TIMER0-5 16K
100a0000	PMU 64K						
10090000	GPU 64K						
			before remap		after remap		
				IMEM 10080000 8k		IMEM 8k	
				10080000 / 00000000			

Fig. 2-1 RKaudi Address Mapping

2.2 System Boot

RKaudi provides system boot from off-chip devices such as SDMMC card, 8bits
async nand flash or toggle nand flash, SPI nor or nand, and eMMC memory.

When boot code is not ready in these devices, also provide system code download into them by USB OTG interface. All of the boot code will be stored in internal bootrom. The following is the whole boot procedure for boot code, which will be stored in bootrom in advance.

The following features are supports.

- Support secure boot mode and non-secure boot mode
- Support system boot from the following device:
 - 8bits Async Nand Flash
 - 8bits Toggle Nand Flash
 - SFC interface
 - eMMC interface
 - SDMMC Card
 - Support system code download by USB OTG

Rockchip Confidential

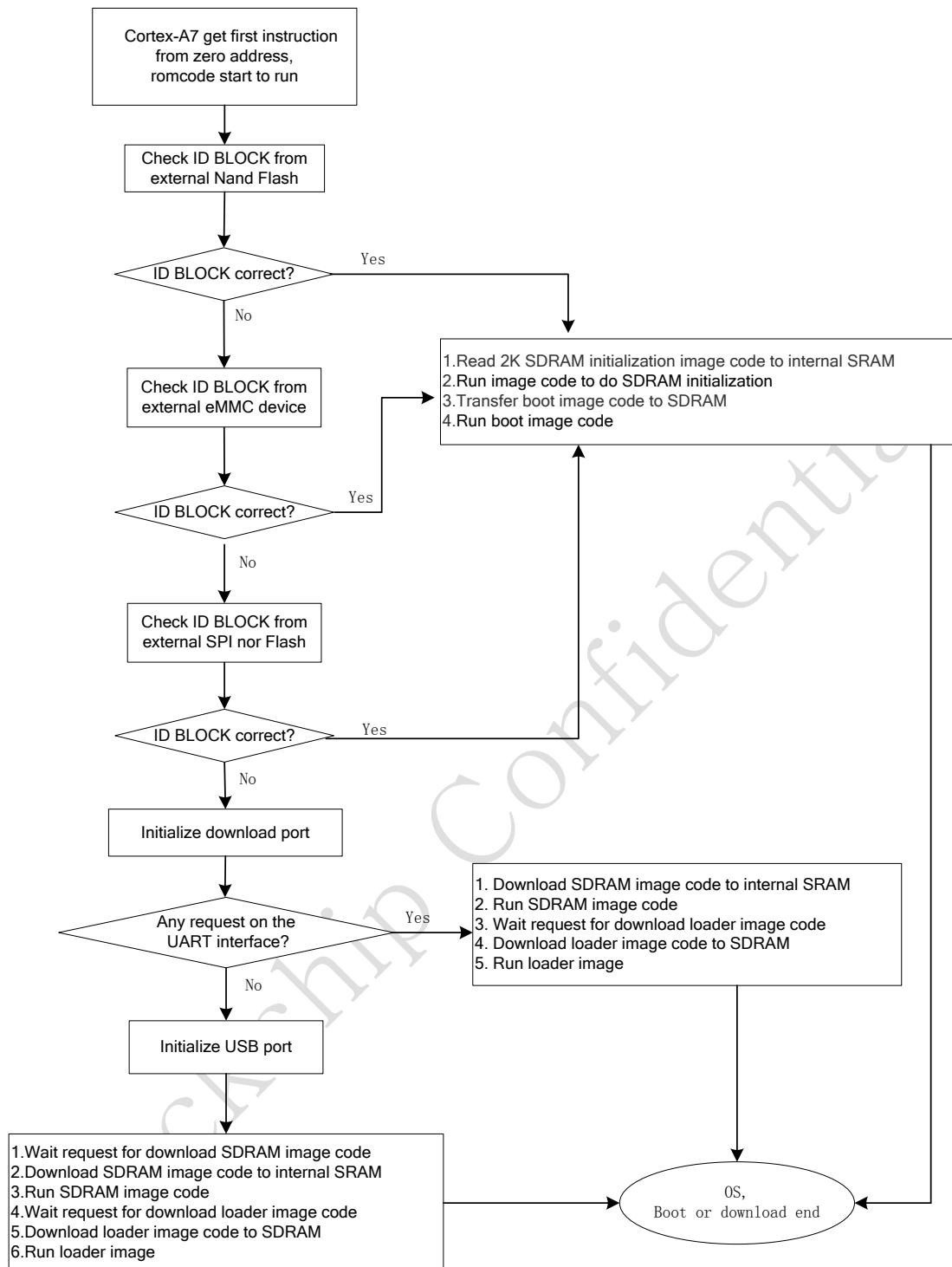


Fig. 2-2 RKaudi boot procedure flow

2.3 System Interrupt connection

RKaudi provides an general interrupt controller(GIC) for Cortex-A7 MPCore processor, which has 112 SPI (shared peripheral interrupts) interrupt sources and 3 PPI(Private peripheral interrupt) interrupt source and separately generates one nIRQ and one nFIQ to CPU. The triggered type for each interrupts is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table. For detailed GIC setting, please refer to

Chapter 12.

Table 2-1 RKaudi Interrupt connection list

IRQ ID	Source	Polarity
PPI	27	VIRTUAL TIMER
	29	SECURE PHYSICAL TIMER
	30	NON-SECURE PHY TIMER
Source(spi)	32	DMAC2(0)
	33	DMAC2(1)
	34	DDR_PCTL
	35	gpu_irqgp
	36	gpu_irqmmu
	37	gpu_irqpp
	38	Video encoder
	39	Video decoder
	40	CIF
	41	LCDC
	42	USB OTG
	43	USB Host EHCI
	44	gps_irq
	45	gps_timer_irq
	46	SD/MMC0
	47	SDIO
	48	eMMC
	49	SAR-ADC
	50	NandC
	51	I2S_2ch
	52	UART0
	53	UART1
	54	UART2
	55	SPI0
	56	I2C0
	57	I2C1
	58	I2C2
	59	I2C3
	60	Timer0
	61	Timer1
	62	PWM
	63	PMU
	64	USB Host OHCI
	65	MIPI_controller
	66	WDT
	67	otg_bvalid_irq
	68	GPIO0

69	GPIO1	High level
70	GPIO2	High level
71	GPIO3	High level
72	CRYPTO	High level
73	reserved	High level
74	peri_ahb_usb arbiter	High level
75	reserved	High level
76	RGA	High level
77	hdmi	High level
78	SD/MMC detect	High level
79	SDIO detect	High level
80	IEP	High level
81	EBC	High level
82	sfc	High level
83	otg0_id_irq	High level
84	otg0_linestate_irq	High level
85	otg1_linestate_irq	High level
86	sd_detectn_irq	High level
87	spdif	High level
88	gmac	High level
89	gmac_tmc	High level
90	tsp	High level
91	timer2	High level
92	timer3	High level
93	timer4	High level
94	timer5	High level
95	sim_card	High level
96	acodec_detectn_irq	High level
97	hevc_mmu_irq	High level
98	hevc_dec_irq	High level
99	vpu_mmu_irq	High level
100	i2s_8ch	High level
101	Reserved	High level
102	Reserved	High level
103	Reserved	High level
104	Reserved	High level
105	Reserved	High level
106	Reserved	High level
107	Reserved	High level
108	pmuirq_a7_0	High level
109	pmuirq_a7_1	High level
110	pmuirq_a7_2	High level

	111	pmuirq_a7_3	High level
	112	axierrirq	High level

2.4 System DMA hardware request connection

RKaudi provides one DMA controller: DMAC inside peripheral system. 15 hardware request ports are used in DMAC_PERI, the trigger type for each of them is high level, not programmable. For detailed descriptions of DMAC, please refer to Chapter 11.

Table 2-2 RKaudi DMAC_BUS Hardware request connection list

Req Number	Source	Polarity
0	I2S_2ch tx	High level
1	I2S_2ch rx	High level
2	Uart0 tx	High level
3	Uart0 rx	High level
4	Uart1 tx	High level
5	Uart1 rx	High level
6	Uart2 tx	High level
7	Uart2 rx	High level
8	SPI tx	High level
9	SPI rx	High level
10	SD/MMC	High level
11	SDIO	High level
12	eMMC	High level
13	SPDIF	High level
14	I2S_8ch tx	High level
15	I2S_8ch rx	High level

Chapter 3 Clock and reset unit

3.1 Overview

The CRU is an APB slave module that is designed for generating all of the internal and system clocks, resets of chip. CRU generates system clock from PLL output clock or external clock source, and generates system reset from external power-on-reset, watchdog timer reset or software reset.

CRU supports the following features:

- Compliance to the AMBA APB interface
- Embedded four PLLs
- Support only one crystal
- Flexible selection of clock source
- Supports the respective gating of all clocks
- Supports the respective software reset of all modules

3.2 Block Diagram

The CRU comprises with:

- Four PLLs
- Register configuration unit
- Clock generate unit
- Reset generate unit

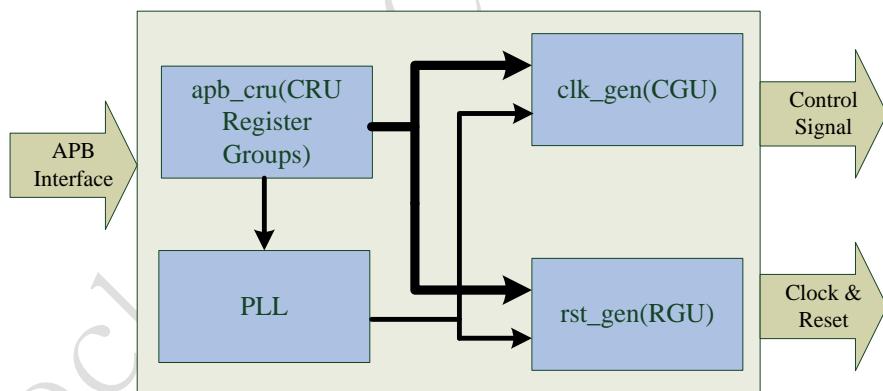


Fig. 3-1 CRU Architecture

3.3 System Clock Solution

The following diagrams show clock architecture (mux and divider information).

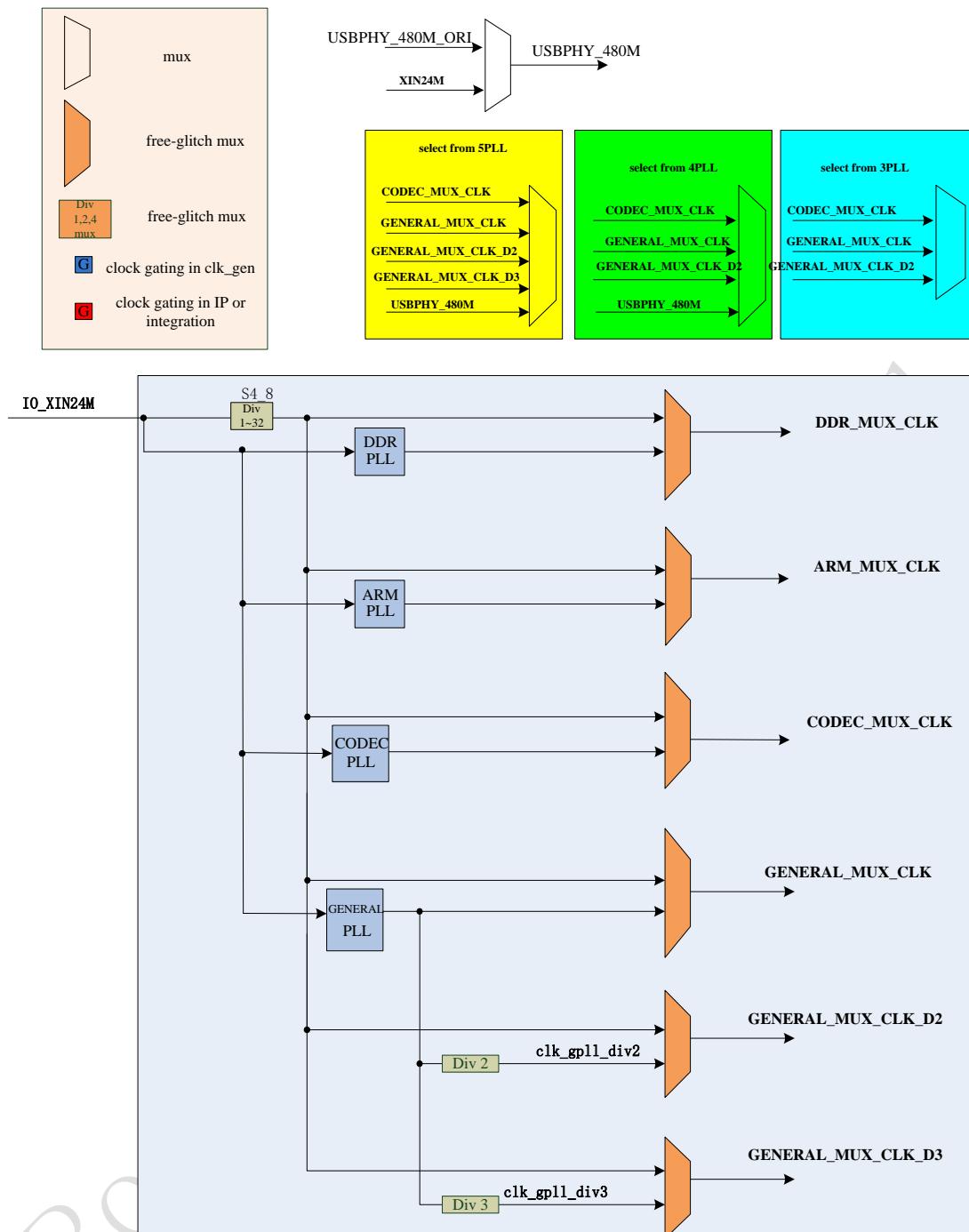
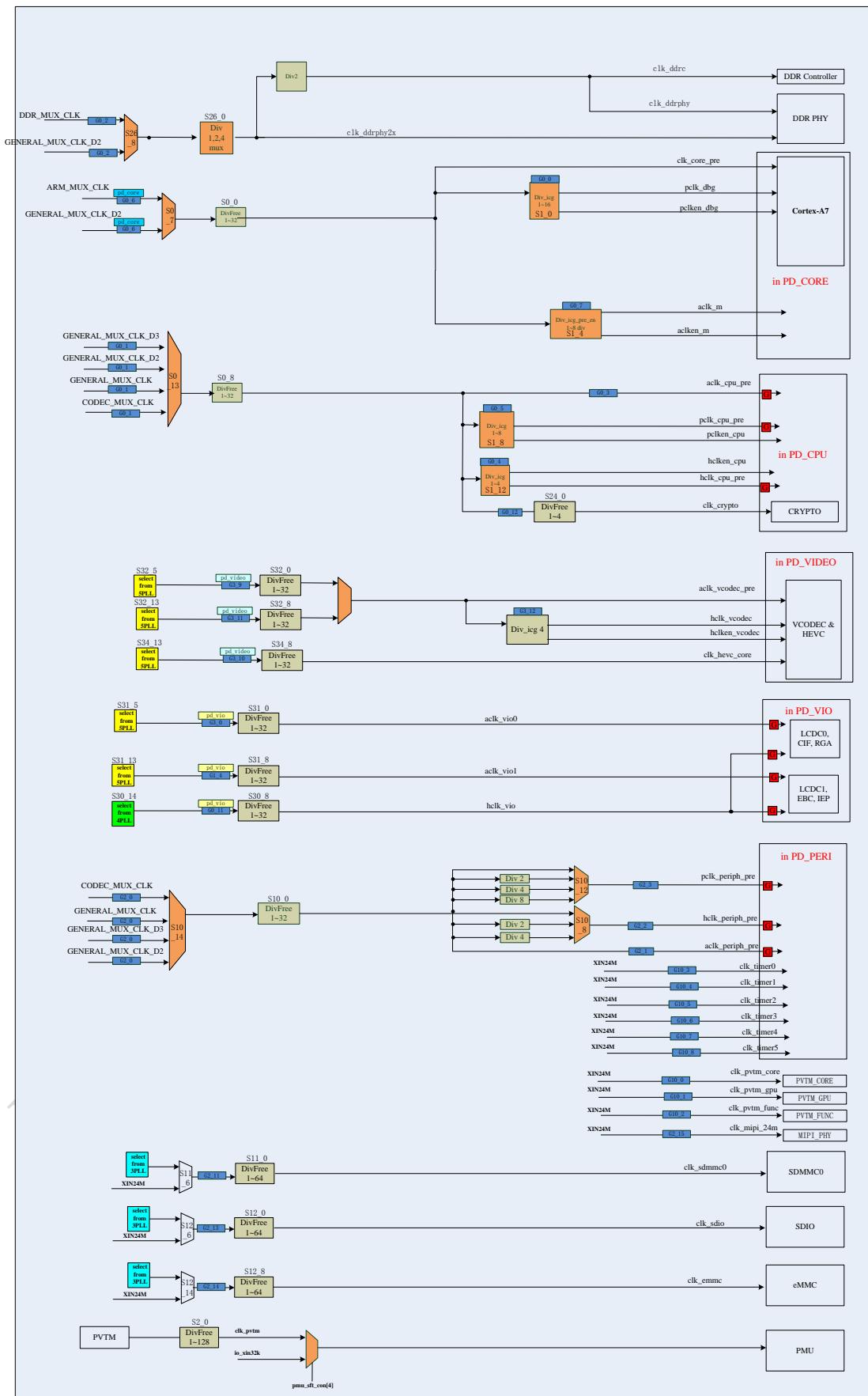


Fig. 3-2 Chip Clock Architecture Diagram 1


Fig. 3-3Chip Clock Architecture Diagram 2

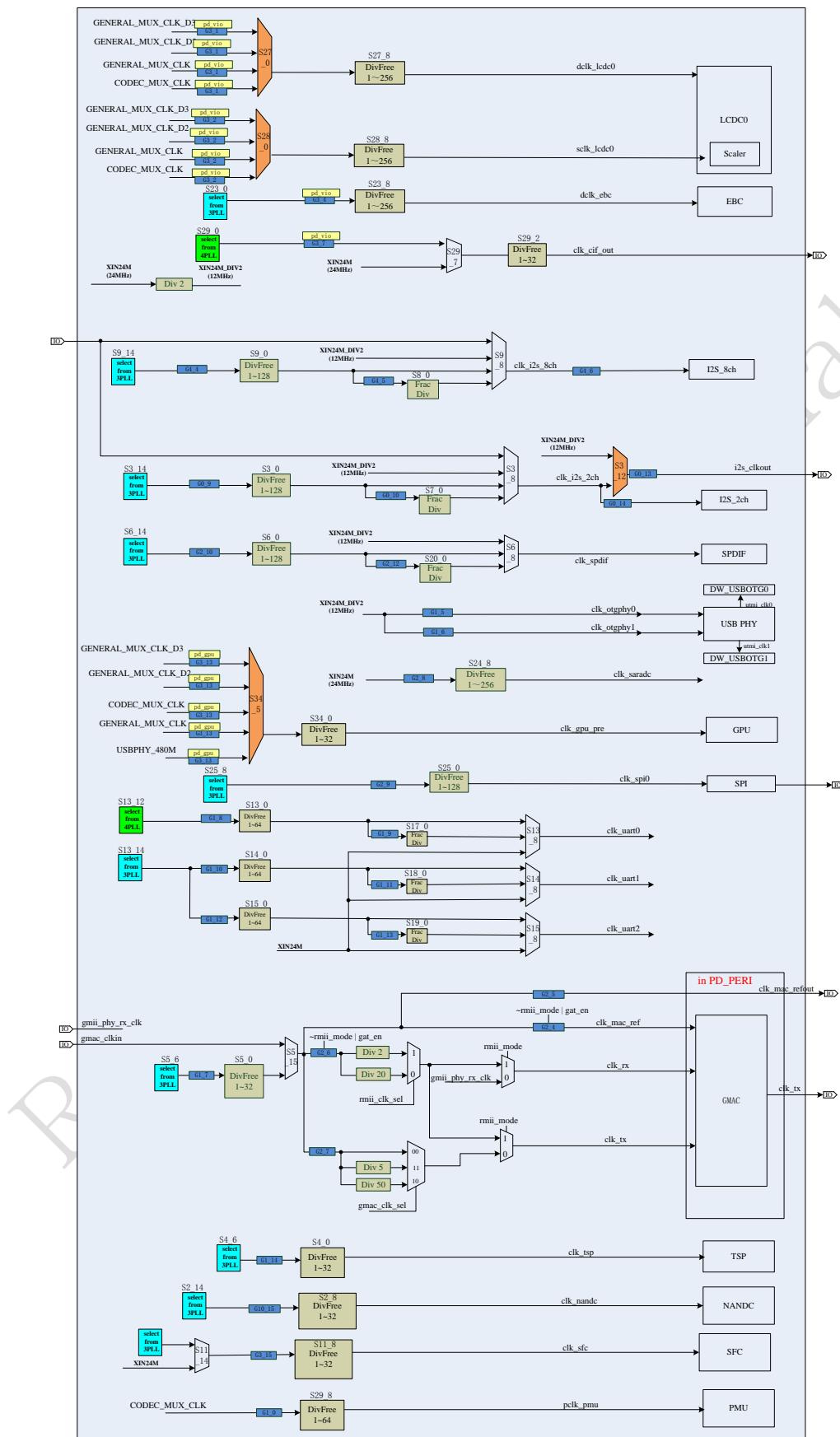


Fig. 3-4 Chip Clock Architecture Diagram 3

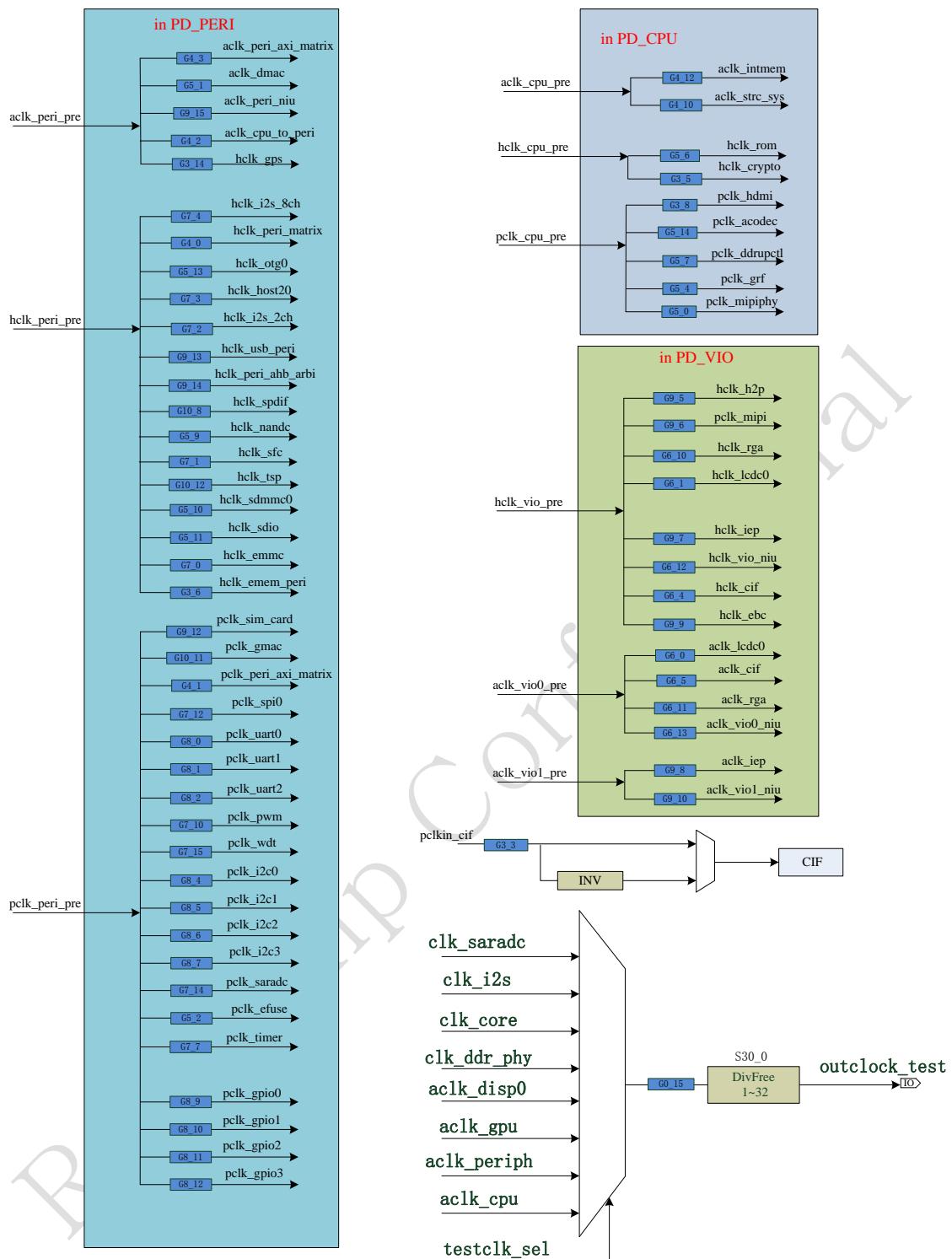


Fig. 3-6 Chip Clock Architecture Diagram 4

Description about input clock

The source of input clock in upper diagrams is listed as following Table.

Table 3-1 Input clock description in clock architecture diagram

Input Clock	Source	IO Name
xin24m	External crystal oscillator (24MHz)	XIN24M

3.4 System Reset Solution

The following diagrams show reset architecture in This device.

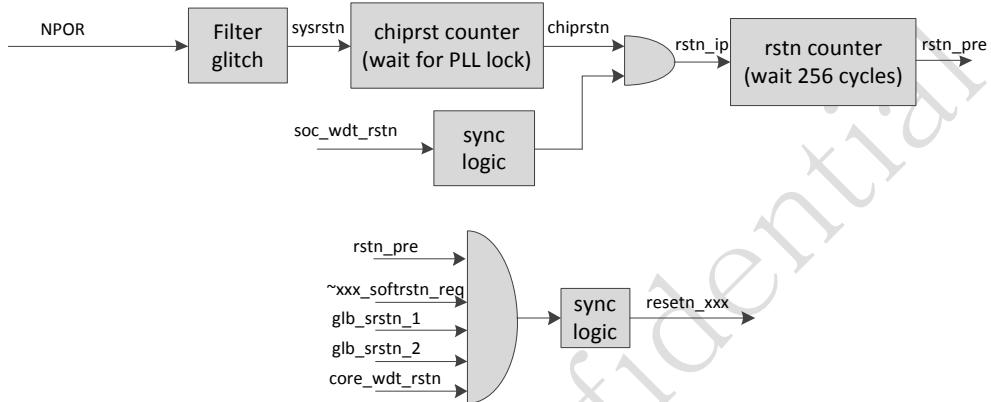


Fig. 3-5Reset Architecture Diagram

Reset source of each reset signal includes hardware reset(NPOR), soc watch dog reset(soc_wdt_rstn), software reset request(~xxx_softrstn_req), global software reset1(glb_srstn_1), global software reset2(glb_srstn_2) and A9 core watch dog reset(core_wdt_rstn).

The 'xxx' of resetn_xxx and xxx_softrstn_req is the module name.

Soc_wdt_rstn is the reset from watch-dog IP in the SoC, but core_wdt_rstn is the reset from A9 core watch-dog block.

Glb_srstn_1 and glb_srstn_2 are the global software reset by programming CRU register. When writing register CRU_GLB_SRST_FST_VALUE as 0xfd9, glb_srstn_1 will be asserted, and when writing register

CRU_GLB_SRST_SND_VALUE as 0xea8, glb_srstn_2 will be asserted. The two software reset will be self-clear by hardware. Glb_srstn_1 will reset the all logic, and Glb_srstn_2 will reset the all logic except GRF and all GPIOs.

3.5 Function Description

There are four PLLs in the chip: ARM PLL, DDR PLL, CODEC PLL and GENERAL PLL, and it supports only onecrystal oscillator: 24MHz. Each PLL can only receive 24MHz oscillator.

Four PLLs all can be set to slow mode or deep slow mode, directly output selectable 24MHz or 32.768kHz. When power on or changing PLL setting, we must force PLL into slow mode to ensure output stable clock.

To maximize the flexibility, some of clocks can select divider source from multy PLLs.

To provide some specific frequency, another solution is integrated: fractional divider. In order to be sure the performance for divided clock, there is some usage limit, we can only get low frequency and divider factor must be larger

than 20.

All clocks can be software gated and all reset can be software generated.

3.6 PLL Introduction

3.6.1 Overview

The chip uses 2.4GHz PLL for all four PLLs. The 2.4GHz PLL is a general purpose, high-performance PLL-based clock generator. The PLL is a multi-function, general purpose frequency synthesizer. Ultra-wide input and output ranges along with best-in-class jitter performance allow the PLL to be used for almost any clocking application. With excellent supply noise immunity, the PLL is ideal for use in noisy mixed signal SoC environments. By combining ultra-low jitter output clocks into a low power, low area, widely programmable design, Silicon Creations can greatly simplify an SoC by enabling a single macro to be used for all clocking applications in the system.

2.4GHz PLL supports the following features:

- Input Frequency Range: 1MHz to 800MHz (Integer Mode) and 10MHz to 800MHz (Fractional Mode)
- Output Frequency Range: 12MHz to 2.4GHz
- 24 bit fractional accuracy, and fractional mode jitter performance to nearly match integer mode performance.
- 4:1 VCO frequency range allows PLL to be optimized for minimum jitter or minimum power.
- Isolated analog supply (2.5V) allows for excellent supply rejection in noisy SoC applications.
- Lock Detect Signal indicates when frequency lock has been achieved.

3.6.2 Block diagram

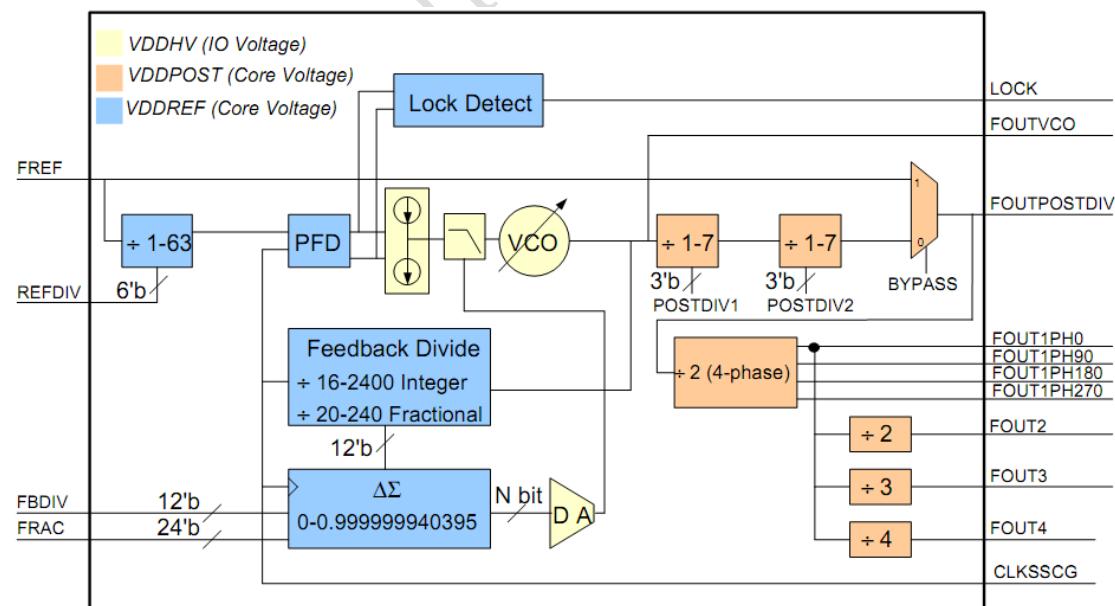


Fig. 3-6PLL Block Diagram

How to calculate the PLL

The Fractional PLL output frequency can be calculated using some simple

formulas. These formulas also embedded within the Fractional PLL Verilog model:

If DSMPD = 1 (DSM is disabled, "integer mode")
 $FOUTVCO = FREF / REFDIV * FBDIV$
 $FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2$

If DSMPD = 0 (DSM is enabled, "fractional mode")
 $FOUTVCO = FREF / REFDIV * (FBDIV + FRAC / 2^{24})$
 $FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2$

Where:

$FOUTVCO$ = Fractional PLL non-divided output frequency
 $FOUTPOSTDIV$ = Fractional PLL divided output frequency (output of second post divider)
 $FREF$ = Fractional PLL input reference frequency
 $REFDIV$ = Fractional PLL input reference clock divider
 $FVCO$ = Frequency of internal VCO
 $FBDIV$ = Integer value programmed into feedback divide
 $FRAC$ = Fractional value programmed into DSM

Changing the PLL Programming

In most cases the PLL programming can be changed on-the-fly and the PLL will simply slew to the new frequency. However, certain changes have the potential to cause glitches on the PLL output clocks. These changes include:

- Switching into or out of BYPASS mode may cause a glitch on FOUTPOSTDIV
- Changing POSTDIV1 or POSTDIV2 may cause a short pulse with width equal to as little as one VCO period on FOUTPOSTDIV
- Changing POSTDIV could cause a shortened pulse on FOUT1PH* or FOUT2/3/4
- Asserting PD or FOUTPOSTDIVPD may cause a glitch on FOUTPOSTDIV

3.7 Register Description

This section describes the control/status registers of the design.

3.7.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_APPL_CON0	0x0000	W	0x000010af	ARM PLL control register0
CRU_APPL_CON1	0x0004	W	0x00001046	ARM PLL control register1
CRU_APPL_CON2	0x0008	W	0x00000001	ARM PLL control register2
CRU_DPLL_CON0	0x0010	W	0x00001064	DDR PLL control register0
CRU_DPLL_CON1	0x0014	W	0x00001043	DDR PLL control register1
CRU_DPLL_CON2	0x0018	W	0x00000001	DDR PLL control register2
CRU_CPLL_CON0	0x0020	W	0x0000107d	Codec PLL control register0
CRU_CPLL_CON1	0x0024	W	0x00001046	Codec PLL control register1
CRU_CPLL_CON2	0x0028	W	0x00000001	Codec PLL control register2
CRU_GPLL_CON0	0x0030	W	0x00004063	General PLL control register0
CRU_GPLL_CON1	0x0034	W	0x00001042	General PLL control register1
CRU_GPLL_CON2	0x0038	W	0x00000001	General PLL control register2
CRU_MODE_CON	0x0040	W	0x00000000	System work mode control register

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL0_CON	0x0044	W	0x00000100	Internal clock select and divide register0
CRU_CLKSEL1_CON	0x0048	W	0x00003113	Internal clock select and divide register1
CRU_CLKSEL2_CON	0x004c	W	0x00000707	Internal clock select and divide register2
CRU_CLKSEL3_CON	0x0050	W	0x0000001f	Internal clock select and divide register3
CRU_CLKSEL4_CON	0x0054	W	0x00000003	Internal clock select and divide register4
CRU_CLKSEL5_CON	0x0058	W	0x00000003	Internal clock select and divide register5
CRU_CLKSEL6_CON	0x005c	W	0x0000021f	Internal clock select and divide register6
CRU_CLKSEL7_CON	0x0060	W	0x0bb8ea60	Internal clock select and divide register7
CRU_CLKSEL8_CON	0x0064	W	0x0bb8ea60	Internal clock select and divide register8
CRU_CLKSEL9_CON	0x0068	W	0x0000001f	Internal clock select and divide register9
CRU_CLKSEL10_CON	0x006c	W	0x0000a100	Internal clock select and divide register10
CRU_CLKSEL11_CON	0x0070	W	0x00000717	Internal clock select and divide register11
CRU_CLKSEL12_CON	0x0074	W	0x00001717	Internal clock select and divide register12
CRU_CLKSEL13_CON	0x0078	W	0x0000121f	Internal clock select and divide register13
CRU_CLKSEL14_CON	0x007c	W	0x0000021f	Internal clock select and divide register14
CRU_CLKSEL15_CON	0x0080	W	0x0000021f	Internal clock select and divide register15
CRU_CLKSEL17_CON	0x0088	W	0x0bb8ea60	Internal clock select and divide register17
CRU_CLKSEL18_CON	0x008c	W	0x0bb8ea60	Internal clock select and divide register18
CRU_CLKSEL19_CON	0x0090	W	0x0bb8ea60	Internal clock select and divide register19
CRU_CLKSEL20_CON	0x0094	W	0x0bb8ea60	Internal clock select and divide register20
CRU_CLKSEL23_CON	0x00a0	W	0x00000100	Internal clock select and divide register23
CRU_CLKSEL24_CON	0x00a4	W	0x00001701	Internal clock select and divide register24
CRU_CLKSEL25_CON	0x00a8	W	0x0000011f	Internal clock select and divide register25
CRU_CLKSEL26_CON	0x00ac	W	0x00000000	Internal clock select and divide register26
CRU_CLKSEL27_CON	0x00b0	W	0x00000301	Internal clock select and divide register27

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL28_CON	0x00b4	W	0x00000301	Internal clock select and divide register28
CRU_CLKSEL29_CON	0x00b8	W	0x00000524	Internal clock select and divide register29
CRU_CLKSEL30_CON	0x00bc	W	0x00000300	Internal clock select and divide register30
CRU_CLKSEL31_CON	0x00c0	W	0x00004040	Internal clock select and divide register31
CRU_CLKSEL32_CON	0x00c4	W	0x00000101	Internal clock select and divide register32
CRU_CLKSEL34_CON	0x00cc	W	0x00004141	Internal clock select and divide register34
CRU_CLKGATE0_CO_N	0x00d0	W	0x00000000	Internal clock gating control register0
CRU_CLKGATE1_CO_N	0x00d4	W	0x00000000	Internal clock gating control register1
CRU_CLKGATE2_CO_N	0x00d8	W	0x00000000	Internal clock gating control register2
CRU_CLKGATE3_CO_N	0x00dc	W	0x00000000	Internal clock gating control register3
CRU_CLKGATE4_CO_N	0x00e0	W	0x00000000	Internal clock gating control register0
CRU_CLKGATE5_CO_N	0x00e4	W	0x00000000	Internal clock gating control register5
CRU_CLKGATE6_CO_N	0x00e8	W	0x00000000	Internal clock gating control register6
CRU_CLKGATE7_CO_N	0x00ec	W	0x00000000	Internal clock gating control register7
CRU_CLKGATE8_CO_N	0x00f0	W	0x00000000	Internal clock gating control register8
CRU_CLKGATE9_CO_N	0x00f4	W	0x00000000	Internal clock gating control register9
CRU_CLKGATE10_C_ON	0x00f8	W	0x00000000	Internal clock gating control register10
CRU_GLB_SRST_FST_VALUE	0x0100	W	0x00000000	The first global software reset config value
CRU_GLB_SRST_SN_D_VALUE	0x0104	W	0x00000000	The second global software reset config value
CRU_SOFRST0_CO_N	0x0110	W	0x00000000	Internal software reset control register0
CRU_SOFRST1_CO_N	0x0114	W	0x00000000	Internal software reset control register1
CRU_SOFRST2_CO_N	0x0118	W	0x00000000	Internal software reset control register2
CRU_SOFRST3_CO_N	0x011c	W	0x00000000	Internal software reset control register3
CRU_SOFRST4_CO_N	0x0120	W	0x00000000	Internal software reset control register4
CRU_SOFRST5_CO_N	0x0124	W	0x00000000	Internal software reset control register5

Name	Offset	Size	Reset Value	Description
CRU_SOFTRST6_CO_N	0x0128	W	0x00000000	Internal software reset control register6
CRU_SOFTRST7_CO_N	0x012c	W	0x00000000	Internal software reset control register7
CRU_SOFTRST8_CO_N	0x0130	W	0x00000000	Internal software reset control register8
CRU_MISC_CON	0x0134	W	0x00008000	SCU control register
CRU_GLB_CNT_TH	0x0140	W	0x3a980064	global reset wait counter threshold
CRU_GLB_RST_ST	0x0150	W	0x00000000	global reset status
CRU_SDMMC_CON0	0x01c0	W	0x00000004	sdmmc control0
CRU_SDMMC_CON1	0x01c4	W	0x00000000	sdmmc control1
CRU_SDIO_CON0	0x01c8	W	0x00000004	sdio0 control0
CRU_SDIO_CON1	0x01cc	W	0x00000000	sdio0 control1
CRU_EMMC_CON0	0x01d8	W	0x00000004	emmc control0
CRU_EMMC_CON1	0x01dc	W	0x00000000	emmc control1
CRU_PLL_PRG_EN	0x01f0	W	0x00000000	PLL program enable

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

3.7.2 Detail Register Description

CRU_APPL_CON0

Address: Operational Base + offset (0x0000)

ARM PLL control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	bp pll bypass
14:12	RW	0x1	postdiv1 PLL factor postdiv1
11:0	RW	0x0af	fbdv PLL factor fbdv

CRU_APPL_CON1

Address: Operational Base + offset (0x0004)

ARM PLL control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask

Bit	Attr	Reset Value	Description
15	RW	0x0	rstmode PLL Reset select 0 : internal reset 1 : software reset
14	RW	0x0	rst PLL Software Reset 0 : normal 1 : reset
13	RW	0x0	pd PLL software power down, active high
12	RW	0x1	dsmpd when 1, PLL work at integer mode when 0, PLL work at frac mode
11	RO	0x0	reserved
10	RW	0x0	lock PLL lock status
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x06	refdiv PLL factor refdiv

CRU_APOLL_CON2

Address: Operational Base + offset (0x0008)

ARM PLL control register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down, active high
26	RW	0x0	foutvcopd buffered VCO clock power down, active high
25	RW	0x0	foutpostdivpd post divide power down, active high
24	RW	0x0	dacpd PLL cancellation DAC power down, active high
23:0	RW	0x000001	frac PLL factor frac

CRU_DPLL_CON0

Address: Operational Base + offset (0x0010)

DDR PLL control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	bp pll bypass

Bit	Attr	Reset Value	Description
14:12	RW	0x1	postdiv1 PLL factor postdiv1
11:0	RW	0x064	fbdv PLL factor fbdv

CRU_DPLL_CON1

Address: Operational Base + offset (0x0014)

DDR PLL control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	rstmode PLL Reset select 0 : internal reset 1 : software reset
14	RW	0x0	rst PLL Software Reset 0 : normal 1 : reset
13	RW	0x0	pd PLL software power down, active high
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11	RO	0x0	reserved
10	RW	0x0	lock PLL lock status
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x03	refdiv PLL factor refdiv

CRU_DPLL_CON2

Address: Operational Base + offset (0x0018)

DDR PLL control register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down, active high
26	RW	0x0	foutvcopd buffered VCO clock power down, active high
25	RW	0x0	foutpostdivpd post divide power down, active high
24	RW	0x0	dacpd PLL cancellation DAC power down, active high

Bit	Attr	Reset Value	Description
23:0	RW	0x000001	frac PLL factor frac

CRU_CPLL_CON0

Address: Operational Base + offset (0x0020)

Codec PLL control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	bp pll bypass
14:12	RW	0x1	postdiv1 PLL factor postdiv1
11:0	RW	0x07d	fbdv PLL factor fbdv

CRU_CPLL_CON1

Address: Operational Base + offset (0x0024)

Codec PLL control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	rstmode PLL Reset select 0 : internal reset 1 : software reset
14	RW	0x0	rst PLL Software Reset 0 : normal 1 : reset
13	RW	0x0	pd PLL software power down, active high
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11	RO	0x0	reserved
10	RW	0x0	lock PLL lock status
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x06	refdiv PLL factor refdiv

CRU_CPLL_CON2

Address: Operational Base + offset (0x0028)

Codec PLL control register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down, active high
26	RW	0x0	foutvcopd buffered VCO clock power down, active high
25	RW	0x0	foutpostdivpd post divide power down, active high
24	RW	0x0	dacpd PLL cancellation DAC power down, active high
23:0	RW	0x000001	frac PLL factor frac

CRU_GPLL_CON0

Address: Operational Base + offset (0x0030)

General PLL control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	bp pll bypass
14:12	RW	0x4	postdiv1 PLL factor postdiv1
11:0	RW	0x063	fbdv PLL factor fbdv

CRU_GPLL_CON1

Address: Operational Base + offset (0x0034)

General PLL control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	rstmode PLL Reset select 0 : internal reset 1 : software reset
14	RW	0x0	rst PLL Software Reset 0 : normal 1 : reset

Bit	Attr	Reset Value	Description
13	RW	0x0	pd PLL software power down, active high
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11	RO	0x0	reserved
10	RW	0x0	lock PLL lock status
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x02	refdiv PLL factor refdiv

CRU_GPLL_CON2

Address: Operational Base + offset (0x0038)

General PLL control register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down, active high
26	RW	0x0	foutvcopd buffered VCO clock power down, active high
25	RW	0x0	foutpostdivpd post divide power down, active high
24	RW	0x0	dacpd PLL cancellation DAC power down, active high
23:0	RW	0x000001	frac PLL factor frac

CRU_MODE_CON

Address: Operational Base + offset (0x0040)

System work mode control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	gpll_work_mode GENERAL PLL work mode select 1'b0: Slow mode, clock from external 24MHz OSC (default) 1'b1: Normal mode, clock from PLL output
11:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	cpll_work_mode CODEC PLL work mode select 1'b0: Slow mode, clock from external 24MHz OSC (default) 1'b1: Normal mode, clock from PLL output
7:5	RO	0x0	reserved
4	RW	0x0	dpll_work_mode DDR PLL work mode select 1'b0: Slow mode, clock from external 24MHz OSC (default) 1'b1: Normal mode, clock from PLL output
3:1	RO	0x0	reserved
0	RW	0x0	apll_work_mode ARM PLL work mode select 1'b0: Slow mode, clock from external 24MHz OSC (default) 1'b1: Normal mode, clock from PLL output

CRU_CLKSEL0_CON

Address: Operational Base + offset (0x0044)

Internal clock select and divide register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:13	RW	0x0	cpu_clk_pll_sel pd_cpu aclk_cpu pll source selection 2'b00: select CODEC PLL 2'b01: select GENERAL PLL 2'b10: select GENERAL PLL DIV2 2'b11: select GENERAL PLL DIV3
12:8	RW	0x01	aclk_cpu_div_con aclk_cpu clock divider frequency $aclk_{cpu}=cpu_{clk_src}/(aclk_{cpu_div_con}+1)$
7	RW	0x0	core_clk_pll_sel core clock pll source selection 1'b0: select ARM PLL 1'b1: select GENERAL PLL div2
6:5	RO	0x0	reserved
4:0	RW	0x00	a7_core_div_con Control A7 core clock divider frequency $clk_{core}=core_{clk_src}/(a7_{core_div_con}+1)$

CRU_CLKSEL1_CON

Address: Operational Base + offset (0x0048)

Internal clock select and divide register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x3	cpu_pclk_div_con Control cpu subsystem APB clock divider frequency $pclk_{cpu} = pclk_{src}/(cpu_pclk_div_con + 1)$
11:10	RO	0x0	reserved
9:8	RW	0x1	cpu_hclk_div_con Control cpu subsystem AHB clock divider frequency $hclk_{cpu} = pclk_{src}/(cpu_hclk_div_con + 1)$
7	RO	0x0	reserved
6:4	RW	0x1	core_aclk_div_con Control A9 core axi clock divider frequency $aclk_{core} = core_clk_src/(core_aclk_div_con + 1)$
3:0	RW	0x3	pclk_dbg_div_con pclk_dbg div control $pclk_{dbg} = pclk_{src}/(pclk_{dbg_div_con} + 1)$

CRU_CLKSEL2_CON

Address: Operational Base + offset (0x004c)

Internal clock select and divide register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	nandc_clk_pll_sel nandc pll source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:8	RW	0x07	nandc_div_con Control NANDC clock divider frequency $clk_{nandc} = clk_{src}/(nandc_div_con + 1)$
7	RO	0x0	reserved
6:0	RW	0x07	pvtm_div_con func pvtm clock divider frequency $clk_{pvtm} = clk_{src}/(pvtm_div_con + 1)$

CRU_CLKSEL3_CON

Address: Operational Base + offset (0x0050)

Internal clock select and divide register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	i2s_2ch_pll_sel Control I2S_2ch PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13	RO	0x0	reserved
12	RW	0x0	i2s_2ch_clkout_sel I2S_2ch output clock selection 1'b0: select cru generated clock 1'b1: select io input clock
11:10	RO	0x0	reserved
9:8	RW	0x0	i2s_2ch_clk_sel Control I2S_2ch clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select io input clock 2'b11: select 12MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x1f	i2s_2ch_pll_div_con Control I2S_2ch PLL output divider frequency $i2s_div_clk = i2s_div_src / (i2s_pll_div_con + 1)$

CRU_CLKSEL4_CON

Address: Operational Base + offset (0x0054)

Internal clock select and divide register4

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	clk_24m_div_con Control clk_24m divider frequency $24m_div_clk = clk_24m / (clk_24m_div_con + 1)$
7:6	RW	0x0	tsp_pll_sel Control TSP clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
5	RO	0x0	reserved
4:0	RW	0x03	clk_tsp_div_con Control clk_tsp divider frequency $tsp_div_clk = tsp_div_src / (clk_tsp_div_con + 1)$

CRU_CLKSEL5_CON

Address: Operational Base + offset (0x0058)

Internal clock select and divide register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	rmii_extclk_sel Control RMII external clock selection 1'b0: select internal divider clock 1'b1: select external input clock
14:8	RO	0x0	reserved
7:6	RW	0x0	mac_pll_sel Control MAC clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
5	RO	0x0	reserved
4:0	RW	0x03	clk_mac_div_con Control clk_mac divider frequency $mac_div_clk = mac_div_src / (clk_mac_div_con + 1)$

CRU_CLKSEL6_CON

Address: Operational Base + offset (0x005c)

Internal clock select and divide register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	spdif_pll_sel Control SPDIF PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13:10	RO	0x0	reserved
9:8	RW	0x2	spdif_clk_sel Control SPDIF clock work frequency selection 2'b00: select divider output from pll divider 2'b01: select divider output from fraction divider 2'b10: select 12MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x1f	spdif_pll_div_con Control SPDIF PLL output divider frequency spdif_div_clk=spdif_div_src/(spdif_pll_div_con+1)

CRU_CLKSEL7_CON

Address: Operational Base + offset (0x0060)

Internal clock select and divide register7

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	i2s_2ch_frac_factor Control I2S 2channel fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL8_CON

Address: Operational Base + offset (0x0064)

Internal clock select and divide register8

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	i2s_8ch_frac_factor Control I2S 8channel fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL9_CON

Address: Operational Base + offset (0x0068)

Internal clock select and divide register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	i2s_8ch_pll_sel Control I2S_8ch PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13:10	RO	0x0	reserved
9:8	RW	0x0	i2s_8ch_clk_sel Control I2S_8ch clock work frequency selection 2'b00: select divider output from pll divider 2'b01: select divider output from fraction divider 2'b10: select io input clock 2'b11: select 12MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x1f	i2s_8ch_pll_div_con Control I2S_8ch PLL output divider frequency $i2s_div_clk = i2s_div_src / (i2s_pll_div_con + 1)$

CRU_CLKSEL10_CON

Address: Operational Base + offset (0x006c)

Internal clock select and divide register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x2	peri_pll_sel Control peripheral clock PLL source selection 2'b00: select general pll clock 2'b01: select codec pll clock 2'b10: select general div2 pll clock 2'b11: select general div3 pll clock

Bit	Attr	Reset Value	Description
13:12	RW	0x2	peri_pclk_div_con Control the divider ratio between aclk_periph and pclk_periph 2'b00: aclk_periph:pclk_periph = 1:1 2'b01: aclk_periph:pclk_periph = 2:1 2'b10: aclk_periph:pclk_periph = 4:1 2'b11: aclk_periph:pclk_periph = 8:1
11:10	RO	0x0	reserved
9:8	RW	0x1	peri_hclk_div_con Control the divider ratio between aclk_periph and hclk_periph 2'b00: aclk_periph:hclk_periph = 1:1 2'b01: aclk_periph:hclk_periph = 2:1 2'b10: aclk_periph:hclk_periph = 4:1
7:5	RO	0x0	reserved
4:0	RW	0x00	peri_aclk_div_con Control periphral clock divider frequency $aclk_{periph} = periph_{clk_src} / (peri_aclk_div_con + 1)$

CRU_CLKSEL11_CON

Address: Operational Base + offset (0x0070)

Internal clock select and divide register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	sfc_clk_pll_sel sfc pll source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select 24M clock
13	RO	0x0	reserved
12:8	RW	0x00	sfc_div_con Control SFC clock divider frequency $clk_{sfc} = sfc_{clk_src} / (sfc_div_con + 1)$
7:6	RW	0x0	mmc0_pll_sel Control mmc clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select 24M clock

Bit	Attr	Reset Value	Description
5:0	RW	0x17	mmc0_div_con Control SDMMC0 divider frequency $\text{clk_sdmmc0} = \text{general_pll_clk}/(\text{mmc0_div_con}+1)$

CRU_CLKSEL12_CON

Address: Operational Base + offset (0x0074)

Internal clock select and divide register12

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	emmc_pll_sel Control eMMC clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select 24M clock
13:8	RW	0x17	emmc_div_con Control EMMC divider frequency $\text{clk_emmc} = \text{general_pll_clk}/(\text{emmc_div_con}+1)$
7:6	RW	0x0	sdio_pll_sel Control SDIO clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select 24M clock
5:0	RW	0x17	sdio_div_con Control SDIO divider frequency $\text{clk_sdio} = \text{general_pll_clk}/(\text{sdio_div_con}+1)$

CRU_CLKSEL13_CON

Address: Operational Base + offset (0x0078)

Internal clock select and divide register13

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x0	uart12_pll_sel Control UART1 and UART2 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13:12	RW	0x1	uart0_pll_sel Control UART0 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select USBPHY 480M clock
11:10	RO	0x0	reserved
9:8	RW	0x2	uart0_clk_sel Control UART0 clock work frequency selection 2'b00: select divider output from pll divider 2'b01: select divider output from fraction divider 2'b10: select 24MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x1f	uart0_div_con Control UART0 divider frequency $\text{clk_uart0} = \text{uart_clk_src} / (\text{uart0_div_con} + 1)$

CRU_CLKSEL14_CON

Address: Operational Base + offset (0x007c)

Internal clock select and divide register14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x2	uart1_clk_sel Control UART1 clock work frequency selection 2'b00: select divider output from pll divider 2'b01: select divider output from fraction divider 2'b10: select 24MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x1f	uart1_div_con Control UART1 divider frequency $\text{clk_uart1} = \text{uart_clk_src} / (\text{uart1_div_con} + 1)$

CRU_CLKSEL15_CON

Address: Operational Base + offset (0x0080)

Internal clock select and divide register15

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x2	uart2_clk_sel Control UART2 clock work frequency selection 2'b00: select divider output from pll divider 2'b01: select divider output from fraction divider 2'b10: select 24MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x1f	uart2_div_con Control UART2 divider frequency clk_uart2=uart_clk_src/(uart2_div_con+1)

CRU_CLKSEL17_CON

Address: Operational Base + offset (0x0088)

Internal clock select and divide register17

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart0_frac_factor Control UART0 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL18_CON

Address: Operational Base + offset (0x008c)

Internal clock select and divide register18

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart1_frac_factor Control UART1 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL19_CON

Address: Operational Base + offset (0x0090)

Internal clock select and divide register19

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart2_frac_factor Control UART2 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL20_CON

Address: Operational Base + offset (0x0094)

Internal clock select and divide register20

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	spdif_frac_factor Control SPDIF fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL23_CON

Address: Operational Base + offset (0x00a0)

Internal clock select and divide register23

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x01	ebc_dclk_div_con Control EBC clock divider frequency $dclk_ebc=ebc_dclk_src/(ebc_dclk_div_con+1)$
7:2	RO	0x0	reserved
1:0	RW	0x0	ebc_dclk_pll_sel Control EBC display clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock

CRU_CLKSEL24_CON

Address: Operational Base + offset (0x00a4)

Internal clock select and divide register24

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x17	saradc_div_con Control SARADC clock divider frequency $clk_saradc=24MHz/(saradc_div_con+1)$
7:2	RO	0x0	reserved
1:0	RW	0x1	crypto_div_con crypto clock divider frequency $clk=clk/(div_con+1)$

CRU_CLKSEL25_CON

Address: Operational Base + offset (0x00a8)

Internal clock select and divide register25

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x1	spi_clk_pll_sel SPI clock pll source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
7	RO	0x0	reserved
6:0	RW	0x1f	spi0_div_con Control SPI0 clock divider frequency $\text{clk_spi0} = \text{general_pll_clk} / (\text{spi0_div_con} + 1)$

CRU_CLKSEL26_CON

Address: Operational Base + offset (0x00ac)

Internal clock select and divide register26

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	ddr_clk_pll_sel DDR clock pll source selection 1'b0: select DDR PLL 1'b1: select GENERAL PLL div2
7:2	RO	0x0	reserved
1:0	RW	0x0	ddr_div_sel Control DDR divider frequency 2'b00: clk_ddr_src:clk_ddrphy = 1:1 2'b01: clk_ddr_src:clk_ddrphy = 2:1 2'b10: clk_ddr_src:clk_ddrphy = 4:1

CRU_CLKSEL27_CON

Address: Operational Base + offset (0x00b0)

Internal clock select and divide register27

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:8	RW	0x03	lcdc0_dclk_div_con Control LCDC clock divider frequency $dclk_{lcdc0} = lcdc_dclk_src / (lcdc0_dclk_div_con + 1)$
7:2	RO	0x0	reserved
1:0	RW	0x1	lcdc0_dclk_pll_sel Control LCDC display clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select general pll div3 clock

CRU_CLKSEL28_CON

Address: Operational Base + offset (0x00b4)

Internal clock select and divide register28

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x03	lcdc_sclk_div_con Control LCDC display clock divider frequency $sclk_{lcdc} = lcdc_sclk_src / (lcdc_sclk_div_con + 1)$
7:2	RO	0x0	reserved
1:0	RW	0x1	lcdc_sclk_pll_sel Control LCDC display clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select general pll div3 clock

CRU_CLKSEL29_CON

Address: Operational Base + offset (0x00b8)

Internal clock select and divide register29

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RO	0x0	reserved
13:8	RW	0x05	pmu_div_con Control PMU clock divider frequency $clk_{pmu}=pmu_{clk_src}/(pmu_div_con+1)$
7	RW	0x0	cif0_clk_sel Control CIF0 clock selection 1'b0: select from PLL divider output 1'b1: select from 24MHz osc
6:2	RW	0x09	cif0_div_con Control CIF0 clock divider frequency $clk_{cif0}=cif0_{clk_src}/(cif0_div_con+1)$
1:0	RW	0x0	cif_pll_sel Control CIF clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: usbphy480M

CRU_CLKSEL30_CON

Address: Operational Base + offset (0x00bc)

Internal clock select and divide register30

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	vio_hclk_pll_sel Control VIO AHB clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select usbphy 480m clock
13	RO	0x0	reserved
12:8	RW	0x03	vio_hclk_div_con Control VIO AHB clock divider frequency $hclk_{vio}=vio_{hclk_src}/(vio_hclk_div_con+1)$
7	RW	0x0	cif0_clkin_inv_sel CIF0 input clock inverter selection 1'b0: select not invert 1'b1: select invert
6:5	RO	0x0	reserved
4:0	RW	0x00	testout_div_con testout clock divider frequency $clk_{testout}=testout_{clk_src}/(testout_div_con+1)$

CRU_CLKSEL31_CON

Address: Operational Base + offset (0x00c0)

Internal clock select and divide register31

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x2	lc当地1_aclk_pll_sel Control LCD1 AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock
12:8	RW	0x00	lc当地1_aclk_div_con Control LCD1 AXI clock divider frequency $aclk_{LCD1} = aclk_{src} / (aclk_{LCD1_div_con} + 1)$
7:5	RW	0x2	lc当地0_aclk_pll_sel Control LCD0 AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock
4:0	RW	0x00	lc当地0_aclk_div_con Control LCD0 AXI clock divider frequency $aclk_{LCD0} = aclk_{src} / (aclk_{LCD0_div_con} + 1)$

CRU_CLKSEL32_CON

Address: Operational Base + offset (0x00c4)

Internal clock select and divide register32

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x0	vdpu_aclk_pll_sel Control VDPU AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock

Bit	Attr	Reset Value	Description
12:8	RW	0x01	<p>vdpu_aclk_div_con Control VDPU AXI clock divider frequency $aclk_{vdpu} = vdp_{aclk_src} / (vdpu_{aclk_div_con} + 1)$</p>
7:5	RW	0x0	<p>vepu_aclk_pll_sel Control VEPU AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock</p>
4:0	RW	0x01	<p>vepu_aclk_div_con Control VEPU AXI clock divider frequency $aclk_{vepu} = vepu_{aclk_src} / (vepu_{aclk_div_con} + 1)$</p>

CRU_CLKSEL34_CON

Address: Operational Base + offset (0x00cc)

Internal clock select and divide register34

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit</p>
15:13	RW	0x2	<p>hevc_core_clk_pll_sel HEVC CORE clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock</p>
12:8	RW	0x01	<p>hevc_core_clk_div_con HEVC CORE clock divider frequency $clk_{hevc_core} = hevc_{core_clk_src} / (hevc_{core_clk_div_con} + 1)$</p>
7:5	RW	0x2	<p>gpu_aclk_pll_sel Control GPU AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock</p>

Bit	Attr	Reset Value	Description
4:0	RW	0x01	<p>gpu_aclk_div_con Control GPU AXI clock divider frequency</p> <p>aclk_gpu=gpu_aclk_src/(gpu_aclk_div_con+1)</p>

CRU_CLKGATE0_CON

Address: Operational Base + offset (0x00d0)

Internal clock gating control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit</p>
15	RW	0x0	<p>testclk_gate_en Test output clock disable When HIGH, disable clock</p>
14	RW	0x0	<p>clk_i2s_2ch_gate_en I2S_2ch clock disable. When HIGH, disable clock</p>
13	RW	0x0	<p>clk_i2s_2ch_out_gate_en I2S_2ch output clock disable. When HIGH, disable clock</p>
12	RW	0x0	<p>clk_crypto_gate_en crypto clock disable. When HIGH, disable clock</p>
11	RW	0x0	<p>hclk_disp_gate_en display AHB clock disable. When HIGH, disable clock</p>
10	RW	0x0	<p>clk_i2s_2ch_frac_src_gate_en I2S_2ch fraction divider source clock disable. When HIGH, disable clock</p>
9	RW	0x0	<p>clk_i2s_2ch_src_gate_en I2S_2ch source clock disable. When HIGH, disable clock</p>
8	RO	0x0	reserved
7	RW	0x0	<p>aclk_core_gate_en ARM core axi clock(aclk_core) disable. When HIGH, disable clock</p>
6	RW	0x0	<p>core_src_clk_gate_en CORE source clock path clock disable. When HIGH, disable clock</p>
5	RW	0x0	<p>pclk_cpu_gate_en CPU system APB clock(pclk_cpu_pre) disable. When HIGH, disable clock</p>
4	RW	0x0	<p>hclk_cpu_gate_en CPU system AHB clock(hclk_cpu_pre) disable. When HIGH, disable clock</p>

Bit	Attr	Reset Value	Description
3	RW	0x0	aclk_cpu_gate_en CPU system AXI clock(aclk_cpu_pre) disable. When HIGH, disable clock
2	RW	0x0	clk_ddrphy_src_gate_en DDR PHY clock(clk_ddrphy) disable. When HIGH, disable clock
1	RW	0x0	cpu_src_clk_gate_en CPU clock source clock disable. When HIGH, disable clock
0	RW	0x0	clk_core_periph_gate_en ARM core peripheral clock(clk_core_periph) disable. When HIGH, disable clock

CRU_CLKGATE1_CON

Address: Operational Base + offset (0x00d4)

Internal clock gating control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	clk_tsp_gate_en clk_tsp clock disable. When HIGH, disable clock
13	RW	0x0	clk_uart2_frac_src_gate_en UART2 fraction divider source clock disable. When HIGH, disable clock
12	RW	0x0	clk_uart2_src_gate_en UART2 source clock disable. When HIGH, disable clock
11	RW	0x0	clk_uart1_frac_src_gate_en UART1 fraction divider source clock disable. When HIGH, disable clock
10	RW	0x0	clk_uart1_src_gate_en UART1 source clock disable. When HIGH, disable clock
9	RW	0x0	clk_uart0_frac_src_gate_en UART0 fraction divider source clock disable. When HIGH, disable clock
8	RW	0x0	clk_uart0_src_gate_en UART0 source clock disable. When HIGH, disable clock
7	RW	0x0	clk_mac_src_gate_en clk_mac source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_otgphy1_gate_en OTGPHY1 clock(clk_otgphy1) disable. When HIGH, disable clock
5	RW	0x0	clk_otgphy0_gate_en OTGPHY0 clock(clk_otgphy0) disable. When HIGH, disable clock
4	RW	0x0	aclk_vio1_src_gate_en aclk_vio1_src clock disable. When HIGH, disable clock
3	RW	0x0	clk_jtag_gate_en JTAG clock disable. When HIGH, disable clock
2:1	RO	0x0	reserved
0	RW	0x0	pclk_pmu_src_gate_en pclk_pmu src clock disable. When HIGH, disable clock

CRU_CLKGATE2_CON

Address: Operational Base + offset (0x00d8)

Internal clock gating control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_mipiphy_24m_gate_en MIPI PHY 24M clock disable. When HIGH, disable clock
14	RW	0x0	clk_emmc_src_gate_en EMMC source clock disable. When HIGH, disable clock
13	RW	0x0	clk_sdio_src_gate_en SDIO source clock disable. When HIGH, disable clock
12	RW	0x0	clk_spdif_frac_src_gate_en SPDIF fraction divider source clock disable. When HIGH, disable clock
11	RW	0x0	clk_mmc0_src_gate_en SDMMC0 source clock disable. When HIGH, disable clock
10	RW	0x0	clk_spdif_src_gate_en SPDIF source clock disable. When HIGH, disable clock
9	RW	0x0	clk_spi0_src_gate_en SPI0 source clock disable. When HIGH, disable clock
8	RW	0x0	clk_saradc_src_gate_en SARADC source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_mac_tx_gate_en clk_mac_tx clock disable. When HIGH, disable clock
6	RW	0x0	clk_mac_rx_gate_en clk_mac_rx clock disable. When HIGH, disable clock
5	RW	0x0	clk_mac_refout_gate_en clk_mac_refout clock disable. When HIGH, disable clock
4	RW	0x0	clk_mac_ref_gate_en clk_mac_ref clock disable. When HIGH, disable clock
3	RW	0x0	pclk_periph_gate_en PERIPH system APB clock(pclk_periph) disable. When HIGH, disable clock
2	RW	0x0	hclk_periph_gate_en PERIPH system AHB clock(hclk_periph) disable. When HIGH, disable clock
1	RW	0x0	aclk_periph_gate_en PERIPH system AXI clock(aclk_periph) disable. When HIGH, disable clock
0	RW	0x0	clk_periph_src_gate_en PERIPH system source clock disable. When HIGH, disable clock

CRU_CLKGATE3_CON

Address: Operational Base + offset (0x00dc)

Internal clock gating control register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_sfc_gate_en sfc clock disable. When HIGH, disable clock
14	RW	0x0	hclk_gps_gate_en GPS AHB bus source axi clock disable. When HIGH, disable clock
13	RW	0x0	aclk_gpu_src_gate_en GPU AXI source clock disable. When HIGH, disable clock
12	RW	0x0	hclk_vdpu_gate_en VDPU AHB source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	aclk_vdpu_src_gate_en VDPU AXI source clock disable. When HIGH, disable clock
10	RW	0x0	clk_hevc_core_src_gate_en HEVC CORE clk source clock(clk_cif0_out) disable. When HIGH, disable clock
9	RW	0x0	aclk_vepu_src_gate_en VEPU AXI source clock disable. When HIGH, disable clock
8	RW	0x0	pclk_hdmi_gate_en HDMI APB bus clock disable. When HIGH, disable clock
7	RW	0x0	clk_cif_out_src_gate_en CIF out clk source clock(clk_cif0_out) disable. When HIGH, disable clock
6	RW	0x0	hclk_emem_peri_gate_en hclk_emem_peri souce clock disable. When HIGH, disable clock
5	RW	0x0	hclk_crypto_gate_en hclk_crypto clock disable. When HIGH, disable clock
4	RW	0x0	dclk_ebc_src_gate_en EBC DCLK souce clock disable. When HIGH, disable clock
3	RW	0x0	pclkin_cif_gate_en CIF pix input clk source clock disable. When HIGH, disable clock
2	RW	0x0	sclk_lcdc0_src_gate_en LCDCO SCLK souce clock disable. When HIGH, disable clock
1	RW	0x0	dclk_lcdc0_src_gate_en LCDCO DCLK souce clock disable. When HIGH, disable clock
0	RW	0x0	aclk_vio0_src_gate_en VIO0 AXI source clock disable. When HIGH, disable clock

CRU_CLKGATE4_CON

Address: Operational Base + offset (0x00e0)

Internal clock gating control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	aclk_intmem_gate_en Internal memory AXI clock(aclk_intmem) disable. When HIGH, disable clock
11	RO	0x0	reserved
10	RW	0x0	aclk_strc_sys_gate_en CPU Structure system AXI clock disable. When HIGH, disable clock
9:7	RO	0x0	reserved
6	RW	0x0	clk_i2s_8ch_gate_en I2S_8ch clock disable. When HIGH, disable clock
5	RW	0x0	clk_i2s_8ch_frac_src_gate_en I2S_8ch fraction divider source clock disable. When HIGH, disable clock
4	RW	0x0	clk_i2s_8ch_src_gate_en I2S_8ch source clock disable. When HIGH, disable clock
3	RW	0x0	aclk_peri_axi_matrix_gate_en PERIPH matrix CPU AXI clock(aclk_peri_axi_matrix) disable. When HIGH, disable clock
2	RW	0x0	aclk_cpu_peri_gate_en PERIPH CPU AXI clock(aclk_cpu_peri) disable. When HIGH, disable clock
1	RW	0x0	pclk_peri_axi_matrix_gate_en PERIPH matrix CPU APB clock(pclk_peri_axi_matrix) disable. When HIGH, disable clock
0	RW	0x0	hclk_peri_axi_matrix_gate_en PERIPH matrix CPU AHB clock(hclk_peri_axi_matrix) disable. When HIGH, disable clock

CRU_CLKGATES5_CON

Address: Operational Base + offset (0x00e4)

Internal clock gating control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	pclk_acodec_gate_en audio codec APB clock disable. When HIGH, disable clock
13	RW	0x0	hclk_otg0_gate_en USB OTG PHY0 AHB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
12	RO	0x0	reserved
11	RW	0x0	hclk_sdio_gate_en SDIO AHB clock disable. When HIGH, disable clock
10	RW	0x0	hclk_sdmmc0_gate_en SDMMC0 AHB clock disable When HIGH, disable clock
9	RW	0x0	hclk_nandc_gate_en NANDC AHB clock disable When HIGH, disable clock
8	RO	0x0	reserved
7	RW	0x0	pclk_ddrupctl_gate_en DDR uPCTL APB clock disable. When HIGH, disable clock
6	RW	0x0	hclk_rom_gate_en ROM AHB clock disable. When HIGH, disable clock
5	RO	0x0	reserved
4	RW	0x0	pclk_grf_gate_en GRF APB clock disable. When HIGH, disable clock
3	RO	0x0	reserved
2	RW	0x0	pclk_efuse_gate_en EFUSE APB clock disable. When HIGH, disable clock
1	RW	0x0	aclk_dmac_gate_en DMAC AXI clock disable. When HIGH, disable clock
0	RW	0x0	pclk_mipiphy_gate_en mipiphy apb clock disable. When HIGH, disable clock

CRU_CLKGATE6_CON

Address: Operational Base + offset (0x00e8)

Internal clock gating control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	aclk_vio0_gate_en VIO0 AXI clock disable. When HIGH, disable clock
12	RW	0x0	hclk_vio_bus_gate_en VIO AHB bus clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	aclk_rga_gate_en RGA AXI clock disable. When HIGH, disable clock
10	RW	0x0	hclk_rga_gate_en RGA AHB clock disable. When HIGH, disable clock
9:6	RO	0x0	reserved
5	RW	0x0	aclk_cif_gate_en CIF AXI clock disable. When HIGH, disable clock
4	RW	0x0	hclk_cif_gate_en CIF AHB clock disable. When HIGH, disable clock
3:2	RO	0x0	reserved
1	RW	0x0	hclk_lcdc0_gate_en LCDCO AHB clock disable. When HIGH, disable clock
0	RW	0x0	aclk_lcdc0_gate_en LCDCO AXI clock disable. When HIGH, disable clock

CRU_CLKGATE7_CON

Address: Operational Base + offset (0x00ec)

Internal clock gating control register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_wdt_gate_en WDT APB clock disable. When HIGH, disable clock
14	RW	0x0	pclk_saradc_gate_en SARADC APB clock disable. When HIGH, disable clock
13	RO	0x0	reserved
12	RW	0x0	pclk_spi0_gate_en SPI0 APB clock disable. When HIGH, disable clock
11	RO	0x0	reserved
10	RW	0x0	pclk_pwm01_gate_en PWM0 and PWM1 APB clock disable. When HIGH, disable clock
9:8	RO	0x0	reserved
7	RW	0x0	pclk_timer_gate_en TIMER APB clock disable. When HIGH, disable clock
6:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	hclk_i2s_8ch_gate_en I2S_8ch AHB clock disable. When HIGH, disable clock
3	RW	0x0	hclk_host_gate_en USB HOST PHY AHB clock disable. When HIGH, disable clock
2	RW	0x0	hclk_i2s_2ch_gate_en I2S_2ch AHB clock disable. When HIGH, disable clock
1	RW	0x0	hclk_sfc_gate_en SFC AHB clock disable. When HIGH, disable clock
0	RW	0x0	hclk_emmc_gate_en EMMC AHB clock disable. When HIGH, disable clock

CRU_CLKGATE8_CON

Address: Operational Base + offset (0x00f0)

Internal clock gating control register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	pclk_gpio3_gate_en GPIO3 APB clock disable. When HIGH, disable clock
11	RW	0x0	pclk_gpio2_gate_en GPIO2 APB clock disable. When HIGH, disable clock
10	RW	0x0	pclk_gpio1_gate_en GPIO1 APB clock disable. When HIGH, disable clock
9	RW	0x0	pclk_gpio0_gate_en GPIO0 APB clock disable. When HIGH, disable clock
8	RO	0x0	reserved
7	RW	0x0	pclk_i2c3_gate_en I2C3 APB clock disable. When HIGH, disable clock
6	RW	0x0	pclk_i2c2_gate_en I2C2 APB clock disable. When HIGH, disable clock
5	RW	0x0	pclk_i2c1_gate_en I2C1 APB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
4	RW	0x0	pclk_i2c0_gate_en I2C0 APB clock disable. When HIGH, disable clock
3	RO	0x0	reserved
2	RW	0x0	pclk_uart2_gate_en UART2 APB clock disable. When HIGH, disable clock
1	RW	0x0	pclk_uart1_gate_en UART1 APB clock disable. When HIGH, disable clock
0	RW	0x0	pclk_uart0_gate_en UART0 APB clock disable. When HIGH, disable clock

CRU_CLKGATE9_CON

Address: Operational Base + offset (0x00f4)

Internal clock gating control register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_peri_niu_clock_en periph NIU AXI clock disable. When HIGH, disable clock
14	RW	0x0	hclk_peri_arbi_clock_en periph arbitor AHB clock disable. When HIGH, disable clock
13	RW	0x0	hclk_usb_peri_clock_en USB peri AHB clock disable. When HIGH, disable clock
12	RW	0x0	pclk_sim_clock_en SIM card APB clock disable. When HIGH, disable clock
11	RO	0x0	reserved
10	RW	0x0	aclk_vio1_clock_en VIO1 AXI clock disable. When HIGH, disable clock
9	RW	0x0	hclk_ebc_clock_en EBC AHB clock disable. When HIGH, disable clock
8	RW	0x0	aclk_iep_clock_en IEP AXI clock disable. When HIGH, disable clock
7	RW	0x0	hclk_iep_clock_en IEP AHB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	pclk_vio_mipi_gate_en pd_vio mipi controller clock disable. When HIGH, disable clock
5	RW	0x0	hclk_vio_h2p_gate_en pd_vio AHB h2p bridge clock disable. When HIGH, disable clock
4	RW	0x0	clk_I2C_gate_en I2C clock disable. When HIGH, disable clock
3	RW	0x0	pclk_pmu_noc_clock_en PD_PMU NOC APB clock disable. When HIGH, disable clock
2	RW	0x0	pclk_pmu_clock_en PMU APB clock disable. When HIGH, disable clock
1	RW	0x0	pclk_dbg_clock_en Debug APB clock disable. When HIGH, disable clock
0	RW	0x0	clk_core_dbg_gate_en Debug core clock disable. When HIGH, disable clock

CRU_CLKGATE10_CON

Address: Operational Base + offset (0x00f8)

Internal clock gating control register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_nandc_gate_en Nandc clock disable. When HIGH, disable clock
14	RW	0x0	clk_old_usb_host_gate_en old usb host clock disable. When HIGH, disable clock
13	RW	0x0	clkin0_tsp_gate_en TSP IO clkin0 clock disable. When HIGH, disable clock
12	RW	0x0	hclk_tsp_gate_en TSP AHB clock disable. When HIGH, disable clock
11	RW	0x0	pclk_gmac_gate_en GMAC APB clock disable. When HIGH, disable clock
10	RW	0x0	aclk_gmac_gate_en GMAC AXI clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	hclk_spdif_8ch_gate_en spdif_8ch AHB clock disable. When HIGH, disable clock
8	RW	0x0	clk_timer5_gate_en Timer5 clock disable. When HIGH, disable clock
7	RW	0x0	clk_timer4_gate_en Timer4 clock disable. When HIGH, disable clock
6	RW	0x0	clk_timer3_gate_en Timer3 clock disable. When HIGH, disable clock
5	RW	0x0	clk_timer2_gate_en Timer2 clock disable. When HIGH, disable clock
4	RW	0x0	clk_timer1_gate_en Timer1 clock disable. When HIGH, disable clock
3	RW	0x0	clk_timer0_gate_en Timer0 clock disable. When HIGH, disable clock
2	RW	0x0	func_pvtm_gate_en func PVTM clock disable. When HIGH, disable clock
1	RW	0x0	gpu_pvtm_gate_en pd_gpu PVTM clock disable. When HIGH, disable clock
0	RW	0x0	core_pvtm_gate_en pd_core PVTM clock disable. When HIGH, disable clock

CRU_GLB_SRST_FST_VALUE

Address: Operational Base + offset (0x0100)

The first global software reset config value

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	glb_srst_fst_value The first global software reset config value If config 0xfdb9, it will generate first global software reset.

CRU_GLB_SRST SND_VALUE

Address: Operational Base + offset (0x0104)

The second global software reset config value

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	glb_srst_snd_value The second global software reset config value If config 0xecfa8, it will generate second global software reset.

CRU_SOFRST0_CON

Address: Operational Base + offset (0x0110)

Internal software reset control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	I2c_srstn_req L2C software reset request. When HIGH, reset relative logic
14	RW	0x0	strc_sys_asrstn_req Structre system AXI software reset request. When HIGH, reset relative logic
13	RWSC	0x0	aclk_core_srstn_req core AXI clock software reset request. When HIGH, reset relative logic
12	RW	0x0	core_top_dbg_srstn_req CPU top debug software reset request. When HIGH, reset relative logic
11	RW	0x0	core3_dbg_srstn_req core3 CPU debug software reset request. When HIGH, reset relative logic
10	RW	0x0	core2_dbg_srstn_req core2 CPU debug software reset request. When HIGH, reset relative logic
9	RW	0x0	core1_dbg_srstn_req core1 CPU debug software reset request. When HIGH, reset relative logic
8	RW	0x0	core0_dbg_srstn_req core0 CPU debug software reset request. When HIGH, reset relative logic
7	RWSC	0x0	core3_srstn_req core3 CPU software reset request. When HIGH, reset relative logic
6	RWSC	0x0	core2_srstn_req core2 CPU software reset request. When HIGH, reset relative logic
5	RWSC	0x0	core1_srstn_req core1 CPU software reset request. When HIGH, reset relative logic
4	RWSC	0x0	core0_srstn_req core0 CPU software reset request. When HIGH, reset relative logic
3	RWSC	0x0	core3_posrstn_req core3 CPU PO software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	RWSC	0x0	core2_posrstn_req core2 CPU PO software reset request. When HIGH, reset relative logic
1	RWSC	0x0	core1_posrstn_req core1 CPU PO software reset request. When HIGH, reset relative logic
0	RWSC	0x0	core0_posrstn_req core0 CPU PO software reset request. When HIGH, reset relative logic

CRU_SOFTRST1_CON

Address: Operational Base + offset (0x0114)

Internal software reset control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	acodec_psrstn_req audio codec software reset request. When HIGH, reset relative logic
14	RW	0x0	efuse_psrstn_req EFUSE APB software reset request. When HIGH, reset relative logic
13	RW	0x0	core_pvtm_srstn_req CORE PVTM software reset request. When HIGH, reset relative logic
12	RO	0x0	reserved
11	RW	0x0	func_pvtm_srstn_req func PVTM software reset request. When HIGH, reset relative logic
10	RW	0x0	gpu_pvtm_srstn_req GPU PVTM software reset request. When HIGH, reset relative logic
9	RW	0x0	i2s_8ch_srstn_req I2S 8channel software reset request. When HIGH, reset relative logic
8	RW	0x0	i2s_2ch_srstn_req I2S 2channel software reset request. When HIGH, reset relative logic
7	RW	0x0	peri_niu_srstn_req periph_niu software reset request. When HIGH, reset relative logic
6	RW	0x0	rom_srstn_req ROM software reset request. When HIGH, reset relative logic
5	RW	0x0	intmem_srstn_req Internal memory software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
4	RW	0x0	spdif_srstn_req SPDIF software reset request. When HIGH, reset relative logic
3	RW	0x0	ahb2apb_hrstn_req AHB2APB software reset request. When HIGH, reset relative logic
2	RW	0x0	cpusys_hrstn_req CPU AHB software reset request. When HIGH, reset relative logic
1:0	RO	0x0	reserved

CRU_SOFTRST2_CON

Address: Operational Base + offset (0x0118)

Internal software reset control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	sfc_srstn_req SFC software reset request. When HIGH, reset relative logic
14	RW	0x0	i2c3_srstn_req I2C3 software reset request. When HIGH, reset relative logic
13	RW	0x0	i2c2_srstn_req I2C2 software reset request. When HIGH, reset relative logic
12	RW	0x0	i2c1_srstn_req I2C1 software reset request. When HIGH, reset relative logic
11	RW	0x0	i2c0_srstn_req I2C0 software reset request. When HIGH, reset relative logic
10	RO	0x0	reserved
9	RW	0x0	uart2_srstn_req UART2 software reset request. When HIGH, reset relative logic
8	RW	0x0	uart1_srstn_req UART1 software reset request. When HIGH, reset relative logic
7	RW	0x0	uart0_srstn_req UART0 software reset request. When HIGH, reset relative logic
6:5	RO	0x0	reserved
4	RW	0x0	mipi phy_psrstn_req mipi phy apb bus software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio3_srstn_req GPIO3 software reset request. When HIGH, reset relative logic
2	RW	0x0	gpio2_srstn_req GPIO2 software reset request. When HIGH, reset relative logic
1	RW	0x0	gpio1_srstn_req GPIO1 software reset request. When HIGH, reset relative logic
0	RW	0x0	gpio0_srstn_req GPIO0 software reset request. When HIGH, reset relative logic

CRU_SOFTRST3_CON

Address: Operational Base + offset (0x011c)

Internal software reset control register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	usb_peri_srstn_req USB PERIPH software reset request. When HIGH, reset relative logic
14	RW	0x0	emem_peri_srstn_req EMEM PERIPH software reset request. When HIGH, reset relative logic
13	RW	0x0	cpu_peri_srstn_req CPU PERIPH software reset request. When HIGH, reset relative logic
12	RW	0x0	smart_card_srstn_req smart_card software reset request. When HIGH, reset relative logic
11	RW	0x0	periphsys_psrstn_req PERIPH APB software reset request. When HIGH, reset relative logic
10	RW	0x0	periphsys_hsrstn_req PERIPH AHB software reset request. When HIGH, reset relative logic
9	RW	0x0	periphsys_asrstn_req PERIPH AXI software reset request. When HIGH, reset relative logic
8	RW	0x0	gmac_srstn_req GMAC software reset request. When HIGH, reset relative logic
7	RW	0x0	grf_srstn_req GRF software reset request. When HIGH, reset relative logic
6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	crypto_srstn_req crypto software reset request. When HIGH, reset relative logic
4	RW	0x0	dap_sys_srstn_req DAP system software reset request. When HIGH, reset relative logic
3	RW	0x0	dap_srstn_req DAP software reset request. When HIGH, reset relative logic
2	RW	0x0	dap_po_srstn_req DAP power software reset request. When HIGH, reset relative logic
1	RO	0x0	reserved
0	RW	0x0	pwm0_srstn_req PWM0 software reset request. When HIGH, reset relative logic

CRU_SOFTRST4_CON

Address: Operational Base + offset (0x0120)

Internal software reset control register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	ddrmsch_srstn_req DDR memory scheduler software reset request. When HIGH, reset relative logic
14:11	RO	0x0	reserved
10	RW	0x0	otgc1_srstn_req OTG controller1 software reset request. When HIGH, reset relative logic Host Controller utmi_clk domain reset
9	RO	0x0	reserved
8	RW	0x0	usb0tg1_srstn_req USBOTG1 software reset request. When HIGH, reset relative logic Host Controller hclk domain reset.
7	RW	0x0	otgc0_srstn_req OTG controller0 software reset request. When HIGH, reset relative logic. OTG Controller utmi_clk domain reset.
6	RO	0x0	reserved
5	RW	0x0	usb0tg0_srstn_req USBOTG0 software reset request. When HIGH, reset relative logic OTG Controller hclk domain reset.

Bit	Attr	Reset Value	Description
4	RW	0x0	nandc_srstn_req NANDC software reset request. When HIGH, reset relative logic
3	RW	0x0	gps_srstn_req GPS software reset request. When HIGH, reset relative logic
2:1	RO	0x0	reserved
0	RW	0x0	dma2_srstn_req DMA2 software reset request. When HIGH, reset relative logic

CRU_SOFTRST5_CON

Address: Operational Base + offset (0x0124)

Internal software reset control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	usbhost0_ehci_srstn_req usbhost0_ehci clock input0 domain software reset request. When HIGH, reset relative logic
13	RW	0x0	tsp_clkin0_srstn_req TSP clock input0 domain software reset request. When HIGH, reset relative logic
12	RW	0x0	tsp_srstn_req TSP software reset request. When HIGH, reset relative logic
11	RW	0x0	ddrctrl_psrstn_req DDR controller APB software reset request. When HIGH, reset relative logic
10	RW	0x0	ddrctrl_srstn_req DDR controller software reset request. When HIGH, reset relative logic
9	RW	0x0	ddrphy_psrstn_req DDR PHY APB software reset request. When HIGH, reset relative logic
8	RW	0x0	ddrphy_srstn_req DDR PHY software reset request. When HIGH, reset relative logic
7	RW	0x0	saradc_srstn_req SARADC software reset request. When HIGH, reset relative logic
6	RW	0x0	wdt_srstn_req WDT software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
5	RO	0x0	reserved
4	RW	0x0	spi0_srstn_req SPI0 software reset request. When HIGH, reset relative logic
3	RW	0x0	emmc_srstn_req EMMC software reset request. When HIGH, reset relative logic
2	RW	0x0	sdio_srstn_req SDIO software reset request. When HIGH, reset relative logic
1	RW	0x0	mmc0_srstn_req SDMMC0 software reset request. When HIGH, reset relative logic
0	RO	0x0	reserved

CRU_SOFTRST6_CON

Address: Operational Base + offset (0x0128)

Internal software reset control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pmu_srstn_req PMU software reset request. When HIGH, reset relative logic
14	RW	0x0	cif0_srstn_req CIF0 software reset request. When HIGH, reset relative logic
13	RW	0x0	rga_hsrstn_req RGA AHB software reset request. When HIGH, reset relative logic
12	RW	0x0	rga_asrstn_req RGA AXI software reset request. When HIGH, reset relative logic
11	RW	0x0	iep_hsrstn_req IEP AHB software reset request. When HIGH, reset relative logic
10	RW	0x0	iep_asrstn_req IEP AXI software reset request. When HIGH, reset relative logic
9	RW	0x0	usbpor_srstn_req USBPHY POR software reset request. When HIGH, reset relative logic. USB phy analog domain reset, including both OTG and HOST phy .

Bit	Attr	Reset Value	Description
8	RW	0x0	utmi1_srst_req UTMI1 software reset request. When HIGH, reset relative logic HOST phy digital domain reset. It should last at least 10 utmi_clk_1 cycles.
7	RW	0x0	utmi0_srstn_req UTMI0 software reset request. When HIGH, reset relative logic OTG phy digital domain reset. It should last at least 10 utmi_clk_0 cycles.
6	RW	0x0	lcdc0_dsrstn_req LCDCO DCLK software reset request. When HIGH, reset relative logic
5	RW	0x0	lcdc0_hsrstn_req LCDCO AHB software reset request. When HIGH, reset relative logic
4	RW	0x0	lcdc0_asrstn_req LCDCO AXI software reset request. When HIGH, reset relative logic
3	RW	0x0	vio_bus_hsrstn_req VIO bus AHB software reset request. When HIGH, reset relative logic
2	RW	0x0	vio0_asrstn_req VIO 0 NIU AXI software reset request. When HIGH, reset relative logic
1	RW	0x0	vio_arbi_hsrstn_req VIO arbitor AHB software reset request. When HIGH, reset relative logic
0	RW	0x0	hdmi_psrstn_req HDMI PCLK software reset request. When HIGH, reset relative logic

CRU_SOFTRST7_CON

Address: Operational Base + offset (0x012c)

Internal software reset control register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	ebc_hsrstn_req EBC AHB software reset request. When HIGH, reset relative logic
11	RW	0x0	ebc_asrstn_req EBC AXI software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
10	RW	0x0	gpu_niu_asrstn_req GPU NIU AXI software reset request. When HIGH, reset relative logic
9	RO	0x0	reserved
8	RW	0x0	gpu_srstn_req GPU core software reset request. When HIGH, reset relative logic
7	RW	0x0	lcdc0_ssrstn_req LCDCO SCLK software reset request. When HIGH, reset relative logic
6	RO	0x0	reserved
5	RW	0x0	pmu_niu_psrstn_req PD_PMU NIU APB software reset request. When HIGH, reset relative logic
4	RW	0x0	vcodec_niu_asrstn_req VCODEC NIU AXI software reset request. When HIGH, reset relative logic
3	RW	0x0	hevc_core_srstn_req HEVC CORE software reset request. When HIGH, reset relative logic
2	RW	0x0	vio1_asrstn_req VIO second AXI software reset request. When HIGH, reset relative logic
1	RW	0x0	vcodec_hsrstn_req VCODEC AHB software reset request. When HIGH, reset relative logic
0	RW	0x0	vcodec_asrstn_req VCODEC AXI software reset request. When HIGH, reset relative logic

CRU_SOFRST8_CON

Address: Operational Base + offset (0x0130)

Internal software reset control register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	vio_mipi_dsi_srstn_req pd_vio mipi_dsi software reset request. When HIGH, reset relative logic
8	RW	0x0	vio_h2p_srstn_req pd_vio h2p bridge software reset request. When HIGH, reset relative logic
7	RW	0x0	timer5_srstn_req Timer5 software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
6	RW	0x0	timer4_srstn_req Timer4 software reset request. When HIGH, reset relative logic
5	RW	0x0	timer3_srstn_req Timer3 software reset request. When HIGH, reset relative logic
4	RW	0x0	timer2_srstn_req Timer2 software reset request. When HIGH, reset relative logic
3	RW	0x0	timer1_srstn_req Timer1 software reset request. When HIGH, reset relative logic
2	RW	0x0	timer0_srstn_req Timer0 software reset request. When HIGH, reset relative logic
1	RW	0x0	dbg_psrstn_req DEBUG APB software reset request. When HIGH, reset relative logic
0	RW	0x0	core_dbg_srstn_req CORE DEBUG software reset request. When HIGH, reset relative logic

CRU_MISC_CON

Address: Operational Base + offset (0x0134)

SCU control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x1	usb480_src_sel USB480 and 24M selection 1'b0: select 24M 1'b1: select 480M
14:11	RO	0x0	reserved
10:8	RW	0x0	testclk_sel Output clock selection for test 3'b000: clk_pvtm 3'b001: sclk_lcdc 3'b010: clk_core 3'b011: clk_ddrphy 3'b100: aclk_vio0 3'b101: aclk_gpu 3'b110: aclk_peri 3'b111: aclk_bus
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	core0_porst_wdt_sel Select reset watchdog when A9 core 0 power on reset 1'b0: not reset watchdog 1'b1: reset watchdog

CRU_GLB_CNT_TH

Address: Operational Base + offset (0x0140)
global reset wait counter threshold

Bit	Attr	Reset Value	Description
31:16	RW	0x3a98	pll_lock_period PLL lock period
15	RW	0x0	wdt_glb_srst_ctrl watch dog trigger global soft reset select 1'b0: watch_dog trigger second global reset 1'b1: watch_dog trigger first global reset
14	RO	0x0	reserved
13:12	RW	0x0	pmu_glb_srst_ctrl watch dog trigger global soft reset select 2'b00: pmu reset by first global soft reset 2'b01: pmu reset by second global soft reset 2'b10: pmu not reset by any global soft reset
11:10	RO	0x0	reserved
9:0	RW	0x064	glb_RST_CNT_TH Global soft reset counter threshold

CRU_GLB_RST_ST

Address: Operational Base + offset (0x0150)
global reset status

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	W1C	0x0	snd_glb_wdt_rst_st second global watch_dog rst flag 1'b0: last hot reset is not second global watch_dog triggered reset 1'b1: last hot reset is second global watch_dog triggered reset
2	W1C	0x0	fst_glb_wdt_rst_st first global watch_dog rst flag 1'b0: last hot reset is not first global watch_dog triggered reset 1'b1: last hot reset is first global watch_dog triggered reset
1	W1C	0x0	snd_glb_rst_st second global rst flag 1'b0: last hot reset is not second global rst 1'b1: last hot reset is second global rst

Bit	Attr	Reset Value	Description
0	W1C	0x0	fst_glb_RST_ST first global rst flag 1'b0: last hot reset is not first global rst 1'b1: last hot reset is first global rst

CRU_SDMMC_CON0

Address: Operational Base + offset (0x01c0)

sdmmc control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	WO	0x0	sdmmc_drv_sel sdmmc drive select sdmmc drive select
10:3	WO	0x00	sdmmc_drv_delaynum sdmmc drive delay number sdmmc drive delay number
2:1	WO	0x2	sdmmc_drv_degree sdmmc drive degree sdmmc drive degree
0	WO	0x0	sdmmc_init_state sdmmc initial state sdmmc initial state

CRU_SDMMC_CON1

Address: Operational Base + offset (0x01c4)

sdmmc control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	WO	0x0	sdmmc_sample_sel sdmmc sample select sdmmc sample select
9:2	WO	0x00	sdmmc_sample_delaynum sdmmc sample delay number sdmmc sample delay number
1:0	WO	0x0	sdmmc_sample_degree sdmmc sample degree sdmmc sample degree

CRU_SDIO_CON0

Address: Operational Base + offset (0x01c8)
sdio0 control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	WO	0x0	sdio0_drv_sel sdio0 drive select sdio0 drive select
10:3	WO	0x00	sdio0_drv_delaynum sdio0 drive delay number sdio0 drive delay number
2:1	WO	0x2	sdio0_drv_degree sdio0 drive degree sdio0 drive degree
0	WO	0x0	sdio0_init_state sdio0 initial state sdio0 initial state

CRU_SDIO_CON1

Address: Operational Base + offset (0x01cc)
sdio0 control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	WO	0x0	sdio0_sample_sel sdio0 sample select sdio0 sample select
9:2	WO	0x00	sdio0_sample_delaynum sdio0 sample delay number sdio0 sample delay number
1:0	WO	0x0	sdio0_sample_degree sdio0 sample degree sdio0 sample degree

CRU_EMMC_CON0

Address: Operational Base + offset (0x01d8)
emmc control0

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	WO	0x0	emmc_drv_sel emmc drive select emmc drive select
10:3	WO	0x00	emmc_drv_delaynum emmc drive delay number emmc drive delay number
2:1	WO	0x2	emmc_drv_degree emmc drive degree emmc drive degree
0	WO	0x0	emmc_init_state emmc initial state emmc initial state

CRU_EMMC_CON1

Address: Operational Base + offset (0x01dc)

emmc control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	WO	0x0	emmc_sample_sel emmc sample select emmc sample select
9:2	WO	0x00	emmc_sample_delaynum emmc sample delay number emmc sample delay number
1:0	WO	0x0	emmc_sample_degree emmc sample degree emmc sample degree

CRU_PLL_PRG_EN

Address: Operational Base + offset (0x01f0)

PLL program enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:0	WO	0x0000	pll_prg_en pll program enable when pll_prg_en is 16'h5a5a, all the pll_con can be programmed

3.8 Timing Diagram

Power on reset timing is shown as follow:

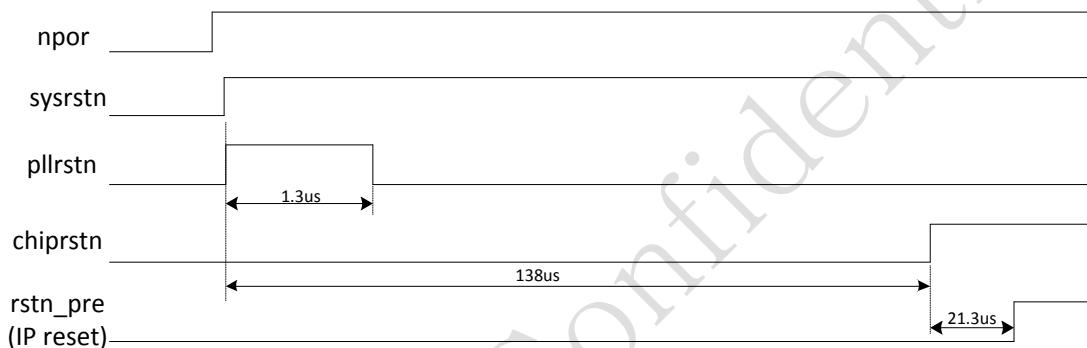


Fig. 3-7 Chip Power On Reset Timing Diagram

NPOR is hardware reset signal from out-chip, which is filtered glitch to obtain signal sysrstn. To make PLLs work normally, the PLL reset signal (pllrstn) must maintain high for more than 1us, and PLLs start to lock when pllrstn deassert, and the PLL max lock time is 1500 PLL REFCLK cycles. And then the system will wait about 138us, and then deactivate reset signal chiprstn. The signal chiprstn is used to generate output clocks in CRU. After CRU start output clocks, the system waits again for 512cycles (21.3us) to deactivate signal rstn_pre, which is used to generate power on reset of all IP.

3.9 Application Notes

3.9.1 PLL usage

The chip uses 2.4GHz for all four PLLs (ARM PLL, DDR PLL, CODEC PLL and GENERAL PLL).

A. PLL output frequency configuration

FBDIV, POSTDIV1, BYPASS can be configured by programming CRU_APLL_CON0, CRU_DPLL_CON0, CRU_CPLL_CON0 and CRU_GPLL_CON0.

DSMPD, REFDIV, POSTDIV2 can be configured by programming CRU_APLL_CON1, CRU_DPLL_CON1, CRU_CPLL_CON1 and CRU_GPLL_CON1.

FRAC can be configured by programming CRU_APLL_CON2, CRU_DPLL_CON2, CRU_CPLL_CON2 and CRU_GPLL_CON2.

- (1) If DSMPD = 1 (DSM is disabled, "integer mode")

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * \text{FBDIV}$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}$$

When FREF is 24MHz, and if 700MHz FOUTPOSTDIV is needed. The configuration can be:

DSMPD = 1
 REFDIV = 6
 FBDIV = 175
 POSTDIV1=1
 POSTDIV2=1

And then

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * \text{FBDIV} = 24/6*175=700$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}=700/1/1=700$$

- (2) If DSMPD = 0 (DSM is enabled, "fractional mode")
 $\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * (\text{FBDIV} + \text{FRAC} / 2^{24})$
 $\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}$

When FREF is 24MHz, and if 491.52MHz FOUTPOSTDIV is needed. The configuration can be:

DSMPD = 0
 REFDIV = 1
 FBDIV = 40
 FRAC = 24'hf5c28f
 POSTDIV1=2
 POSTDIV2=1

And then

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * (\text{FBDIV} + \text{FRAC} / 2^{24}) = 24/1*(40+24'hf5c28f/2^{24})= 983.04$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}=983.04/2/1=491.52$$

B. PLL frequency range requirement

All the value range requirements are as follow.

FREF(Input Frequency Range in Integer Mode):1MHz to 800MHz

FREF(Input Frequency Range in Fractional Mode):10MHz to 800MHz

FREF/REFDIV(The divided reference frequency): 1 to 50MHz

FOUTVCO: 600MHz to 2.4GHz

C. PLL setting consideration

- If the POSTDIV value is changed during operation a short pulse (glitch) may occur on FOUTPOSTDIV. The minimum width of the short pulse will be equal to twice the period of the VCO. Therefore, if the circuitry clocked by the PLL is sensitive to short pulses, the new divide value should be re-timed so that it is synchronous with the rising edge of the output clock (FOUTPOSTDIV). Glitches cannot occur on any of the other outputs.
- For lowest power operation, the minimum VCO and FREF frequencies should be used. For minimum jitter operation, the highest VCO and FREF frequencies should be used. The normal operating range for the VCO is described above in .
- The supply rejection will be worse at the low end of the VCO range so care should be taken to keep the supply clean for low power applications.
- The feedback divider is not capable of dividing by all possible settings due to the use of a power-saving architecture. The following settings are valid for FBDIV:

- DSMPD=1 (Integer Mode):
12,13,14,16-4095 (practical value is limited to 3200, 2400, or 1600 (FVCOMAX / FREFMIN))
- DSMPD=0 (Fractional Mode):
19-4091 (practical value is limited to 320, 240, or 160 (FVCOMAX / FREFMIN))
- The PD input places the PLL into the lowest power mode. In this case, all analog circuits are turned off and FREF will be "ignored". The FOUTPOSTDIV and FOUTVCO pins are forced to logic low (0V).
- The BYPASS pin controls a mux which selects FREF to be passed to the FOUTPOSTDIV when active high. However, the PLL continues to run as it normally would if bypass were low. This is a useful feature for PLL testing since the clock path can be verified without the PLL being required to work. Also, the effect that the PLL induced supply noise has on the output buffering can be evaluated. It is not recommended to switch between BYPASS mode and normal mode for regular chip operation since this may result in a glitch. Also, FOUTPOSTDIVPD should be set low if the PLL is to be used in BYPASS mode.

3.9.2 PLL frequency change and lock check

The PLL programming support changed on-the-fly and the PLL will simply slew to the new frequency.

PLL lock state can be check in CRU_APLL_CON1[10], CRU_DPLL_CON1[10], CRU_CPLL_CON1[10], CRU_GPLL_CON1[10] register. The lock state is high when both original hardware PLL lock and PLL counter lock are high. The PLL counter lock initial value is CRU_GLB_CNT_TH[31:16].

The max delay time is 1500 REF_CLK.

PLL locking consists of three phases.

- Phase 1 is control voltage slewing. During this phase one of the clocks (reference or divide) is much faster than the other, and the PLL frequency adjusts almost continuously. When locking from power down, the divide clock is initially very slow and steadily increases frequency. Slew time is about 2-5 s. It will take slightly longer for faster VCO settings when locking from power down, since the PLL must slew further.
- Phase 2 is small signal phase acquisition. During this phase, the internal up/down signals alternate semi-chaotically as the phase slowly adjusts until the two signals are aligned. The duration of this phase depends on the loop bandwidth and is faster with higher bandwidth. Bandwidth can be estimated as FREF / REFDIV / 20 for integer mode and FREF / REFDIV / 40 for fractional mode. The duration of small signal locking is about 1/Bandwidth.
- Phase 3 is the digital cycle count. After the last cycle slip is detected, an internal counter waits 256 FREF / REFDIV cycles before the lock signal goes high. This is frequently the dominant factor in lock time – especially for slower reference clock signals or large reference divide settings. This time can be calculated as 256*REFDIV/FREF.

3.9.3 Fractional divider usage

To get specific frequency, clocks of I2S, SPDIF, UART, HSADC can be generated by fractional divider. Generally you must set that denominator is 20 times larger than numerator to generate precise clock frequency. So the fractional divider

applies only to generate low frequency clock like I2S, UART and HSADC.

3.9.4 Global software reset

Two global software resets are designed in the chip, you can program CRU_GLB_SRST_FST_VALUE[15:0] as 0xfd9 to assert the first global software reset glb_srstn_1 and program CRU_GLB_SRST_SND_VALUE[15:0] as 0xec8 to assert the second global software reset glb_srstn_2. These two software resets are self-deasserted by hardware.

Glb_srstn_1 resets almost all logic.

Glb_srstn_2 resets almost all logic except GRF and GPIOs.

After global reset, the reset trigger source can be check in CRU_GLB_RST_ST.

Chapter 4 System Debug

4.1 Overview

The chip uses the DAPLITE Technology to support real-time debug.

4.1.1 Features

- Invasive debug with core halted
- SW-DP

4.1.2 Debug components address map

The following table shows the debug components address in memory map:

Module	Base Address
DAP_ROM	0x20020000

4.2 Block Diagram

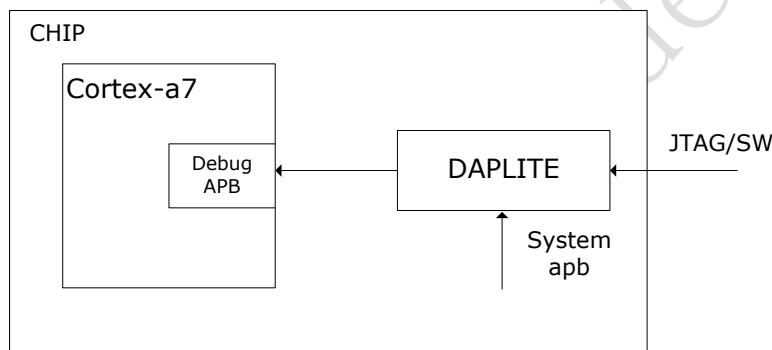


Fig.4-1 Debug system structure

4.3 Function description

4.3.1 DAP

The DAP has following components:

- Serial Wire JTAG Debug Port(SWJ-DP)
- APB Access Port(APB-AP)
- ROM table

The debug port is the host tools interface to access the DAP-Lite. This interface controls any access ports provided within the DAP-Lite. The DAP-Lite supports a combined debug port which includes both JTAG and Serial Wire Debug(SWD), with a mechanism that supports switching between them.

The APB-AP acts as a bridge between SWJ-DP and APB bus which translate the Debug request to APB bus.

The DAP provides an internal ROM table connected to the master Debug APB port of the APB-Mux. The Debug ROM table is loaded at address 0x00000000 and 0x80000000 of this bus and is accessible from both APB-AP and the system APB input. Bit [31] of the address bus is not connected to the ROM Table, ensuring that both views read the same value. The ROM table stores the locations of the components on the Debug APB.

4.4 Register description

4.4.1 DAP APB-AP register summary

Name	Offset	Size	Reset	Description
DAP_CSW	0x000	W	0x00000002	Control/Status Word, CSW
DAP_TAR	0x004	W	0x00000000	Transfer Address, TAR
Reserved	0x008	W	NA	Reserved
DAP_DRW	0x00c	W	NA	Data Read/Write, DRW
DAP_BD0	0x010	W	NA	Bank Data 0, BD0
DAP_BD1	0x014	W	NA	Bank Data 1, BD1
DAP_BD2	0x018	W	NA	Bank Data 2, BD2
DAP_BD3	0x01c	W	NA	Bank Data 3, BD3
Reserved	0x20-0xf4	W	NA	Reserved
DAP_ROM_ADDR	0xf8	W	NA	Debug ROM Address, ROM
DAP_IDR	0xfc	W	0x14770002	Identification Register, IDR

4.4.2 DAP APB-AP Detailed Register Description

DAP_CSW

Address: APBAP_BASE + offset(0x000)

Control/Status Word

Bits	Attr	Reset Value	Description
31	RW	0x0	Software access enable. Drives DBGSWENABLE to enable or disable software access to the Debug APB bus in the APB multiplexor. b1 = Enable software access b0 = Disable software access. Reset value = b0. On exit from reset, defaults to b1 to enable software access.
31:12	RW	0x0	Reserved
11:8	R	0x0	Specifies the mode of operation. b0000 = Normal download/upload model b0001-b1111 = Reserved Reset value = b0000.
7	R	0x0	Transfer in progress. This field indicates if a transfer is currently in progress on the APB master port.
6	R	0x0	Transfer in progress. This field indicates if a transfer is currently in progress on the APB master port. Indicates the status of the DEVICEEN input. • If APB-AP is connected to the Debug APB, that is, a bus connected only to debug and trace components, it must be permanently enabled by tying DEVICEEN HIGH. This ensures that trace components can still be programmed when DBGEN is LOW. In practice, it is expected that the APB-AP is almost always used in this way. • If APB-AP is connected to a system APB dedicated to the non-secure world, DEVICEEN must be

			connected to DBGEN. • If APB-AP is connected to a system APB dedicated to the secure world, DEVICEEN must be connected to SPIDEN.
5:4	RW	0x0	Auto address increment and packing mode on Read or Write data access. Does not increment if the transaction completes with an error response or the transaction is aborted. Auto address incrementing is not performed on access to banked data registers 0x10-0x1C. The status of these bits is ignored in these cases. b11 = Reserved b10 = Reserved b01 = Increment b00 = Auto increment OFF. Increment occurs in word steps. Reset value = b00.
3	R	0x0	Reserved
2:0	R	0x2	Size of the access to perform. Fixed at b010 = 32 bits. Reset value = b010.

DAP_TAR

Address: APBAP_BASE + offset(0x004)

Transfer Address

Bits	Attr	Reset Value	Description
31:2	RW	0x0	Address[31:2] of the current transfer PADDR[31:2]=TAR[31:2] for accesses from Data Read/Write Register at 0x0C. PADDR[31:2]=TAR[31:4]+DAPADDR[3:2] for accesses from Banked Data Registers at 0x10-0x1C and 0x0C.
1:0	R	0x0	Reserved

DAP_DRW

Address: APBAP_BASE + offset(0x00c)

Data Read/Write

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Write mode: Data value to write for the current transfer. Read mode: Data value read from the current transfer.

DAP_BD0-DAP_BD3

Address: APBAP_BASE + offset(0x010) - APBAP_BASE + offset(0x01c)

Bank Data 0-3

Bits	Attr	Reset Value	Description
31:0	RW	0x0	If DAPADDR[7:4] = 0x0001, so accessing APB-AP registers in the range 0x10-0x1C, then the derived PADDR[31:0] is: • Write mode: Data value to write for the current transfer to external address TAR[31:4]+

			DAPADDR[3:2] + 2'b00. • Read mode: Data value read from the current transfer from external address TAR[31:4]+ DAPADDR[3:2] + 2'b00. Auto address incrementing is not performed on DAP accesses to BD0-BD3. Reset value = 0x00000000
--	--	--	--

DAP_ROM_ADDR

Address: 0xf8

ROM address

Bits	Attr	Reset Value	Description
31:12	R	0x800000	Base address of the ROM Table. The ROM provides a look-up table of all CoreSight Debug APB components. Read only. Set to 0xFFFF if no ROM is present. In the initial CoreSight release this must be set to 0x80000.
11:0	R	0x000	Set to 0x000 if ROM is present. Set to 0xFFFF if ROM table is not present. In the initial CoreSight release this must be set to 0x000.

DAP_IDR

Address: APBAP_BASE + offset(0x0fc)

Bits	Attr	Reset Value	Description
31:28	R	0x1	Revision. Reset value is 0x1 for APB-AP.
27:24	R	0x4	JEDEC bank. 0x4 indicates ARM Limited.
23:17	R	0x3b	JEDEC code. 0x3B indicates ARM Limited.
16	R	0x1	Memory AP. 0x1 indicates a standard register map is used.
15:8	R	0x00	Reserved
7:0	R	0x02	Identity value. Reset value is 0x02 for APB-AP.

DAP_DEBUG_ROM

Address: 0xf8

ROM address

Bits	Attr	Reset Value	Description
31:0	RO	-	Base address of a ROM table. The ROM provides a look-up table for system components. Set to 0xFFFFFFFF in the AHB-AP in the initial release

DAP_AHB_IDR

Address: APBAP_BASE + offset(0x0fc)

Bits	Attr	Reset Value	Description
31:28	R	0x4	Revision. Reset value is 0x4 for AHB-AP.
27:24	R	0x4	JEDEC bank. 0x4 indicates ARM Limited.
23:17	R	0x3b	JEDEC code. 0x3B indicates ARM Limited.
16	R	0x1	Memory AP. 0x1 indicates a standard register map is used.
15:8	R	0x00	Reserved
7:0	R	0x01	Identity value. Reset value is 0x01 for AHB-AP.

4.4.3 DAP-ROM register summary

Name	Offset	Size	Reset Value	Description
DAP_ROMENTRY0	0x0000	W	0x00001003	CTI4 entry register
DAP_ROMENTRY1	0x0004	W	0x00002003	TPIU entry register
DAP_ROMENTRY2	0x0008	W	0x00003003	Trace Funnel register
DAP_ROMENTRY3	0x000c	W	0x00004003	Cortex-A9 ROM entry register
DAP_ROM_PERIPHID4	0x0fd0	W	0x00000004	Peripheral ID4
DAP_ROM_PERIPHID5	0x0fd4	W	0x00000000	Peripheral ID5
DAP_ROM_PERIPHID6	0x0fd8	W	0x00000000	Peripheral ID6
DAP_ROM_PERIPHID7	0x0fdc	W	0x00000000	Peripheral ID7
DAP_ROM_PERIPHID0	0x0fe0	W	0x000000c4	Peripheral ID0
DAP_ROM_PERIPHID1	0x0fe4	W	0x000000b4	Peripheral ID1
DAP_ROM_PERIPHID2	0x0fe8	W	0x0000006b	Peripheral ID2
DAP_ROM_PERIPHID3	0x0fec	W	0x00000020	Peripheral ID3
DAP_ROM_COMPONID0	0x0ff0	W	0x0000000d	Component ID0
DAP_ROM_COMPONID1	0x0ff4	W	0x00000010	Component ID1
DAP_ROM_COMPONID2	0x0ff8	W	0x00000005	Component ID2
DAP_ROM_COMPONID3	0x0ffc	W	0x000000b1	Component ID3

4.4.4 DAP-ROM Detailed Register Description

DAP_ROMENTRY0

Address: DAPROM_BASE + offset(0x0000)

TPIU entry register

Bits	Attr	Reset Value	Description
31:0	R	0x00001003	TPIU entry register

DAP_ROMENTRY1

Address: DAPROM_BASE + offset(0x0004)

Cortex-A9 Debug entry register

Bits	Attr	Reset Value	Description
31:0	R	0x00002003	Cortex-A9 Debug entry register

DAP_ROMENTRY2

Address: DAPROM_BASE + offset(0x0008)

Cortex-A9 ETM entry register

Bits	Attr	Reset Value	Description
31:0	R	0x00003003	Cortex-A9 ETM entry register

DAP_ROMENTRY3

Address: DAPROM_BASE + offset(0x000c)

Cortex-A9 CTI entry register

Bits	Attr	Reset Value	Description
31:0	R	0x00004003	Cortex-A9 CTI entry register

DAP_ROM_PERIPHID4

Address: DAPROM_BASE + offset(0x0fd0)

Peripheral ID4

Bits	Attr	Reset Value	Description
31:0	R	0x00000004	Peripheral ID4

DAP_ROM_PERIPHID5

Address: DAPROM_BASE + offset(0x0fd4)

Peripheral ID5

Bits	Attr	Reset Value	Description
31:0	R	0x00000000	Peripheral ID5

DAP_ROM_PERIPHID6

Address: DAPROM_BASE + offset(0x0fd8)

Peripheral ID6

Bits	Attr	Reset Value	Description
31:0	R	0x00000000	Peripheral ID6

DAP_ROM_PERIPHID7

Address: DAPROM_BASE + offset(0x0fdc)

Peripheral ID7

Bits	Attr	Reset Value	Description
31:0	R	0x00000000	Peripheral ID7

DAP_ROM_PERIPHIDO

Address: DAPROM_BASE + offset(0x0fe0)

Peripheral ID0

Bits	Attr	Reset Value	Description
31:0	R	0x000000c4	Peripheral ID0

DAP_ROM_PERIPHID1

Address: DAPROM_BASE + offset(0x0fe4)

Peripheral ID1

Bits	Attr	Reset Value	Description
31:0	R	0x000000b4	Peripheral ID1

DAP_ROM_PERIPHID2

Address: DAPROM_BASE + offset(0x0fe8)

Peripheral ID2

Bits	Attr	Reset Value	Description
31:0	R	0x00000006b	Peripheral ID2

DAP_ROM_PERIPHID3

Address: DAPROM_BASE + offset(0x0fec)

Peripheral ID3

Bits	Attr	Reset Value	Description
31:0	R	0x00000020	Peripheral ID3

DAP_ROM_COMPONENTID0

Address: DAPROM_BASE + offset(0x0ff0)

Component ID0

Bits	Attr	Reset Value	Description
31:0			

31:0	R	0x0000000d	Component ID0
------	---	------------	---------------

DAP_ROM_COMPONID1

Address: DAPROM_BASE + offset(0x0ff4)

Component ID0

Bits	Attr	Reset Value	Description
31:0	R	0x00000010	Component ID1

DAP_ROM_COMPONID2

Address: DAPROM_BASE + offset(0x0ff8)

Component ID0

Bits	Attr	Reset Value	Description
31:0	R	0x00000005	Component ID2

DAP_ROM_COMPONID3

Address: DAPROM_BASE + offset(0x0ffC)

Component ID0

Bits	Attr	Reset Value	Description
31:0	R	0x000000b1	Component ID3

Notes: Attr: **RW**- Read/writable, **RO**- read only, **WO**- write only, **RWTC**-Readable and write "1" to clear the asserted bit from "1" to "0".

4.5 Interface description

4.5.1 DAP SWJ-DP interface

The following figure is the DAP SWJ-DP interface, the SWJ-DP is a combined JTAG-DP and SW-DP that enable you connect either a Serial Wire Debug(SWJ) to JTAG probe to a target.

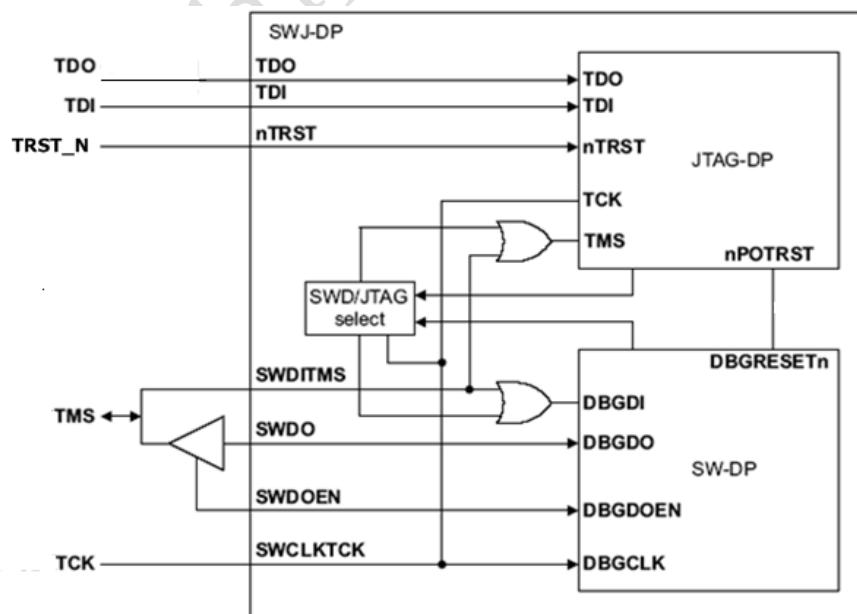


Fig.4-2 DAP SWJ interface

4.5.2 DAP SW-DP interface

This implementation is taken from ADIV5.1 and operates with a synchronous serial interface. This uses a single bidirectional data signal, and a clock signal.

The figure below describes the interaction between the timing of transactions on the serialwire interface, and the DAP internal bus transfers. It shows when the target responds with a WAIT acknowledgement.

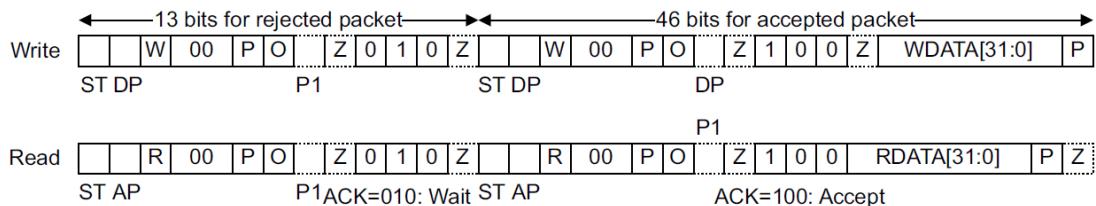


Fig.4-3SW-DP acknowledgement timing

Table 4-1 SW-DP Interface Description

Module pin	Direction	Pad name	IOMUX
I2C0 Interface			
jtag_tck	I	IO_MMC0d2_JTAGt ck1_GPIO1c4	GRF_GPIO1C_IOMUX[9:8] =2'b10 & mmc0_detn
jtag_tms	I/O	IO_MMC0d3_JTAGt ms1_GPIO1c5	GRF_GPIO1C_IOMUX[11:10]=2'b10 & mmc0_detn

Note :mmc0_detn, when high, no sd card is used.

Chapter 5 General register file(GRF)

5.1 Overview

The general register file will be used to do static set by software, which is composed of many registers for system control.

5.1.1 Features

The function of general register file is:

- IOMUX control
- Control the state of GPIO in power-down mode
- GPIO PAD pull down and pull up control
- Used for common system control
- Used to record the system state

5.2 GRF Register Description

5.2.1 Register Summary

Name	Offset	Size	Reset Value	Description
GRF_GPIO0A_IOMUX	0x00a8	W	0x00000000	GPIO0A iomux control
GRF_GPIO0B_IOMUX	0x00ac	W	0x00000000	GPIO0B iomux control
GRF_GPIO0C_IOMUX	0x00b0	W	0x00000000	GPIO0C iomux control
GRF_GPIO0D_IOMUX	0x00b4	W	0x00000000	GPIO0D iomux control
GRF_GPIO1A_IOMUX	0x00b8	W	0x00000c00	GPIO1A iomux control
GRF_GPIO1B_IOMUX	0x00bc	W	0x00000030	GPIO1B iomux control
GRF_GPIO1C_IOMUX	0x00c0	W	0x00000000	GPIO1C iomux control
GRF_GPIO1D_IOMUX	0x00c4	W	0x00000000	GPIO1D iomux control
GRF_GPIO2A_IOMUX	0x00c8	W	0x00000000	GPIO2A iomux control
GRF_GPIO2B_IOMUX	0x00cc	W	0x00000000	GPIO2B iomux control
GRF_GPIO2C_IOMUX	0x00d0	W	0x00000000	GPIO2C iomux control
GRF_GPIO2D_IOMUX	0x00d4	W	0x00000000	GPIO2D iomux control
GRF_GPIO3A_IOMUX	0x00d8	W	0x00000000	GPIO3A iomux control
GRF_GPIO3B_IOMUX	0x00dc	W	0x00000000	GPIO3B iomux control
GRF_GPIO3C_IOMUX	0x00e0	W	0x00000000	GPIO3D iomux control
GRF_GPIO3D_IOMUX	0x00e4	W	0x00000000	GPIO3D iomux control
GRF_GPIO2C_IOMUX_2	0x00e8	W	0x00000000	GPIO2C iomux control
GRF_CIF_IOMUX	0x00ec	W	0x00000000	CIF iomux control
GRF_CIF_IOMUX1	0x00f0	W	0x00000000	CIF iomux control register1
GRF_GPIO_DS	0x0100	W	0x00000000	GPIO DS control
GRF_GPIO0L_PULL	0x0118	W	0x00000000	GPIO0A / GPIO0B pull up/down control
GRF_GPIO0H_PULL	0x011c	W	0x00000000	GPIO0C / GPIO0D pull up/down control
GRF_GPIO1L_PULL	0x0120	W	0x00000000	GPIO0A / GPIO0B pull up/down control
GRF_GPIO1H_PULL	0x0124	W	0x00000000	GPIO1C / GPIO1D pull up/down control

Name	Offset	Size	Reset Value	Description
GRF_GPIO2L_PULL	0x0128	W	0x00000000	GPIO2A / GPIO2B pull up/down control
GRF_GPIO2H_PULL	0x012c	W	0x00000000	GPIO2C / GPIO2D pull up/down control
GRF_GPIO3L_PULL	0x0130	W	0x00000000	GPIO3A / GPIO3B pull up/down control
GRF_GPIO3H_PULL	0x0134	W	0x00000000	GPIO3C / GPIO3D pull up/down control
GRF_ACODEC_CON	0x013c	W	0x00000020	SoC control register
GRF_SOC_CON0	0x0140	W	0x00000120	SoC control register
GRF_SOC_CON1	0x0144	W	0x00000000	SoC control register
GRF_SOC_CON2	0x0148	W	0x00000084	SoC control register
GRF_SOC_STATUS0	0x014c	W	0x00000000	SoC status register
GRF_LVDS_CON0	0x0150	W	0x00000000	LVDS control register
GRF_DMAC_CON0	0x015c	W	0x00000000	DMAC control register
GRF_DMAC_CON1	0x0160	W	0x00000000	DMAC control register
GRF_DMAC_CON2	0x0164	W	0x00000010	DMAC control register
GRF_MAC_CON0	0x0168	W	0x00000810	GMAC control register0
GRF_MAC_CON1	0x016c	W	0x00004040	GMAC control register1
GRF_TVE_CON	0x0170	W	0x00000000	TV encoder control register
GRF_UOC0_CON0	0x017c	W	0x00000000	OTG control register
GRF_UOC1_CON1	0x0184	W	0x00000000	usb host control register
GRF_UOC1_CON2	0x0188	W	0x0000c820	UOC1 control register 2
GRF_UOC1_CON3	0x018c	W	0x00000b40	UOC1 control register 3
GRF_UOC1_CON4	0x0190	W	0x0000001c	USB HOST 2.0 control register
GRF_UOC1_CON5	0x0194	W	0x00000000	USB HOST 2.0 control register
GRF_DDRC_STAT	0x019c	W	0x00000000	DDRC status
GRF_SOC_STATUS1	0x01a4	W	0x00000000	SoC status register
GRF_CPU_CON0	0x01a8	W	0x00002002	CPU control register
GRF_CPU_CON1	0x01ac	W	0x00000000	CPU control register
GRF_CPU_CON2	0x01b0	W	0x0000003f	CPU control register
GRF_CPU_CON3	0x01b4	W	0x00002aaa	CPU control register
GRF_CPU_STATUS0	0x01c0	W	0x00000000	CPU status register
GRF_CPU_STATUS1	0x01c4	W	0x00000000	CPU status register
GRF_OS_REG0	0x01c8	W	0x00000000	software OS register
GRF_OS_REG1	0x01cc	W	0x00000000	software OS register
GRF_OS_REG2	0x01d0	W	0x00000000	software OS register
GRF_OS_REG3	0x01d4	W	0x00000000	software OS register
GRF_OS_REG4	0x01d8	W	0x00000000	software OS register
GRF_OS_REG5	0x01dc	W	0x00000000	software OS register
GRF_OS_REG6	0x01e0	W	0x00000000	software OS register
GRF_OS_REG7	0x01e4	W	0x00000000	software OS register
GRF_PVTM_CON0	0x0200	W	0x00000000	PVTM control register
GRF_PVTM_CON1	0x0204	W	0x00000000	PVTM control register
GRF_PVTM_CON2	0x0208	W	0x00000000	PVTM control register
GRF_PVTM_CON3	0x020c	W	0x00000000	PVTM control register
GRF_PVTM_STATUS0	0x0210	W	0x00000000	PVTM status register0

Name	Offset	Size	Reset Value	Description
GRF_PVTM_STATUS1	0x0214	W	0x00000000	PVTM status register1
GRF_PVTM_STATUS2	0x0218	W	0x00000000	PVTM status register2
GRF_PVTM_STATUS3	0x021c	W	0x00000000	PVTM status register3
GRF_DFI_WRNUM	0x0220	W	0x00000000	DFI write number register
GRF_DFI_RDNUM	0x0224	W	0x00000000	DFI read number register
GRF_DFI_ACTNUM	0x0228	W	0x00000000	DFI active number register
GRF_DFI_TIMERVAL	0x022c	W	0x00000000	DFI work time
GRF_NIF_FIFO0	0x0230	W	0x00000000	NIF status register
GRF_NIF_FIFO1	0x0234	W	0x00000000	NIF status register
GRF_NIF_FIFO2	0x0238	W	0x00000000	NIF status register
GRF_NIF_FIFO3	0x023c	W	0x00000000	NIF status register
GRF_USBPHY0_CON0	0x0280	W	0x00008618	usbphy control register
GRF_USBPHY0_CON1	0x0284	W	0x0000e007	usbphy control register
GRF_USBPHY0_CON2	0x0288	W	0x000082aa	usbphy control register
GRF_USBPHY0_CON3	0x028c	W	0x00000200	usbphy control register
GRF_USBPHY0_CON4	0x0290	W	0x00000002	usbphy control register
GRF_USBPHY0_CON5	0x0294	W	0x00000000	usbphy control register
GRF_USBPHY0_CON6	0x0298	W	0x00000004	usbphy control register
GRF_USBPHY0_CON7	0x029c	W	0x000068c0	usbphy control register
GRF_USBPHY1_CON0	0x02a0	W	0x00008618	usbphy control register
GRF_USBPHY1_CON1	0x02a4	W	0x0000e007	usbphy control register
GRF_USBPHY1_CON2	0x02a8	W	0x000082aa	usbphy control register
GRF_USBPHY1_CON3	0x02ac	W	0x00000200	usbphy control register
GRF_USBPHY1_CON4	0x02b0	W	0x00000002	usbphy control register
GRF_USBPHY1_CON5	0x02b4	W	0x00000000	usbphy control register
GRF_USBPHY1_CON6	0x02b8	W	0x00000004	usbphy control register
GRF_USBPHY1_CON7	0x02bc	W	0x000068c0	usbphy control register
GRF_UOC_STATUS0	0x02c0	W	0x00000000	SoC status register 0
GRF_CHIP_TAG	0x0300	W	0x0000293c	chip tag register
GRF_MMC_DET_CNT	0x0304	W	0x0000fdb9	mmc0 detect filter counter register
GRF_EFUSE_PRG_EN	0x037c	W	0x00000000	efuse program register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

5.2.2 Detailed Register Description

GRF_GPIO0A_IOMUX

Address: Operational Base + offset (0x00a8)

GPIO0A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0a7_sel GPIO0A[7] iomux select 01: i2c3_sda 10: hdmi_ddcsda 00: gpio
13:12	RW	0x0	gpio0a6_sel GPIO0A[6] iomux select 01: i2c3_scl 10: hdmi_ddcscl 00: gpio
11:8	RO	0x0	reserved
7:6	RW	0x0	gpio0a3_sel GPIO0A[3] iomux select 01: i2c1_sda 10: mmc1_cmd 00: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio0a2_sel GPIO0A[2] iomux select 1: i2c1_scl 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio0a1_sel GPIO0A[1] iomux select 1: i2c0_sda 0: gpio
1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio0a0_sel GPIO0A[0] iomux select 1: i2c0_scl 0: gpio

GRF_GPIO0B_IOMUX

Address: Operational Base + offset (0x00ac)

GPIO0B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio0b7_sel GPIO0B[7] iomux select 1: hdmi_hotplugin 0: gpio
13:12	RW	0x0	gpio0b6_sel GPIO0B[6] iomux select 01:i2s_sdi 10: spi_csn0 00: gpio
11:10	RW	0x0	gpio0b5_sel GPIO0B[5] iomux select 01:i2s_sdo 10: spi_rxd 00: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio0b4_sel GPIO0B[4] iomux select 1: i2s_lrcktx 0: gpio
7:6	RW	0x0	gpio0b3_sel GPIO0B[3] iomux select 01:i2s_lrckrx 10: spi_txd 00: gpio
5:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x0	gpio0b1_sel GPIO0B[1] iomux select 01:i2s_sclk 10: spi_clk 00: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio0b0_sel GPIO0B[0] iomux select 1: i2s_mclk 0: gpio

GRF_GPIO0C_IOMUX

Address: Operational Base + offset (0x00b0)

GPIO0C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio0c7_sel GPIO0C[7] iomux select 1: nand_cs1 0: gpio
13:9	RO	0x0	reserved
8	RW	0x0	gpio0c4_sel GPIO0C[4] iomux select 1: hdmi_cecsda 0: gpio
7:4	RO	0x0	reserved
3:2	RW	0x0	gpio0c1_sel GPIO0C[1] iomux select 01: sc_io 10: uart0_rstn 00: gpio
1:0	RO	0x0	reserved

GRF_GPIO0D_IOMUX

Address: Operational Base + offset (0x00b4)

GPIOOD iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x0	gpio0d6_sel GPIOOD[6] iomux select 1: mmc1_pwren 0: gpio
11:9	RO	0x0	reserved
8	RW	0x0	gpio0d4_sel GPIOOD[4] iomux select 1:pwm_2 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio0d3_sel GPIOOD[3] iomux select 1: pwm_1 0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio0d2_sel GPIOOD[2] iomux select 1: pwm_0 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio0d1_sel GPIOOD[1] iomux select 1: uart2_ctsn 0: gpio
1:0	RW	0x0	gpio0d0_sel GPIOOD[0] iomux select 01: uart2_rtsn 10: pmic_sleep 00: gpio

GRF_GPIO1A_IOMUX

Address: Operational Base + offset (0x00b8)

GPIO1A iomux control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15	RO	0x0	reserved
14	RW	0x0	<p>gpio1a7_sel</p> <p>GPIO1A[7] iomux select</p> <p>1: mmc0_wrprt</p> <p>0: gpio</p>
13:12	RO	0x0	reserved
11:10	RW	0x3	<p>gpio1a5_sel</p> <p>GPIO1A[5] iomux select</p> <p>01: i2s_sdi</p> <p>10: sdmmc_data3</p> <p>00: gpio</p>
9:8	RW	0x0	<p>gpio1a4_sel</p> <p>GPIO1A[4] iomux select</p> <p>01: i2s_sdo</p> <p>10: sdmmc_data2</p> <p>00: gpio</p>
7	RO	0x0	reserved
6	RW	0x0	<p>gpio1a3_sel</p> <p>GPIO1A[3] iomux select</p> <p>1: i2s_lrcktx</p> <p>0: gpio</p>
5:4	RW	0x0	<p>gpio1a2_sel</p> <p>GPIO1A[2] iomux select</p> <p>01: i2s_lrckrx</p> <p>10: sdmmc_data1</p> <p>00: gpio</p>
3:2	RW	0x0	<p>gpio1a1_sel</p> <p>GPIO1A[1] iomux select</p> <p>01: i2s_sclk</p> <p>10: sdmmc_data0</p> <p>11: pmic_sleep</p> <p>00: gpio</p>

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio1a0_sel GPIO1A[0] iomux select 01: i2s_mclk 10: sdmmc_clkout 11: xin32k 00: gpio

GRF_GPIO1B_IOMUX

Address: Operational Base + offset (0x00bc)

GPIO1B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio1b7_sel GPIO1B[7] iomux select 1: mmc0_cmd 0: gpio
13	RO	0x0	reserved
12	RW	0x0	gpio1b6_sel GPIO1B[6] iomux select 1: mmc0_pwren 0: gpio
11:9	RO	0x0	reserved
8	RW	0x0	gpio1b4_sel GPIO1B[4] iomux select 1: spi_csn1 0: gpio
7:6	RW	0x0	gpio1b3_sel GPIO1B[3] iomux select 01: spi_csn0 10: uart1_rtsn 00: gpio
5:4	RW	0x3	gpio1b2_sel GPIO1B[2] iomux select 01: spi_rxd 10: uart1_sin 00: gpio

Bit	Attr	Reset Value	Description
3:2	RW	0x0	gpio1b1_sel GPIO1B[1] iomux select 01: spi_txd 10: uart1_sout 00: gpio
1:0	RW	0x0	gpio1b0_sel GPIO1B[0] iomux select 01: spi_clk 10: uart1_ctsn 00: gpio

GRF_GPIO1C_IOMUX

Address: Operational Base + offset (0x00c0)

GPIO1C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio1c7_sel GPIO1C[7] iomux select 01: nand_cs3 10: emmc_rstnout 00: gpio
13:12	RW	0x0	gpio1c6_sel GPIO1C[6] iomux select 01: nand_cs2 10: emmc_cmd 00: gpio
11:10	RW	0x0	gpio1c5_sel GPIO1C[5] iomux select 10: jtag_tms when sdmmc0_detectn is invalid 01: mmc0_d3 00: gpio
9:8	RW	0x0	gpio1c4_sel GPIO1C[4] iomux select 10: jtag_tck when sdmmc0_detectn is invalid 01: mmc0_d2 00: gpio

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio1c3_sel GPIO1C[3] iomux select 01: mmc0_d1 10: uart2_rx 00: gpio
5:4	RW	0x0	gpio0c2_sel GPIO0C[2] iomux select 01:mmc0_d0 10: uart2_tx 00:gpio
3	RO	0x0	reserved
2	RW	0x0	gpio1c1_sel GPIO1C[1] iomux select 1: mmc0_detn 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio1c0_sel GPIO1C[0] iomux select 1: mmc0_clkout 0: gpio

GRF_GPIO1D_IOMUX

Address: Operational Base + offset (0x00c4)

GPIO1D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio1d7_sel GPIO1D[7] iomux select 01: nand_d7 10: emmc_d7 11: spi_csn1 00: gpio

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio1d6_sel GPIO1D[6] iomux select 01: nand_d6 10: emmc_d6 11: spi_csn0 00: gpio
11:10	RW	0x0	gpio1d5_sel GPIO1D[5] iomux select 01: nand_d5 10: emmc_d5 11: spi_txd1 00: gpio
9:8	RW	0x0	gpio1d4_sel GPIO1D[4] iomux select 01: nand_d4 10: emmc_d4 11: spi_rxd1 00: gpio
7:6	RW	0x0	gpio1d3_sel GPIO1D[3] iomux select 01: nand_d3 10: emmc_d3 11: sfc_d3 00: gpio
5:4	RW	0x0	gpio1d2_sel GPIO1D[2] iomux select 01: nand_d2 10: emmc_d2 11: sfc_d2 00: gpio
3:2	RW	0x0	gpio1d1_sel GPIO1D[1] iomux select 01: nand_d1 10: emmc_d1 11: sfc_d1 00: gpio
1:0	RW	0x0	gpio1d0_sel GPIO1D[0] iomux select 01: nand_d0 10: emmc_d0 11: sfc_d0 00: gpio

GRF_GPIO2A_IOMUX

Address: Operational Base + offset (0x00c8)

GPIO2A iomux control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:14	RW	0x0	<p>gpio2a7_sel</p> <p>GPIO2A[7] iomux select</p> <p>01: nand_dqs</p> <p>10: emmc_clkout</p> <p>00: gpio</p>
13	RO	0x0	reserved
12	RW	0x0	<p>gpio2a6_sel</p> <p>GPIO2A[6] iomux select</p> <p>1: nand_cs0</p> <p>0: gpio</p>
11:10	RW	0x0	<p>gpio2a5_sel</p> <p>GPIO2A[5] iomux select</p> <p>01: nand_wp</p> <p>10: emmc_pwren</p> <p>00: gpio</p>
9:8	RW	0x0	<p>gpio2a4_sel</p> <p>GPIO2A[4] iomux select</p> <p>01: nand_rdy</p> <p>10: emmc_cmd</p> <p>11: sfc_clk</p> <p>00: gpio</p>
7:6	RW	0x0	<p>gpio2a3_sel</p> <p>GPIO2A[3] iomux select</p> <p>01: nand_rdn</p> <p>10: sfc_csn1</p> <p>00: gpio</p>
5:4	RW	0x0	<p>gpio2a2_sel</p> <p>GPIO2A[2] iomux select</p> <p>01:nand_wrn</p> <p>10: sfc_csn0</p> <p>00: gpio</p>
3:2	RW	0x0	<p>gpio2a1_sel</p> <p>GPIO2A[1] iomux select</p> <p>01:nand_cle</p> <p>00: gpio</p>

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio2a0_sel GPIO2A[0] iomux select 01: nand_ale 10: spi_clk 00: gpio

GRF_GPIO2B_IOMUX

Address: Operational Base + offset (0x00cc)

GPIO2B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2b7_sel GPIO2B[7] iomux select 01: lcdc0_d13 10: ebc_sdce5 11: gmac_rxer 00: gpio
13:12	RW	0x0	gpio2b6_sel GPIO2B[6] iomux select 01: lcdc0_d12 10: ebc_sdce4 11: gmac_clk 00: gpio
11:10	RW	0x0	gpio2b5_sel GPIO2B[5] iomux select 01: lcdc0_d11 10: ebc_sdce3 11: gmac_txen 00: gpio
9:8	RW	0x0	gpio2b4_sel GPIO2B[4] iomux select 01: lcdc0_d10 10: ebc_sdce2 11: gmac_mdio 00: gpio

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio2b3_sel GPIO2B[3] iomux select 01: lcdc0_den 10: ebc_gdclk 11: gmac_rxclk 00: gpio
5:4	RW	0x0	gpio2b2_sel GPIO2B[2] iomux select 01: lcdc0_vsync 10: ebc_sdoe 11: gmac_crs 00: gpio
3:2	RW	0x0	gpio2b1_sel GPIO2B[1] iomux select 01: lcdc0_hsync 10: ebc_sdle 11: gmac_txclk 00: gpio
1:0	RW	0x0	gpio2b0_sel GPIO2B[0] iomux select 01: lcdc0_dclk 10: ebc_sdclk 11: gmac_rxdrv 00: gpio

GRF_GPIO2C_IOMUX

Address: Operational Base + offset (0x00d0)

GPIO2C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:6	RW	0x0	gpio2c3_sel GPIO2C[3] iomux select 01: lcdc0_d17 10: ebc_gdpwr0 11: gmac_txd0 00: gpio

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio2c2_sel GPIO2C[2] iomux select 01: lcdc0_d16 10: ebc_gdsp 11: gmac_txd1 00: gpio
3:2	RW	0x0	gpio2c1_sel GPIO2C[1] iomux select 01: lcdc0_d15 10: ebc_gdoe 11: gmac_rxd0 00: gpio
1:0	RW	0x0	gpio2c0_sel GPIO2C[0] iomux select 01: lcdc0_d14 10: ebc_vcom 11: gmac_rxd1 00: gpio

GRF_GPIO2D_IOMUX

Address: Operational Base + offset (0x00d4)

GPIO2D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14:12	RW	0x0	gpio2d0_sel GPIO2D[0] iomux select 001: lcdc0_d22 010: ebc_gdpwr1 011: gps_clk 100: gmac_col 000: gpio
11:10	RW	0x0	gpio2d5_sel GPIO2D[5] iomux select 01: sc_det 10: uart0_ctsn 00: gpio

Bit	Attr	Reset Value	Description
9:8	RO	0x0	reserved
7:6	RW	0x0	gpio2d3_sel GPIO2D[3] iomux select 01: sc_clk 10: uart0_sin 00: gpio
5:4	RW	0x0	gpio2d2_sel GPIO2D[2] iomux select 01: sc_rst 10: uart0_sout 00: gpio
3:2	RW	0x0	gpio2d1_sel GPIO2D[1] iomux select 01: lcdc0_d23 10: ebc_gdpwr2 11: gmac_mdc 00: gpio
1:0	RO	0x0	reserved

GRF_GPIO3A_IOMUX

Address: Operational Base + offset (0x00d8)

GPIO3A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0	reserved

GRF_GPIO3B_IOMUX

Address: Operational Base + offset (0x00dc)

GPIO3B iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:7	RO	0x0	reserved
6	RW	0x0	gpio3a3_sel GPIO3A[3] iomux select 1: testclk_out 0: gpio
5:0	RO	0x0	reserved

GRF_GPIO3C_IOMUX

Address: Operational Base + offset (0x00e0)

GPIO3D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3:2	RW	0x0	gpio3c1_sel GPIO3C[1] iomux select 1: otg_drvvbus 0: gpio
1:0	RO	0x0	reserved

GRF_GPIO3D_IOMUX

Address: Operational Base + offset (0x00e4)

GPIO3D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:7	RO	0x0	reserved
6	RW	0x0	gpio3d3_sel GPIO3D[3] iomux select 1: spdif_tx 0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio3d2_sel GPIO3D[2] iomux select 1: pwm_irin 0: gpio
3:0	RO	0x0	reserved

GRF_GPIO2C_IOMUX2

Address: Operational Base + offset (0x00e8)

GPIO2C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	gpio2c7_sel GPIO2C[7] iomux select 001: lcdc0_d21 010: ebc_border1 011: gps_mag 100: gmac_txd3 000: gpio
11	RO	0x0	reserved
10:8	RW	0x0	gpio2c6_sel GPIO2C[6] iomux select 001: lcdc0_d20 010: ebc_border0 011: gps_sign 100: gmac_txd2 000: gpio
7	RO	0x0	reserved
6:4	RW	0x0	gpio2c5_sel GPIO2C[5] iomux select 001: lcdc0_d19 010: ebc_sdshr 011: i2c2_scl 100: gmac_rxd2 000: gpio
3	RO	0x0	reserved
2:0	RW	0x0	gpio2c4_sel GPIO2C[4] iomux select 001: lcdc0_d18 010: ebc_gdrl 011: i2c2_sda 100: gmac_rxd3 000: gpio

GRF_CIF_IOMUX

Address: Operational Base + offset (0x00ec)

CIF iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14	RW	0x0	cifd7_sel cif_d7 iomux select 1: ts_d7 0: cif_d7
13	RO	0x0	reserved
12	RW	0x0	cifd6_sel cif_d6 iomux select 1: ts_d6 0: cif_d6
11	RO	0x0	reserved
10	RW	0x0	cifd5_sel cif_d5 iomux select 1: ts_d5 0: cif_d5
9	RO	0x0	reserved
8	RW	0x0	cifd4_sel cif_d4 iomux select 1: ts_d4 0: cif_d4
7	RO	0x0	reserved
6	RW	0x0	cifd3_sel cif_d3 iomux select 1: ts_d3 0: cif_d3
5	RO	0x0	reserved
4	RW	0x0	cifd2_sel cif_d2 iomux select 1: ts_d2 0: cif_d2
3	RO	0x0	reserved
2	RW	0x0	cifd1_sel cif_d1 iomux select 1: ts_d1 0: cif_d1
1	RO	0x0	reserved
0	RW	0x0	cifd0_sel cif_d0 iomux select 1: ts_d0 0: cif_d0

GRF_CIF_IOMUX1

Address: Operational Base + offset (0x00f0)

CIF iomux control register1

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:7	RO	0x0	reserved
6	RW	0x0	cif_clkout_sel cif_clkout iomux select 1: ts_clk 0: cif_clkout
5	RO	0x0	reserved
4	RW	0x0	cif_ckin_sel cif_ckin iomux select 1: ts_valid 0: cif_ckin
3	RO	0x0	reserved
2	RW	0x0	cif_href_sel cif_href iomux select 1: ts_error 0: cif_href
1	RO	0x0	reserved
0	RW	0x0	cif_vsync_sel cif_vsync iomux select 1: ts_sync 0: cif_vsync

GRF_GPIO_DS

Address: Operational Base + offset (0x0100)

GPIO DS control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;</p>
15:12	RO	0x0	reserved
11:10	RW	0x0	<p>gpio_0c4_ds driver strength setting 00: 2.6mA ~ 6.8mA 01: 5.1mA ~ 14mA 10: 7.7mA ~ 20mA 11: 10mA ~ 27mA</p>
9:8	RW	0x0	<p>gpio_0b7_ds driver strength setting 00: 2.6mA ~ 6.8mA 01: 5.1mA ~ 14mA 10: 7.7mA ~ 20mA 11: 10mA ~ 27mA</p>
7:6	RW	0x0	<p>gpio_0a7_ds driver strength setting 00: 2.6mA ~ 6.8mA 01: 5.1mA ~ 14mA 10: 7.7mA ~ 20mA 11: 10mA ~ 27mA</p>
5:4	RW	0x0	<p>gpio_0a6_ds driver strength setting 00: 2.6mA ~ 6.8mA 01: 5.1mA ~ 14mA 10: 7.7mA ~ 20mA 11: 10mA ~ 27mA</p>
3	RW	0x0	<p>gpio_0c4_sl slew rata: 0:slow 1:fast</p>
2	RW	0x0	<p>gpio_0b7_sl slew rata: 0:slow 1:fast</p>
1	RW	0x0	<p>gpio_0a7_sl slew rata: 0:slow 1:fast</p>

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio_0a6_sl slew rata: 0:slow 1:fast

GRF_GPIO0L_PULL

Address: Operational Base + offset (0x0118)

GPIO0A / GPIO0B pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0b_pull GPIO0B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable bit8 - GPIO0B[0] pull up/down control bit9 - GPIO0B[1] pull up/down control bit10 - GPIO0B[2] pull up/down control ... bit15 - GPIO0B[7] pull up/down control
7:0	RW	0x00	gpio0a_pull GPIO0A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable bit0 - GPIO0A[0] pull up/down control bit1 - GPIO0A[1] pull up/down control bit2 - GPIO0A[2] pull up/down control ... bit7 - GPIO0A[7] pull up/down control

GRF_GPIO0H_PULL

Address: Operational Base + offset (0x011c)

GPIO0C / GPIO0D pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;</p>
15:8	RW	0x00	<p>gpio0d_pull GPIO0D pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable bit8 - GPIO0D[0] pull up/down control bit9 - GPIO0D[1] pull up/down control bit10 - GPIO0D[2] pull up/down control ... bit15 - GPIO0D[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio0c_pull GPIO0C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable bit0 - GPIO0C[0] pull up/down control bit1 - GPIO0C[1] pull up/down control bit2 - GPIO0C[2] pull up/down control ... bit7 - GPIO0C[7] pull up/down control</p>

GRF_GPIO1L_PULL

Address: Operational Base + offset (0x0120)

GPIO0A / GPIO0B pull up/down control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:8	RW	0x00	<p>gpio1b_pull</p> <p>GPIO1B pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit8 - GPIO1B[0] pull up/down control</p> <p>bit9 - GPIO1B[1] pull up/down control</p> <p>bit10 - GPIO1B[2] pull up/down control</p> <p>...</p> <p>bit15 - GPIO1B[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio1a_pull</p> <p>GPIO1A pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit0 - GPIO1A[0] pull up/down control</p> <p>bit1 - GPIO1A[1] pull up/down control</p> <p>bit2 - GPIO1A[2] pull up/down control</p> <p>...</p> <p>bit7 - GPIO1A[7] pull up/down control</p>

GRF_GPIO1H_PULL

Address: Operational Base + offset (0x0124)

GPIO1C / GPIO1D pull up/down control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:8	RW	0x00	<p>gpio1d_pull</p> <p>GPIO1D pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit8 - GPIO1D[0] pull up/down control</p> <p>bit9 - GPIO1D[1] pull up/down control</p> <p>bit10 - GPIO1D[2] pull up/down control</p> <p>...</p> <p>bit15 - GPIO1D[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio1c_pull</p> <p>GPIO1C pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit0 - GPIO1C[0] pull up/down control</p> <p>bit1 - GPIO1C[1] pull up/down control</p> <p>bit2 - GPIO1C[2] pull up/down control</p> <p>...</p> <p>bit7 - GPIO1C[7] pull up/down control</p>

GRF_GPIO2L_PULL

Address: Operational Base + offset (0x0128)

GPIO2A / GPIO2B pull up/down control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:8	RW	0x00	<p>gpio2b_pull</p> <p>GPIO2B pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit8 - GPIO2B[0] pull up/down control</p> <p>bit9 - GPIO2B[1] pull up/down control</p> <p>bit10 - GPIO2B[2] pull up/down control</p> <p>...</p> <p>bit15 - GPIO2B[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio2a_pull</p> <p>GPIO2A pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit0 - GPIO2A[0] pull up/down control</p> <p>bit1 - GPIO2A[1] pull up/down control</p> <p>bit2 - GPIO2A[2] pull up/down control</p> <p>...</p> <p>bit7 - GPIO2A[7] pull up/down control</p>

GRF_GPIO2H_PULL

Address: Operational Base + offset (0x012c)

GPIO2C / GPIO2D pull up/down control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:8	RW	0x00	<p>gpio2d_pull</p> <p>GPIO2d pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit8 - GPIO2D[0] pull up/down control</p> <p>bit9 - GPIO2D[1] pull up/down control</p> <p>bit10 - GPIO2D[2] pull up/down control</p> <p>...</p> <p>bit15 - GPIO2D[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio2c_pull</p> <p>GPIO2C pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit0 - GPIO2C[0] pull up/down control</p> <p>bit1 - GPIO2C[1] pull up/down control</p> <p>bit2 - GPIO2C[2] pull up/down control</p> <p>...</p> <p>bit7 - GPIO2C[7] pull up/down control</p>

GRF_GPIO3L_PULL

Address: Operational Base + offset (0x0130)

GPIO3A / GPIO3B pull up/down control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:8	RW	0x00	<p>gpio3b_pull</p> <p>GPIO3B pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit8 - GPIO3B[0] pull up/down control</p> <p>bit9 - GPIO3B[1] pull up/down control</p> <p>bit10 - GPIO3B[2] pull up/down control</p> <p>...</p> <p>bit15 - GPIO3B[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio3a_pull</p> <p>GPIO3A pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit0 - GPIO3A[0] pull up/down control</p> <p>bit1 - GPIO3A[1] pull up/down control</p> <p>bit2 - GPIO3A[2] pull up/down control</p> <p>...</p> <p>bit7 - GPIO3A[7] pull up/down control</p>

GRF_GPIO3H_PULL

Address: Operational Base + offset (0x0134)

GPIO3C / GPIO3D pull up/down control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:8	RW	0x00	<p>gpio3d_pull</p> <p>GPIO3d pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit8 - GPIO3D[0] pull up/down control</p> <p>bit9 - GPIO3D[1] pull up/down control</p> <p>bit10 - GPIO3D[2] pull up/down control</p> <p>...</p> <p>bit15 - GPIO3D[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio3c_pull</p> <p>GPIO3C pull up/down enable</p> <p>Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO.</p> <p>0: pull up/down enable, PAD type will decide to be up or down ,not related with this value</p> <p>1: pull up/down disable</p> <p>bit0 - GPIO3C[0] pull up/down control</p> <p>bit1 - GPIO3C[1] pull up/down control</p> <p>bit2 - GPIO3C[2] pull up/down control</p> <p>...</p> <p>bit7 - GPIO3C[7] pull up/down control</p>

GRF_ACODEC_CON

Address: Operational Base + offset (0x013c)

SoC control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable bit 0~bit 15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;</p>
15:6	RO	0x0	reserved
5:4	RW	0x2	<p>acodec_detectn_debounce_sel acodec_detectn debounce time select 00:5ms 01:15ms 10:35ms 11:50ms</p>
3	RW	0x0	<p>acodec_detectn_fall_int_en acodec detectn negedge interrupt enable 0: interrupt disable 1: interrupt enable</p>
2	RW	0x0	<p>acodec_detectn_rise_int_en acodec detectn posedge interrupt enable 0: interrupt disable 1: interrupt enable</p>
1	RW	0x0	<p>acodec_detectn_fall_int_pd acodec detectn negedge interrupt pending bit wirte 1 to it, it will be cleared.</p>
0	RW	0x0	<p>acodec_detectn_rise_int_pd acodec detectn posedge interrupt pending bit wirte 1 to it, it will be cleared.</p>

GRF_SOC_CON0

Address: Operational Base + offset (0x0140)

SoC control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable bit 0~bit 15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;</p>
15	RW	0x0	<p>hdmiphy_dclk_sel 1'b0: dclk from lcdc 1'b1: dclk from cru</p>
14	RW	0x0	<p>ddr_16bit_en When 16bit ddr is used , this bit should be 1.</p>
13	RW	0x0	<p>msch4_mainpartialpop 0:16bit ddr 1:8bit ddr</p>
12	RW	0x0	<p>soc_remap remap bit control When soc_remap = 1, the bootrom is mapped to address 0x10100000 and internal memory is mapped to address 0x0.</p>
11	RW	0x0	<p>acodec_ad2da_loop acodec loopback enable 0 : acodec loopback disable 1: acodec loopback enable</p>
10	RW	0x0	<p>acodec_sel 0: i2s_sdi from gpio is selected 1: i2s_sdi from acodec is selected</p>
9	RO	0x0	reserved
8	RW	0x1	<p>force_jtag this bit is used to force iomux to jtag. 0: disable 1: enable</p>
7	RW	0x0	<p>mobile_ddr_sel this bit is used to tell ddr monitor the type of ddr used. 0: DDR2/DDR3 1: LPDDR2/LPDDR3</p>
6	RW	0x0	<p>dfl_eff_stat_en dfl monitor start to work. 1: dfl monitor works. 0: dfl monitor stops.</p>

Bit	Attr	Reset Value	Description
5:4	RW	0x2	sd_detectn_debounce_sel sd_detectn debounce time select 00:5ms 01:15ms 10:35ms 11:50ms
3	RW	0x0	sd_detectn_fall_int_en SD detectn negedge interrupt enable 0: interrupt disable 1: interrupt enable
2	RW	0x0	sd_detectn_rise_int_en SD detectn posedge interrupt enable 0: interrupt disable 1: interrupt enable
1	RW	0x0	sd_detectn_fall_int_pd SD detectn negedge interrupt pending bit wirte 1 to it, it will be cleared.
0	RW	0x0	sd_detectn_rise_int_pd SD detectn posedge interrupt pending bit wirte 1 to it, it will be cleared.

GRF_SOC_CON1

Address: Operational Base + offset (0x0144)

SoC control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit 0~bit 15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	hevc_vpu_sel hevc vpu select 0: select vpu 1: select hevc
14	RW	0x0	mipi_phy_lane3_enable MIPI phy enalbe lane3 in TTL mode 0: not TTL mode 1: TTL mode

Bit	Attr	Reset Value	Description
13	RW	0x0	mipi_phy_lane2_enable MIPI phy enable lane2 in TTL mode 0: not TTL mode 1: TTL mode
12	RW	0x0	mipi_lane1_enable MIPI phy enable lane1 in TTL mode 0: not TTL mode 1: TTL mode
11	RW	0x0	mipi_phy_lane0_enable MIPI phy enable lane0 in TTL mode 0: not TTL mode 1: TTL mode
10	RW	0x0	vpu_sel vdpu vepu clock select 0: select vepu aclk as vpu main clock 1: select vdpu aclk as vpu main clock
9:8	RO	0x0	reserved
7	RW	0x0	mipi_phy_enableck MIPI phy enable ck in TTL mode 0: not TTL mode 1: TTL mode
6	RW	0x0	emmc_iomux_sel emmc iomux select 0: select 3026 sdmmc iomux 1: select new iomux
5	RW	0x0	i2s_iomux_sel i2s iomux select 0: select 3026 sdmmc iomux 1: select new iomux
4:3	RW	0x0	spi_iomux_sel spi iomux select 00: select 3026 sdmmc iomux 01: select new iomux1 10: select new iomux2
2:0	RO	0x0	reserved

GRF_SOC_CON2

Address: Operational Base + offset (0x0148)

SoC control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15	RW	0x0	crypto_pwr_idlereq NOC idle request, high valid.
14	RW	0x0	msch_pwr_idlereq NOC idle request, high valid.
13	RW	0x0	core_pwr_idlereq NOC idle request, high valid.
12	RW	0x0	peri_pwr_idlereq NOC idle request, high valid.
11	RW	0x0	vio_pwr_idlereq NOC idle request, high valid.
10	RW	0x0	vpu_pwr_idlereq NOC idle request, high valid.
9	RW	0x0	gpu_pwr_idlereq NOC idle request, high valid.
8	RW	0x0	sys_pwr_idlereq NOC idle request, high valid.
7	RW	0x1	msch4_mainddr3 When DDR3 is used , software should configure this bit to 1.
6:4	RO	0x0	reserved
3	RW	0x0	usb_host_sel usb host select 0: select ehci usb host 1: select old usb host
2	RW	0x1	ddrphy_low_power_en ddrphy low power enable 1'b0: ddrphy into low-power 1'b1: normal
1	RW	0x0	upctl_c_sysreq software config enter DDR self-refresh by lowpower interface 1'b1: request enter self-refresh 1'b0: not enter self-refresh

Bit	Attr	Reset Value	Description
0	RW	0x0	upctl_c_active_in ddr clock active in. External signal from system that flags if a hardware low power request can be accepted or should always be denied. 0: may be accepted 1: will be denied

GRF_SOC_STATUS0

Address: Operational Base + offset (0x014c)

SoC status register

Bit	Attr	Reset Value	Description
31	RO	0x0	acodec_hpdet The flag indicates whether has the headset to be inserted. 1: having headset to be inserted. 0:don't have.
30	RO	0x0	reserved
29	RO	0x0	crypto_pwr_idle NOC idle state. "1" indicates idle.
28	RO	0x0	msch_pwr_idle NOC idle state. "1" indicates idle.
27	RO	0x0	sys_pwr_idle NOC idle state. "1" indicates idle.
26	RO	0x0	gpu_pwr_idle NOC idle state. "1" indicates idle.
25	RO	0x0	vpu_pwr_idle NOC idle state. "1" indicates idle.
24	RO	0x0	vio_pwr_idle NOC idle state. "1" indicates idle.
23	RO	0x0	peri_pwr_idle NOC idle state. "1" indicates idle.
22	RO	0x0	core_pwr_idle NOC idle state. "1" indicates idle.
21	RO	0x0	crypto_pwr_idleack NOC idle acknowledge. high valid.
20	RO	0x0	msch_pwr_idleack NOC idle acknowledge. high valid.
19	RO	0x0	sys_pwr_idleack NOC idle acknowledge. high valid.
18	RO	0x0	gpu_pwr_idleack NOC idle acknowledge. high valid.
17	RO	0x0	vpu_pwr_idleack NOC idle acknowledge. high valid.
16	RO	0x0	vio_pwr_idleack NOC idle acknowledge. high valid.
15	RO	0x0	peri_pwr_idleack NOC idle acknowledge. high valid.
14	RO	0x0	core_pwr_idleack NOC idle acknowledge. high valid.

Bit	Attr	Reset Value	Description
13	RO	0x0	host20_iddig host2.0 iddig state. it will always be "0".
12:11	RO	0x0	host20_linestate host 2.0 linestate status This bus reflects the state of the single-ended receivers. In Suspend or Sleep mode, this bus is a combinatorial output (directly reflecting the current state of D- and D+, respectively). 2'b11: SE1 (D+ high, D- high) 2'b10: K state for high-speed and full-speed USB traffic; J state for low-speed USB traffic (D+ low, D- high) 2'b01: J state for high-speed and full-speed USB traffic; K state for low-speed USB traffic (D+ high, D- low) 2'b00: SE0 (D+ low, D- low) During normal high-speed packet transfers, the line indicates a high-speed J state.
10	RO	0x0	host20_bvalid host 2.0 bvalid status B-Device Session Valid Indicator Function: This controller signal is output from the USB 2.0 Session Valid comparator and indicates whether the session for a B-device is valid. 1: The session for the B-device is valid. 0: The session for the B-device is not valid.
9	RO	0x0	host20_vbusvalid host 2.0 vbus valid status VBUS Valid Indicator Function: This controller signal is output from the USB 2.0 VBUS Valid comparator and indicates whether the VBUS output is at a valid level. 1: The VBUS output is valid. 0: The VBUS output is not valid.
8	RO	0x0	otg0_iddig otg iddig status 0: indicate otg work as host 1: indicate otg work as device

Bit	Attr	Reset Value	Description
7:6	RO	0x0	<p>otg_linestate otg linestate status This bus reflects the state of the single-ended receivers. In Suspend or Sleep mode, this bus is a combinatorial output (directly reflecting the current state of D- and D+, respectively).</p> <p>2'b11: SE1 (D+ high, D- high) 2'b10: K state for high-speed and full-speed USB traffic; J state for low-speed USB traffic (D+ low, D- high) 2'b01: J state for high-speed and full-speed USB traffic; K state for low-speed USB traffic (D+ high, D- low) 2'b00: SE0 (D+ low, D- low)</p> <p>During normal high-speed packet transfers, the line indicates a high-speed J state.</p>
5	RO	0x0	<p>otg_bvalid otg bvalid status B-Device Session Valid Indicator Function: This controller signal is output from the USB 2.0 Session Valid comparator and indicates whether the session for a B-device is valid. 1: The session for the B-device is valid. 0: The session for the B-device is not valid.</p>
4	RO	0x0	<p>otg_vbusvalid otg vbus valid status VBUS Valid Indicator Function: This controller signal is output from the USB 2.0 VBUS Valid comparator and indicates whether the VBUS output is at a valid level. 1: The VBUS output is valid. 0: The VBUS output is not valid.</p>
3:0	RO	0x0	<p>pll_lock PLL lock status :generalpll_lock, codecpll_lock, armpll_lock, ddrpll_lock 1: pll is lock 0: pll is unlock</p>

GRF_LVDS_CON0

Address: Operational Base + offset (0x0150)

LVDS control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable</p> <p>bit0 ~ bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:14	RO	0x0	reserved
13	RW	0x0	<p>mipi_phy_lane3_forcemode</p> <p>MIPI phy lane3 force x mode</p> <p>0: disable</p> <p>1: enable</p>
12	RW	0x0	<p>mipi_phy_lane2_forcemode</p> <p>MIPI phy lane2 force x mode</p> <p>0: disable</p> <p>1: enable</p>
11	RW	0x0	<p>mipi_phy_lane1_forcemode</p> <p>MIPI phy lane1 force x mode</p> <p>0: disable</p> <p>1: enable</p>
10	RW	0x0	<p>mipi_phy_lane0_forcemode</p> <p>MIPI phy lane0 force x mode</p> <p>0: disable</p> <p>1: enable</p>
9	RW	0x0	<p>mipi_dsi_forcemode</p> <p>MIPI phy force x mode</p> <p>0: disable</p> <p>1: enable</p>
8	RW	0x0	<p>mipi_phy_lane0_turndisable</p> <p>MIPI phy lane0 turndisable</p>
7	RW	0x0	<p>mipi_phy_ttl_mode</p> <p>MIPI phy work in TTL mode</p> <p>0: disable</p> <p>1: enable</p>
6	RW	0x0	<p>lvds_mode</p> <p>work in lvds mode</p> <p>0: disable</p> <p>1: enable</p>
5	RW	0x0	<p>mipi_ctrl_dpcolor</p> <p>MIPI controller dpi color</p> <p>0: disable</p> <p>1: enable</p>

Bit	Attr	Reset Value	Description
4	RW	0x0	mipi_ctrl_dpishutdown MIPI controller dpi shut down 0: disable 1: enable
3	RW	0x0	lvds_msbsel LVDS lane input format 0: MSB is on D0 1: MSB is on D7
2:1	RW	0x0	lvds_select LVDS output format 00: 8bit mode format-1 01: 8bit mode format-2 10: 8bit mode format-3 11: 6bit mode
0	RW	0x0	ebc_mac_sel LVDS data from ebc or mac or lvds selection 1'b0 : lvds 1'b1 : ebc

GRF_DMAC_CON0

Address: Operational Base + offset (0x015c)

DMAC control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	dmac_boot_from_pc DMAC boot_from_pc input control Controls the location in which the DMAC executes its initial instruction, after it exits from reset : 0= DMAC waits for an instruction from APB interface 1= DMAC manager thread executes the instruction that is located at the address that boot_addr[31:0] provided.

GRF_DMAC_CON1

Address: Operational Base + offset (0x0160)

DMAC control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;</p>
15:0	RW	0x0000	<p>dmac_boot_addr dmac_boot_addr[27:12] DMAC boot_addr[27:12] input control Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.</p>

GRF_DMAC_CON2

Address: Operational Base + offset (0x0164)

DMAC control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;</p>
15:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x1	dmac_drtype DMAC type of acknowledgement or request for peripheral signals: 00 : single level request 01 : burst level request 10 : acknowledging a flush request 11 : reserved
3:0	RW	0x0	dmac_boot_addr dmac_boot_addr[31:28] DMAC boot_addr[31:28] input control Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

GRF_MAC_CON0

Address: Operational Base + offset (0x0168)

GMAC control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	rxclk_dly_ena_gmac RGMII RX clock delayline enable 1'b1: enable 1'b0: disable
14	RW	0x0	txclk_dly_ena_gmac RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
13:7	RW	0x10	clk_rx_dl_cfg_gmac RGMII RX clock delayline value
6:0	RW	0x10	clk_tx_dl_cfg_gmac RGMII TX clock delayline value

GRF_MAC_CON1

Address: Operational Base + offset (0x016c)

GMAC control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x1	rmii_mode RMII mode selection 1'b1: RMII mode
13:12	RW	0x0	gmac_clk_sel RGMII clock selection 2'b00: 125MHz 2'b11: 25MHz 2'b10: 2.5MHz
11	RW	0x0	rmii_clk_sel RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz
10	RW	0x0	gmac_speed MAC speed 1'b1: 100-Mbps 1'b0: 10-Mbps
9	RW	0x0	gmac_flowctrl GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
8:6	RW	0x1	gmac_phy_intf_sel PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved
5:0	RO	0x0	reserved

GRF_TVE_CON

Address: Operational Base + offset (0x0170)

TV encoder control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12:7	RW	0x00	gain gain
6	RW	0x0	enctr2 enctr2
5	RW	0x0	enctr1 enctr1
4	RW	0x0	enctr0 enctr0
3	RW	0x0	ensc0 ensc0
2	RW	0x0	endac endac
1	RW	0x0	envbg envbg
0	RW	0x0	enextref enextref

GRF_UOC0_CON0

Address: Operational Base + offset (0x017c)

OTG control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable bit 0 ~ bit 15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;</p>
15	RW	0x0	<p>otg0_bvalid_irq_pd otg0 bvalid interrupt pending bit write 1 to this bit , it will be cleared.</p>
14	RW	0x0	<p>otg0_bvalid_irq_en otg0 bvalid interrupt enable 1: interrupt enable 0: interrupt disable</p>
13	RW	0x0	<p>otg0_linestate_irq_pd otg0 linestate interrupt pending write 1 to this bit , it will be cleared.</p>
12	RW	0x0	<p>otg0_linestate_irq_en otg0 linestate change interrupt enable 1: interrupt enable 0: interrupt disable</p>
11	RO	0x0	reserved
10	RW	0x0	<p>iddig_status control software iddig value 0:host 1:device</p>
9	RW	0x0	<p>iddig_sft_sel 0 : iddig to otg controller select usbphy output 1: iddig to otg controller select grf_uoc0_con5[10]</p>
8	RW	0x0	<p>utmi_dmpulldown 0: DM 15 KOhm pull down disabled 1: DM 15 Kohm pull down enable</p>
7	RW	0x0	<p>utmi_dppulldown 0: DP 15 KOhm pull down disabled 1: DP 15 Kohm pull down enable</p>
6	RW	0x0	<p>utmi_termselect USB Termination Select 1: Full-speed terminations are enabled. 0: High-speed terminations are enabled.</p>

Bit	Attr	Reset Value	Description
5:4	RW	0x0	utmi_xcvrselect Transceiver Select 11: Sends an LS packet on an FS bus or receives an LS packet. 10: LS Transceiver 01: FS Transceiver 00: HS Transceiver
3:2	RW	0x0	utmi_opmode UTMI+ Operational Mode Function: This controller bus selects the UTMI+ operational mode. 11: Normal operation without SYNC or EOP generation. If the XCVRSEL bus is not set to 00 while OPMODE[1:0] is set to 11, USB PHY behavior is undefined. 10: Disable bit stuffing and NRZI encoding 01: Non-Driving 00: Normal
1	RW	0x0	utmi_suspend_n Suspend Assertion 1: Normal operating mode 0: Suspend mode
0	RW	0x0	usbphy_soft_con_sel 0: software control usb phy disable 1 : software control usb phy enable

GRF_UOC1_CON1

Address: Operational Base + offset (0x0184)

usb host control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x0	usbphy1_vdm_src_en open dm voltage source
11	RW	0x0	usbphy1_vdp_src_en open dp voltage source

Bit	Attr	Reset Value	Description
10	RW	0x0	usbphy1_rdm_pdwn_en open dm pull down resistor
9	RW	0x0	usbphy1_idp_src_en open dp source current
8	RW	0x0	usbphy1_idm_sink_en open dm sink current enable
7	RW	0x0	usbphy1_idp_sink_en open dp sink current enable
6:0	RO	0x0	reserved

GRF_UOC1_CON2

Address: Operational Base + offset (0x0188)

UOC1 control register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	usbhost0_incr4_en USB HOST0 incr4_en bit control
14	RW	0x1	usbhost0_incr16_en USB HOST0 incr16_en bit control
13	RW	0x0	usbhost0_hubsetup_min USB HOST0 hubsetup_min bit control
12	RW	0x0	usbhost0_app_start_clk USB HOST0 app_start_clk bit control
11:6	RW	0x20	usbhost0_fladj_val_common USB HOST0 fladj_val_common bit control
5:0	RW	0x20	usbhost0_fladj USB HOST0 fladj bit control

GRF_UOC1_CON3

Address: Operational Base + offset (0x018c)

UOC1 control register 3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usbhost0_ohci_susp_lgcy USB HOST0 ohci_susp_lgcy bit control
14	RW	0x0	usbhost0_ohci_cntsel USB HOST0 ohci_cntsel bit control
13	RW	0x0	usbhost0_app_prt_ovrcur USB HOST0 app_prt_ovrcur bit control
12	RO	0x0	reserved
11	RW	0x1	usbhost0_word_if USB HOST0 word_if bit control
10	RW	0x0	usbhost0_sim_mode USB HOST0 sim_mode bit control
9	RW	0x1	usbhost0_incrx_en USB HOST0 incr_x_en bit control
8	RW	0x1	usbhost0_incr8_en USB HOST0 incr8_en bit control
7	RO	0x0	reserved
6	RW	0x1	usbhost0_ohci_clkcktrst USB HOST0 ohci_clkcktrst bit control
5:1	RO	0x0	reserved
0	RW	0x0	usbhost0_autoppd_on_overcur USB HOST0 autoppd_on_overcur bit control

GRF_UOC1_CON4

Address: Operational Base + offset (0x0190)
 USB HOST 2.0 control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15	RW	0x0	usbphy_commonon USBPHY common on
14	RO	0x0	reserved
13	RW	0x0	bypasssel0 Transmitter Digital Bypass mode Enable. When 1, otg used as a uart port.
12	RW	0x0	bypassdmen0 DM0 Transmitter Digital Bypass Enable. high valid.
11	RW	0x0	host20disable when 1, host 2.0 phy disable
10	RW	0x0	otgphydisable when 1, otgphy is disabled.
9:8	RO	0x0	reserved
7:6	RW	0x0	utmihost_scaledown_mode utmihost scaledown mode
5	RW	0x0	<p>utmiotg_idpullup</p> <p>Analog ID Input Sample Enable</p> <p>Function: This controller signal controls ID line sampling.</p> <p>1: ID pin sampling is enabled, and the IDDIG output is valid.</p> <p>0: ID pin sampling is disabled, and the IDDIG output is not valid.</p>
4	RW	0x1	utmiotg_dppulldown D+ Pull-Down Resistor Enable
3	RW	0x1	utmiotg_dmpulldown D- Pull-Down Resistor Enable
2	RW	0x1	<p>utmiotg_drvvbus</p> <p>Drive VBUS</p> <p>1: The VBUS Valid comparator is enabled.</p> <p>0: The VBUS Valid comparator is disabled.</p>
1	RW	0x0	utmisrp_chrgvbus VBUS Input Charge Enable
0	RW	0x0	utmisrp_dischrgvbus VBUS Input Discharge Enable

GRF_UOC1_CON5

Address: Operational Base + offset (0x0194)

USB HOST 2.0 control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable</p> <p>bit 0 ~ bit 15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15	RW	0x0	<p>otg1_linestate_irq_pd</p> <p>otg1 linestate interrupt pending</p> <p>write 1 to this bit , it will be cleared.</p>
14	RW	0x0	<p>otg1_linestate_irq_en</p> <p>otg1 linestate interrupt enable</p>
13:9	RO	0x0	reserved
8	RW	0x0	<p>utmi_dmpulldown</p> <p>0: DM 15 KOhm pull down disabled</p> <p>1: DM 15 Kohm pull down enable</p>
7	RW	0x0	<p>utmi_dppulldown</p> <p>0: DP 15 KOhm pull down disabled</p> <p>1: DP 15 Kohm pull down enable</p>
6	RW	0x0	<p>utmi_termselect</p> <p>USB Termination Select</p> <p>1: Full-speed terminations are enabled.</p> <p>0: High-speed terminations are enabled.</p>
5:4	RW	0x0	<p>utmi_xcvrselect</p> <p>Transceiver Select</p> <p>11: Sends an LS packet on an FS bus or receives an LS packet.</p> <p>10: LS Transceiver</p> <p>01: FS Transceiver</p> <p>00: HS Transceiver</p>

Bit	Attr	Reset Value	Description
3:2	RW	0x0	<p>utmi_opmode UTMI+ Operational Mode Function: This controller bus selects the UTMI+ operational mode.</p> <p>11: Normal operation without SYNC or EOP generation. If the XCVRSEL bus is not set to 00 while OPMODE[1:0] is set to 11, USB PHY behavior is undefined.</p> <p>10: Disable bit stuffing and NRZI encoding</p> <p>01: Non-Driving</p> <p>00: Normal</p>
1	RW	0x0	<p>utmi_suspend_n Suspend Assertion 1: Normal operating mode 0: Suspend mode</p>
0	RW	0x0	<p>usbphy_soft_con_sel 0: software control usb phy disable 1 : software control usb phy enable</p>

GRF_DDRC_STAT

Address: Operational Base + offset (0x019c)

DDRC status

Bit	Attr	Reset Value	Description
31:21	RW	0x000	<p>gpu_idle gpu idle staus</p>
20	RW	0x0	<p>ddrupctl_c_active confirm that system external to PCTL can accept a Low-power request. high valid.</p>
19	RW	0x0	<p>upctl_c_sysack PCTL low-power request status response. high valid.</p>
18:16	RO	0x0	<p>ddrupctl_stat Current state of the protocol controller 3'b000 = Init_mem 3'b001 = Config 3'b010 = Config_req 3'b011 = Access 3'b100 = Access_req 3'b101 = Low_power 3'b110 = Low_power_entry_req 3'b111 = Low_power_exit_req</p>
15:0	RO	0x0000	<p>ddrupctl_bbflags Bank busy indication NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on when precharge, activates, reads/writes. Bit0 indication Bank0 busy, bit1 indication Bank1 busy, and so on.</p>

GRF_SOC_STATUS1

Address: Operational Base + offset (0x01a4)

SoC status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10	RO	0x0	mipi_ctrl_edpihalt
9	RO	0x0	mipi_ctrl_shutdown
8	RO	0x0	mipi_ctrl_rstz
7	RO	0x0	mipi_ctrl_forcepll
6	RO	0x0	gmac_portselect
5:0	RO	0x00	timer_en_status bit 0 : timer 0 enable status bit 1 : timer 1 enable status 1 means timer is enabled.

GRF_CPU_CON0

Address: Operational Base + offset (0x01a8)

CPU control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0 ~ bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	l2_data_latency select L2 data ram write latency: 0: new design 1: old design
14	RO	0x0	reserved
13	RW	0x1	deviceen_dap Enabling access to the connected debug device or memory system 0: disable 1: enable

Bit	Attr	Reset Value	Description
12	RW	0x0	I2rstdisable Disable automatic L2 cache invalidate at reset. high valid.
11:8	RW	0x0	I1rstdisable Disable automatic data cache, instruction cache and TLB invalidate at reset. 4bits corresponding 4 cores. high valid.
7:3	RO	0x0	reserved
2:0	RW	0x2	ema_mem_ctrl memory EMA signal control

GRF_CPU_CON1

Address: Operational Base + offset (0x01ac)

CPU control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0 ~ bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:8	RW	0x0	cfgte_a7 Controls processor state for exception handling (TE bit) at reset.
7:4	RW	0x0	vinithi_a7 Cortex-A7 vinithi bit control. location of the exception vectors at reset. Sampled during reset. 0= 0x0000_0000 1= 0xffff_0000
3:0	RW	0x0	cfgend_a7 One bit for each processor. 0 = Little-endian 1 = Big-endian

GRF_CPU_CON2

Address: Operational Base + offset (0x01b0)

CPU control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	wirte_enable bit0 ~ bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:9	RO	0x0	reserved
8	RW	0x0	cfgsdisable Disables write access to some secure GIC registers. When CFGSDISABLE is asserted, the GIC prevents writes to any register locations that control the operating state of an LSPI 1'b0: enable 1'b1: disable
7	RW	0x0	evento_clear Event output. evento is active when one SEV instruction is executed. this bit used to clear evento signal. 1'b0: un-clear 1'b1: clear
6	RW	0x0	eventi Event input for processor wake-up from WFE state. This pin must be asserted for at least one CLKIN clock cycle. When this signal is asserted, it acts as a WFE wake-up event to all the processors in the multiprocessor device.
5	RW	0x1	dbgselfaddrv Debug self-address offset valid 1'b0: invalid 1'b1: valid
4	RW	0x1	dbgromaddrv Debug ROM physical address valid: 1'b0: invalid 1'b1: valid
3	RW	0x1	spniden Secure privileged non-invasive debug enable
2	RW	0x1	niden Non-invasive debug enable

Bit	Attr	Reset Value	Description
1	RW	0x1	spiden Secure privileged invasive debug enable
0	RW	0x1	dbgen Debug enable

GRF_CPU_CON3

Address: Operational Base + offset (0x01b4)

CPU control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0 ~ bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0	reserved

GRF_CPU_STATUS0

Address: Operational Base + offset (0x01c0)

CPU status register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:3	RO	0x0	smpnamp_a7 Signals AMP or SMP mode for each Cortex-A7 processor. 0 Asymmetric. 1 Symmetric.
2	RO	0x0	jtagnsw_dap coresight jtagnsw signal status 1: JTAG is selected. 0: SWD is selected.
1	RO	0x0	jtagtop_dap coresight jtagtop signal status "1" means jtag state machine is in one of the top four modes: test-logic-reset, run-test/idle, select-DR-scan, select-IR-scan.
0	RO	0x0	evento_rising_edge evento signal rising edge status

GRF_CPU_STATUS1

Address: Operational Base + offset (0x01c4)

CPU status register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:5	RO	0x0	core_wfi_status core WFI status, 4bit corresponding 4 cores.
4:1	RO	0x0	core_wfe_status core WFE status, 4bit corresponding 4 cores.
0	RO	0x0	I2c_wfi_status L2 WFI status.

GRF_OS_REG0

Address: Operational Base + offset (0x01c8)

software OS register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG1

Address: Operational Base + offset (0x01cc)

software OS register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG2

Address: Operational Base + offset (0x01d0)

software OS register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG3

Address: Operational Base + offset (0x01d4)

software OS register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG4

Address: Operational Base + offset (0x01d8)

software OS register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG5

Address: Operational Base + offset (0x01dc)
software OS register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG6

Address: Operational Base + offset (0x01e0)
software OS register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG7

Address: Operational Base + offset (0x01e4)
software OS register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_PVTM_CON0

Address: Operational Base + offset (0x0200)
PVTM control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	pvtm_func_osc_en func PVT monitor oscillator enable 1'b1: enable 1'b0: disable
12	RW	0x0	pvtm_func_start func PVT monitor start control
11:10	RO	0x0	reserved
9	RW	0x0	pvtm_gpu_osc_en pd_gpu PVT monitor oscillator enable 1'b1: enable 1'b0: disable
8	RW	0x0	pvtm_gpu_start pd_gpu PVT monitor start control
7:2	RO	0x0	reserved
1	RW	0x0	pvtm_core_osc_en pd_core PVT monitor oscillator enable 1'b1: enable 1'b0: disable
0	RW	0x0	pvtm_core_start pd_core PVT monitor start control

GRF_PVTM_CON1

Address: Operational Base + offset (0x0204)

PVTM control register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_core_cal_cnt pd_core pvtm calculator counter

GRF_PVTM_CON2

Address: Operational Base + offset (0x0208)

PVTM control register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_gpu_cal_cnt pd_gpu pvtm calculator counter

GRF_PVTM_CON3

Address: Operational Base + offset (0x020c)

PVTM control register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_func_cal_cnt func pvtm calculator counter

GRF_PVTM_STATUS0

Address: Operational Base + offset (0x0210)

PVTM status register0

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	pvtm_func_freq_done func pvtm frequency calculate done status
1	RO	0x0	pvtm_core_freq_done pd_core pvtm frequency calculate done status
0	RO	0x0	pvtm_gpu_freq_done pd_gpu pvtm frequency calculate done status

GRF_PVTM_STATUS1

Address: Operational Base + offset (0x0214)

PVTM status register1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_core_freq_cnt pd_core pvtm frequency count

GRF_PVTM_STATUS2

Address: Operational Base + offset (0x0218)

PVTM status register2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_gpu_freq_cnt pd_gpu pvtm frequency count

GRF_PVTM_STATUS3

Address: Operational Base + offset (0x021c)

PVTM status register3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_gpu_freq_cnt pd_gpu pvtm frequency count

GRF_DFI_WRNUM

Address: Operational Base + offset (0x0220)

DFI write number register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_eff_wr_num the total number of write operation on DFI interface.

GRF_DFI_RDNUM

Address: Operational Base + offset (0x0224)

DFI read number register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_eff_rd_num the total number of read operation on DFI interface.

GRF_DFI_ACTNUM

Address: Operational Base + offset (0x0228)

DFI active number register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_eff_act_num the total number of active operation on DFI interface.

GRF_DFI_TIMERVAL

Address: Operational Base + offset (0x022c)

DFI work time

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_timer_val the total time for DFI monitor works.

GRF_NIF_FIFO0

Address: Operational Base + offset (0x0230)

NIF status register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	nif0_fifo0 status for msch4 signals. It will not be cleared by system reset. bit 10 ~ bit 0 : msch4_n_acol[10:0] bit 11 ~ bit 13 : msch4_n_abank[2:0] bit 14 ~ bit 29 : msch4_n_arow[15:0] bit 31 ~ bit 30 : msch4_n_arank_sel[1:0]

GRF_NIF_FIFO1

Address: Operational Base + offset (0x0234)

NIF status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif0_fifo1 status for msch4 signals. It will not be cleared by system reset. bit 10 ~ bit 0 : msch4_n_acol[10:0] bit 11 ~ bit 13 : msch4_n_abank[2:0] bit 14 ~ bit 29 : msch4_n_arow[15:0] bit 31 ~ bit 30 : msch4_n_arank_sel[1:0]

GRF_NIF_FIFO2

Address: Operational Base + offset (0x0238)

NIF status register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	nif0_fifo2 status for msch4 signals. It will not be cleared by system reset. bit 10 ~ bit 0 : msch4_n_acol[10:0] bit 11 ~ bit 13 : msch4_n_abank[2:0] bit 14 ~ bit 29 : msch4_n_arow[15:0] bit 31 ~ bit 30 : msch4_n_arank_sel[1:0]

GRF_NIF_FIFO3

Address: Operational Base + offset (0x023c)

NIF status register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	nif0_fifo3 status for msch4 signals. It will not be cleared by system reset. bit 10 ~ bit 0 : msch4_n_acol[10:0] bit 11 ~ bit 13 : msch4_n_abank[2:0] bit 14 ~ bit 29 : msch4_n_arow[15:0] bit 31 ~ bit 30 : msch4_n_arank_sel[1:0]

GRF_USBPHY0_CON0

Address: Operational Base + offset (0x0280)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RW	0x4	squel_trigger_con bit 2 ~ bit 0 of squel_trigger_con. 0000:112.5mV 1001:162.5mV 1011:175mV 1100:150mV(default) 1110:125mV

Bit	Attr	Reset Value	Description
12:11	RW	0x0	non_driving Registers for non-driving state control. non-driving state is controlled by op-mode by default, when bit[11] is configured with "1", user can control non-driving state through bit[12].
10:8	RW	0x6	tx_clk_phase_con USB Tx Clock phase configure, 000 represent the earliest phase , and 111 the latest, single step delay is 256ps
7:5	RW	0x0	rx_clk_phase_con USB Rx Clock phase configure,000 represent the earliest phase , and 111 the latest, single step delay is 256ps
4:3	RW	0x3	fls_eye_height FS/LS eye height configure , 00 represent the largest slew rate , 11 represent the smallest slew rate
2:0	RW	0x0	hs_eye_diag_adjust HS eye diagram adjust, open HS pre-emphasize function to increase HS slew rate, only used when large cap loading is attached. 001:open pre-emphasize in sof or eop state 010: open pre-emphasize in chirp state 100: open pre-emphasize in non-chirp state 111: always open pre-emphasize other combinations : reserved

GRF_USBPHY0_CON1

Address: Operational Base + offset (0x0284)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:13	RW	0x7	hs_eye_height bit2 ~ bit 0 of hs_eye_height. HS eye height tuning ,more zeros represent bigger eye, more ones represent smaller eye
12:3	RO	0x0	reserved
2	RW	0x1	current_comp_en Enable current compensation, active high.
1	RW	0x1	res_comp_en Enable resistance compensation, active high.
0	RW	0x1	squel_trigger_con bit 3 of squel_trigger_con. 0000:112.5mV 1001:162.5mV 1011:175mV 1100:150mV(default) 1110:125mV

GRF_USBPHY0_CON2

Address: Operational Base + offset (0x0288)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	odt_compensation bit 0 of odt_compensation. ODT Compensation voltage reference 000:200mV 001:187.5mV(default) 010:225mV 110:175mV 111:162.5mV
14:13	RW	0x0	voltage_tolerance_adjust 5V tolerance detection reference adjust, 11 represent the highest trigger point, keeping the default value is greatly appreciated
12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:10	RW	0x0	auto_compensation_bypass auto compensation bypass , "11" will bypass current and ODT compensation, customers can set the driver strength and current manually. For larger HS eye height, customer can give more "0" for hs_eye_height; For larger HS/FS/LS slew rate , give more "1" for hfs_driver_strength.
9:5	RW	0x15	hfs_driver_strength HS/FS driver strength tuning , "11111" represent the largest slew rate and "10000" represents the smallest slew rate
4:0	RW	0x0a	hs_eye_height bit7 ~ bit 3 of hs_eye_height. HS eye height tuning ,more zeros represent bigger eye, more ones represent smaller eye

GRF_USBPHY0_CON3

Address: Operational Base + offset (0x028c)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	vol_toleran_det_con 5V tolerance detection function controlling bit through registers, only active when bit[65] is set "1".
13:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RW	0x1	odt_auto_refresh A port ODT auto refresh bypass, active low, this register should only be used when auto_compensation_bypass were set to "11". In bypass mode , customer can configure driver strength through hfs_driver_strength.
8	RW	0x0	bg_out_voltage_adjust BG output voltage reference adjust, keeping the default value is greatly appreciated.
7:5	RW	0x0	compen_current_ref compensation current tuning reference 000:200mV(default) 001:187.5mV 010:225mV 110:175mV 111:162.5mV
4:2	RW	0x0	bias_current_ref bias current tuning reference 000:400mV(default) 001:362.5mV 010:350mV 101:425mV 111:450mV
1:0	RW	0x0	odt_compensation bit 2 ~ bit 1 of odt_compensation. ODT Compensation voltage reference 000:200mV 001:187.5mV(default) 010:225mV 110:175mV 111:162.5mV

GRF_USBPHY0_CON4

Address: Operational Base + offset (0x0290)

usbphy control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:2	RO	0x0	reserved
1	RW	0x1	bypass_5v_tolerance_det Bypass 5V tolerance detection function, active high
0	RO	0x0	reserved

GRF_USBPHY0_CON5

Address: Operational Base + offset (0x0294)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0	reserved

GRF_USBPHY0_CON6

Address: Operational Base + offset (0x0298)

usbphy control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RW	0x0	session_end_con session_end reference tuning
12:10	RW	0x0	b_session_con B_sessionvalid reference tuning
9:7	RW	0x0	a_session_con A_sessionvalid reference tuning
6	RW	0x0	force_vbus_valid force output vbus_valid asserted, active high
5	RW	0x0	force_session_end_val force output session_end asserted, active high
4	RW	0x0	force_b_session_val force output B_sessionvalid asserted, active high
3	RW	0x0	force_a_session_val force output A_sessionvalid asserted, active high
2	RW	0x1	turn_off_diff_receiver Turn off differential receiver in suspend mode to save power, active low.
1:0	RO	0x0	reserved

GRF_USBPHY0_CON7

Address: Operational Base + offset (0x029c)
 usbphy control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15	RO	0x0	reserved
14:11	RW	0xd	<p>host_discon_con</p> <p>HOST disconnect detection trigger point configure, only used in HOST mode</p> <p>0000: 575mV</p> <p>0001: 600mV</p> <p>1001:625mV</p> <p>1101:650mV(default)</p>
10:8	RO	0x0	reserved
7	RW	0x1	<p>bypass_squelch_trigger</p> <p>bypass squelch trigger point auto configure in chirp modes , active high</p>
6	RW	0x1	<p>half_bit_pre_empha_en</p> <p>half bit pre-emphasize enable, active high.</p> <p>"1" represent half bit pre-emphasis , "0" for full bit</p>
5:3	RO	0x0	reserved
2:0	RW	0x0	<p>vbus_valid_con</p> <p>vbus_valid reference tuning</p>

GRF_USBPHY1_CON0

Address: Operational Base + offset (0x02a0)
 usbphy control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable</p> <p>bit0~bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software .</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software .</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software .</p> <p>When bit 31=0, bit 15 cannot be written by software;</p>
15:13	RW	0x4	<p>squel_trigger_con</p> <p>bit 2 ~ bit 0 of squel_trigger_con.</p> <p>0000:112.5mV</p> <p>1001:162.5mV</p> <p>1011:175mV</p> <p>1100:150mV(default)</p> <p>1110:125mV</p>
12:11	RW	0x0	<p>non_driving</p> <p>Registers for non-driving state control.</p> <p>non-driving state is controlled by op-mode by default, when bit[11] is configured with "1", user can control non-driving state through bit[12].</p>
10:8	RW	0x6	<p>tx_clk_phase_con</p> <p>USB Tx Clock phase configure, 000 represent the earliest phase , and 111 the latest, single step delay is 256ps</p>
7:5	RW	0x0	<p>rx_clk_phase_con</p> <p>USB Rx Clock phase configure,000 represent the earliest phase , and 111 the latest, single step delay is 256ps</p>
4:3	RW	0x3	<p>fls_eye_height</p> <p>FS/LS eye height configure , 00 represent the largest slew rate , 11 represent the smallest slew rate</p>
2:0	RW	0x0	<p>hs_eye_diag_adjust</p> <p>HS eye diagram adjust, open HS pre-emphasize function to increase HS slew rate, only used when large cap loading is attached.</p> <p>001:open pre-emphasize in sof or eop state</p> <p>010: open pre-emphasize in chirp state</p> <p>100: open pre-emphasize in non-chirp state</p> <p>111: always open pre-emphasize other combinations : reserved</p>

GRF_USBPHY1_CON1

Address: Operational Base + offset (0x02a4)
 usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RW	0x7	hs_eye_height bit2 ~ bit 0 of hs_eye_height. HS eye height tuning ,more zeros represent bigger eye, more ones represent smaller eye
12:3	RO	0x0	reserved
2	RW	0x1	current_comp_en Enable current compensation, active high.
1	RW	0x1	res_comp_en Enable resistance compensation, active high.
0	RW	0x1	squel_trigger_con bit 3 of squel_trigger_con. 0000:112.5mV 1001:162.5mV 1011:175mV 1100:150mV(default) 1110:125mV

GRF_USBPHY1_CON2

Address: Operational Base + offset (0x02a8)
 usbphy control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;</p>
15	RW	0x1	<p>odt_compensation bit 0 of odt_compensation. ODT Compensation voltage reference 000:200mV 001:187.5mV(default) 010:225mV 110:175mV 111:162.5mV</p>
14:13	RW	0x0	<p>voltage_tolerance_adjust 5V tolerance detection reference adjust, 11 represent the highest trigger point, keeping the default value is greatly appreciated</p>
12	RO	0x0	reserved
11:10	RW	0x0	<p>auto_compensation_bypass auto compensation bypass , "11" will bypass current and ODT compensation, customers can set the driver strength and current manually. For larger HS eye height, customer can give more "0" for hs_eye_height; For larger HS/FS/LS slew rate , give more "1" for hfs_driver_strength.</p>
9:5	RW	0x15	<p>hfs_driver_strength HS/FS driver strength tuning , "11111" represent the largest slew rate and "10000" represents the smallest slew rate</p>
4:0	RW	0x0a	<p>hs_eye_height bit7 ~ bit 3 of hs_eye_height. HS eye height tuning ,more zeros represent bigger eye, more ones represent smaller eye</p>

GRF_USBPHY1_CON3

Address: Operational Base + offset (0x02ac)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	vol_toleran_det_con 5V tolerance detection function controlling bit through registers, only active when bit[65] is set "1".
13:10	RO	0x0	reserved
9	RW	0x1	odt_auto_refresh A port ODT auto refresh bypass, active low, this register should only be used when auto_compensation_bypass were set to "11". In bypass mode , customer can configure driver strength through hfs_driver_strength.
8	RW	0x0	bg_out_voltage_adjust BG output voltage reference adjust, keeping the default value is greatly appreciated.
7:5	RW	0x0	compen_current_ref compensation current tuning reference 000:200mV(default) 001:187.5mV 010:225mV 110:175mV 111:162.5mV
4:2	RW	0x0	bias_current_ref bias current tuning reference 000:400mV(default) 001:362.5mV 010:350mV 101:425mV 111:450mV

Bit	Attr	Reset Value	Description
1:0	RW	0x0	odt_compensation bit 2 ~ bit 1 of odt_compensation. ODT Compensation voltage reference 000:200mV 001:187.5mV(default) 010:225mV 110:175mV 111:162.5mV

GRF_USBPHY1_CON4

Address: Operational Base + offset (0x02b0)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:2	RO	0x0	reserved
1	RW	0x1	bypass_5v_tolerance_det Bypass 5V tolerance detection function, active high
0	RO	0x0	reserved

GRF_USBPHY1_CON5

Address: Operational Base + offset (0x02b4)

usbphy control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0	reserved

GRF_USBPHY1_CON6

Address: Operational Base + offset (0x02b8)
 usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RW	0x0	session_end_con session_end reference tuning
12:10	RW	0x0	b_session_con B_sessionvalid reference tuning
9:7	RW	0x0	a_session_con A_sessionvalid reference tuning
6	RW	0x0	force_vbus_valid force output vbus_valid asserted, active high
5	RW	0x0	force_session_end_val force output session_end asserted, active high
4	RW	0x0	force_b_session_val force output B_sessionvalid asserted, active high

Bit	Attr	Reset Value	Description
3	RW	0x0	force_a_session_val force output A_sessionvalid asserted, active high
2	RW	0x1	turn_off_diff_receiver Turn off differential receiver in suspend mode to save power, active low.
1:0	RO	0x0	reserved

GRF_USBPHY1_CON7

Address: Operational Base + offset (0x02bc)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14:11	RW	0xd	host_discon_con HOST disconnect detection trigger point configure, only used in HOST mode 0000: 575mV 0001: 600mV 1001:625mV 1101:650mV(default)
10:8	RO	0x0	reserved
7	RW	0x1	bypass_squelch_trigger bypass squelch trigger point auto configure in chirp modes , active high
6	RW	0x1	half_bit_pre_empha_en half bit pre-emphasize enable, active high. "1" represent half bit pre-emphasis , "0" for full bit
5:3	RO	0x0	reserved
2:0	RW	0x0	vbus_valid_con vbus_valid reference tuning

GRF_UOC_STATUS0

Address: Operational Base + offset (0x02c0)

SoC status register 0

Bit	Attr	Reset Value	Description
31:26	RW	0x00	usbhost0_stat_ehci_usbsts USB host0 ehci_usbsts bit status
25:15	RW	0x000	usbhost0_stat_ehci_xfer_cnt USB host0 ehci_xfer counter status
14	RW	0x0	usbhost0_stat_ehci_xfer_prdc USB host0 ehci_xfer_prdc bit status
13:10	RW	0x0	usbhost0_stat_ehci_lpsmc_state USB host0 ehci_lpsmc_state bit status
9	RW	0x0	usbhost0_stat_ehci_bufacc USB host0 ehci_bufacc bit status
8	RW	0x0	usbhost0_stat_ohci_globalsuspend USB host0 ohci_globalsuspend bit status
7	RW	0x0	Copy1 usbhost0_stat_dp_attached USB HOST0 dp_attached signal status
6	RW	0x0	usbhost0_stat_cp_detected USB HOST0 cp_detected signal status
5	RW	0x0	usbhost0_stat_dcp_attached USB HOST0 dcp_attached signal status
4	RW	0x0	usbhost0_stat_ohci_bufacc USB HOST0 ohci_bufacc signal status
3	RW	0x0	usbhost0_stat_ohci_rmtwkp USB HOST0 ohci_rmtwkp signal status
2	RW	0x0	usbhost0_stat_ohci_drwe USB HOST0 ohci_drwe signal status
1	RW	0x0	usbhost0_stat_ohci_rwe USB HOST0 ohci_rwe signal status
0	RW	0x0	usbhost0_stat_ohci_ccs USB HOST0 ohci_ccs signal status

GRF_CHIP_TAG

Address: Operational Base + offset (0x0300)

chip tag register

Bit	Attr	Reset Value	Description
31:0	RO	0x0000293c	chip_tag

GRF_MMC_DET_CNT

Address: Operational Base + offset (0x0304)

mmc0 detect filter counter register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RO	0x0fdb9	mmc_det_value mmc0 detect filter counter initial value

GRF_EFUSE_PRG_EN

Address: Operational Base + offset (0x037c)

efuse program register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RO	0x0	efuse_prg_en 0: efuse program disable 1: efuse program enable
12:0	RO	0x0	reserved

Chapter 6 Embedded Processor (Cortex-A7)

6.1 Overview

The Cortex-A7 MP subsystem of the device is based on the symmetricmultiprocessor (SMP) architecture, thus the quad Cortex-A7 MPU subsystem delivers higher performanceand optimal power management, debug and emulation capabilities.

The Cortex-A7 MP subsystem incorporates four Cortex-A7 central processing units (CPUs), level 2(L2) cache shared between the four CPUs, and uses PL310 as L2 cache controller. Each CPU has 32KBof level 1 (L1) instruction cache, 32KB of L1 data cache, separate dedicated power domain, and includesone Neon and Vector Floating Point Unit coprocessors. The Cortex-A7 MP subsystemalso includes standard CoreSight components to support SMP debug and emulation, snoop control unit(SCU), interrupt controller (GIC), and clock and reset manager.

The key features of the Cortex-A7 MP subsystem include:

- ARM Coretex-A7 based quad MPU subsystem with SMP architecture
 - Cortex-A7 core revision r0p5
 - Full implements the ARMv7-A architecture profile that includes SIMD and VFP
 - 32KB L1 I-cache and 32KB L1 D-cacheper CPU
 - In-order pipeline with direct and indirect branch prediction
 - Harvard Level 1 (L1) memory system with a Memory Management Unit (MMU)
 - SCU ensures memory coherency between the 4 CPUs
 - Interrupt controller with 128 hardware interrupt inputs
- 256KB L2 cache shared between the 4 CPUs
 - Fixed line length of 64 bytes.
 - Physically indexed and tagged cache.
 - 8-way set-associative cache structure.
 - Pseudo-random cache replacement policy.

6.2 Block Diagram

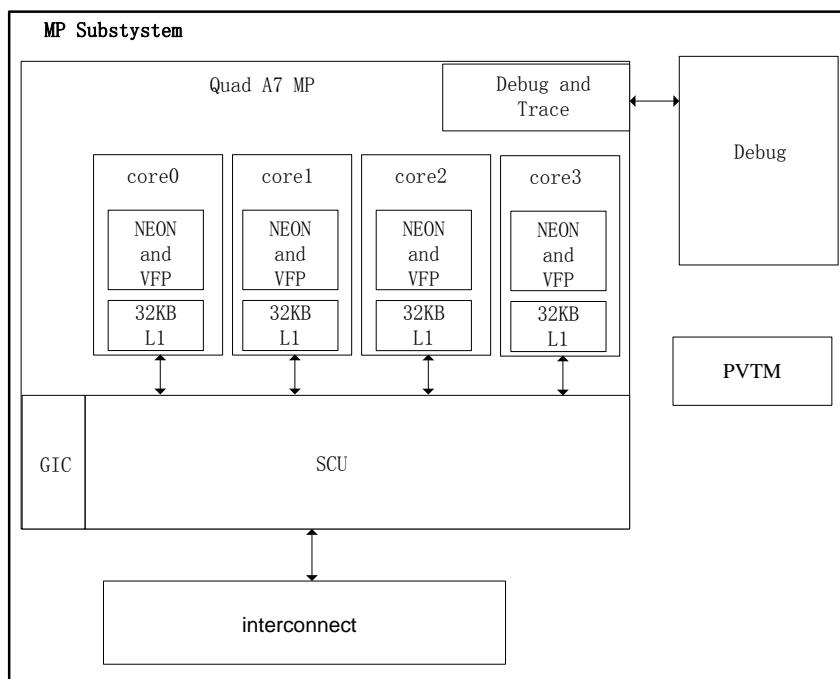


Fig.6-1MP Subsystem architecture

6.3 Function description

Please refer to the document Cortex-A7_MPCore_Technical_Reference_Manual.pdf for the cpu detail description.

6.4 Register description

Please refer to the document Cortex-A7_MPCore_Technical_Reference_Manual.pdf for the cpu detail description.

Chapter 7 Interconnect

7.1 Overview

The chip-level interconnect consists of one cpu_sys interconnect and peri_sys interconnects. It enables communication among the modules and subsystems in the device.

The cpu_sys interconnect handles many types of data transfers, especially exchanges with system-on-chip (SoC)/external memories. It transfers data with a maximum width of 128 bits from the initiator to the target. It is a little-endian platform.

The peri_sys interconnect belongs to peripheral system which is responsible for peripheral devices control such as USB device, flash device, UART, SPI etc.

7.2 Axi interconnect

The interconnect comprises with:

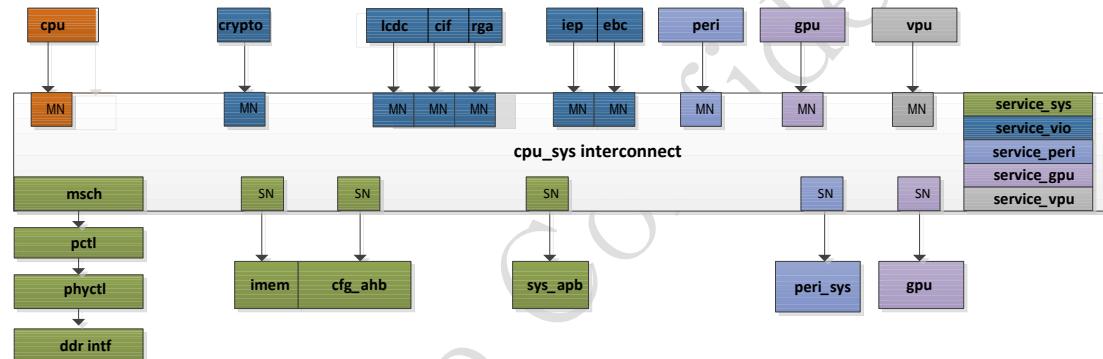


Fig.7-1 Block Diagram

7.2.1 Integration description

The main interconnect is connected with all the related IPs of the system, the interface between the IP and the interconnect is called as NIU(native interface unit). All the connected NIU are listed as below:

Table 7-1 Master NIU

Master NIU	Description
MN_cpu	Master interface of pd_core system
MN_crypto	Master interface of crypto in pd_cpu system
MN_lcdc0	Master interface of LCDC0 in pd_vio system
MN_cif	Master interface of CIF in pd_vio system
MN_rga	Master interface of RGA in pd_vio system
MN_iep	Master interface of IEP in pd_vio system

MN_ebc	Master interface of EBC in pd_vio system
MN_peri	Master interface of pd_peri system
MN_gpu	Master interface of GPU in pd_gpu system
MN_vpu	Master interface of Video codec in pd_vcodec system

Table 7-2 slave NIU

Slave NIU	Description
SN_msch	Memory scheduler of pd_cpu system
SN_imem	Internal memory of pd_cpu system
SN_cfg_ahb	AHB interface for ahb slave of pd_cpu system
SN_sys_apb	APB slave interface of pd_cpu system
SN_peri_sys	AXI slave interface of pd_peri system
SN_gpu_slv	GPU slave interface of pd_gpu system

The cpu_sysinterconnect is divided into six clock domain. Each clock domain has its own elements.

Following table shows the elements in each clock domain.

Table 7-3 Clock and Power domain

Clock domain	Elements
sys_clk_dm	Cpu
	Crypto
	Service_sys
	Imem
	Cfg_ahb
	System_apb
vio_clk_dm	Lcdc0
	Cif
	IEP

	Rga
	Ebc
	Service_vio
gpu_clk_dm	Gpu
	Gpu_slv
	Service_gpu
vpu_clk_dm	VPU
	Service_vpu
peri_clk_dm	Peri
	Peri_sys
	Service_peri
msch_clk_dm	Msch

7.2.2 Function description

Following figure lists the functional paths between the cpu_sys interconnect master NIUs and the slaveNIUs. The functional paths in the figure are indicated by the following:

- A cell contains a plus sign when a functional path exists
- A cell is blank when a functional path does not exist

Master	Slave										
	gpu_apb_slv	msch	peri_sys	sys_apb	imem	cfg_ahb	service_sys	service_peri	service_vio	service_gpu	service_vpu
CPU0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CRYPTO		✓			✓						
DMAC		✓	✓								
GPU		✓									
LCDC0		✓									
IEP		✓									
RGA		✓									
CIF		✓									
EBC		✓									
VPU		✓									
PERI		✓									

Fig.7-2 Connectivity

The cpu_sys interconnect embedded a Memory Scheduler NIU which converts multiple transport operations from master NIU to a generic socket master intended for a DRAM controller front-end. The NIU enables maximum DRAM throughput for concurrent data flows from these ports. Because the NIU supports context management and read data buffering, it provides seamless and cost-effective connection to NIF memory controller protocols(PCTL).

7.2.3 Memory Scheduler

Memory scheduler is a special NIU of the interconnect, it mainly deal with the transaction inside the interconnect and convert it to the transaction which the ddrprotocol controller can recognize.

There are two memory schedulers in the interconnect, each is in different clock domain. These two memory schedulers are totally equal in function .

Following table shows the software configurable setting for the memory scheduler when the system connected to different size of ddr device.

The DDRCONF[3:0] is a configurable register inside the interconnect.

R: indicates Row bits

B: indicates Bank bits

C: indicates Column bits

D: indicates Chip selects bits

Table 7-4 DDR configuration

DDR CONF[3:0]	
0	C RRRD RRRR RRRRRRRR RBBC CCCC CCCCCCCC
1	C RRDR RRRR RRRRRRRR RBBC CCCC CCCCCCCC
2	C RDRR RRRR RRRRRRRR RBBC CCCC CCCCCCCC
3	C DRRR RRRR RRRRRRRR RBBC CCCC CCCCCCCC
4	D RRRR RRRRRRRRRR BBBC CCCC CCCCCCCC
5	C CRRR DRRR RRRR RRRR RRBB BCCC CCCC CCCC
6	C CRRD RRRR RRRRRRRR RRBB BCCC CCCC CCCC
7	C CRDR RRRR RRRRRRRR RRBB BCCC CCCC CCCC
8	C CDRR RRRR RRRRRRRR RRBB BCCC CCCC CCCC
9	C DRRR RRRR RRRRRRRR RBBC CCCC CCCCCCCC
10	C RRDB BBRR RRRR RRRRRRRR CCCC CCCCCCCC
11	C CRDB BBRR RRRR RRRRRRRR RCCC CCCC CCCC
12	C RRRR RRRRRRRRRR DBBB CCCC CCCCCCCC
13	C RRRR RRRRRRRRRR DBBB CCCC CCCCCCCC
14	C RRRR RRRRRRRRRR RDBB BCCC CCCC CCCC
15	C RRRR RRRRRRRRRR RDBB BCCC CCCC CCCC

7.2.4 Register Description(main interconnect)

Registers Summary

The register in the cpu axi interconnect are composed by seven groups. Each group composed by some sub-groups.

Main-group	Sub-group	Sub-Offset address	Main-Base address
service_sys	msch_cfg	0x0	0x10128000
	crypto_qos	0x80	

service_vio	rga_qos	0x0	0x1012f000
	ebc_qos	0x80	
	iep_qos	0x100	
	lc当地_0_qos	0x180	
	vip0_qos	0x200	
service_peri	Peri_qos	0x0	0x1012c000
service_gpu	Gpu_qos	0x0	0x1012d000
service_vpu	Vpu_qos	0x0	0x1012e000
service_core	cpu0_qos	0x0	0x1012a000

memory scheduler (msch_cfg)register summary

Name	Offset	Size	Reset Value	Description
coreid	0x0000	W	0x21501602	core id of memory scheduler 0
revisionid	0x0004	W	0x0126f200	revisionid
ddrconf	0x0008	W	0x00000000	ddr configuration register
ddrtiming	0x000c	W	0x2475931c	ddr timing register
ddrmode	0x0010	W	0x00000000	ddr mode register
readlatency	0x0014	W	0x00000032	read latency register

service_core: cpup_qosregister summary

Name	Offset	Size	Reset Value	Description
priority	0x0008	W	0x00000101	priority register
mode	0x000c	W	0x00000003	qos mode register
bandwidth	0x0010	W	0x00000005	qos bandwidth register
saturation	0x0014	W	0x00000040	qos saturation register

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

other master's qos register is same with service_core

2. Detail Register Description

CoreId

Address: Operational Base + offset (0x0000)

Core ID register

Bit	Attr	Reset Value	Description
31:8	RW	0x000000	CoreChecksum A global checksum of all hardware parameters
7:0	RW	0x00	CoreTypeId 0 : rate adapter 1 : observer 2 : memory scheduler

RevisionId

Address: Operational Base + offset (0x0004)

Revision ID register

Bit	Attr	Reset Value	Description
31:8	RW	0x000000	ToolId identifier of the version
7:0	RW	0x00	UserId user-defined identifier

DdrConf

Address: Operational Base + offset (0x0008)

Memory scheduler configuration register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	DdrConf select the ddrrank,row,bank,col sequence

DdrTiming

Address: Operational Base + offset (0x000c)

Memory scheduler timing register

Bit	Attr	Reset Value	Description
31	RW	0x0	BwRatio Bandwidth ratio
30:26	RW	0x00	WrToRd Minimum time between the last DRAM Write command and a Read command. It is equal to: $(WL * tCkD) + tWTR$
25:21	RW	0x00	RdToWr Minimum time between the last DRAM Read command and a Write command. For DDR3, it is equal to $(RL - WL + 2) * tCkD$.
20:18	RW	0x0	BurstLen DRAM burst duration on the DRAM data bus. Also equal to minimum time between two DRAM commands. It is equal to: $BL/2 * tCkD$
17:12	RW	0x00	WrToMiss Minimum time between the last DRAM Write command and a new Read or Write command in another page of the same bank. It is equal to: $(WL * tCkD) + tWR + tRP + tRCD$

Bit	Attr	Reset Value	Description
11:6	RW	0x00	RdToMiss Minimum time between the last DRAM Read command and a new Read or Write command in another page of the same bank. The interval of time is equal to: tRTP + tRP + tRCD - (BL/2*tCkD)
5:0	RW	0x00	ActToAct Minimum time between two consecutive DRAM Activate commands on the same bank.

DdrMode

Address: Operational Base + offset (0x0010)

Memory scheduler mode register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	AutoPrecharge

ReadLatency

Address: Operational Base + offset (0x0014)

Memory scheduler read latency register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	ReadLatency read latency

Priority

Address: Operational Base + offset (0x0008)

CPU master0 priority register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:2	RW	0x2	P1 sets the high hurry level (i.e. when the measured bandwidth does not exceed the setting) when in Regulator mode, or read urgency level when in fixed or limiter mode.
1:0	RW	0x0	P0 sets the low hurry level, that is, when the measured bandwidth exceeds the setting, when in regulator mode, or write urgency level when in fixed or limiter mode.

CoreSrv_Mode

Address: Operational Base + offset (0x000c)

CPU master0 QoS mode register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x3	Mode determines which of the following modes the QoSGenerator will adopt at reset: 0: None 1:Fixed 2:Limiter 3:Regulator

CoreSrv_Bandwidth

Address: Operational Base + offset (0x0010)

CPU master0 QoSbandwidth register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:0	RW	0x007	<p>Bandwidth</p> <p>Parameter bandwidth determines the bandwidth triggering of the limiter or the regulator. It is expressed in bytes per second. This parameter becomes available when limiter or regulator hardware is implemented.</p>

CoreSrv_Saturation

Address: Operational Base + offset (0x0014)

CPU master0 QoS saturation register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x3ff	<p>Saturation</p> <p>Parameter saturation determines the excursion of the payload counter, used to estimate the bandwidth, expressed in bytes. This parameter becomes available when limiter or regulator hardware is implemented.</p>

7.3 Peri AXI interconnect

7.3.1 Integration description

PERI axi interconnect is a 64bits data width axi interconnect which mainly respond to peripheral devices' data transaction.

This axi interconnect has the following master and slave:

MASTER

1. DMAC

The 64bits axi master, mainly respond to data transfer between peripheral and DDR SDRAM.

2. AHB_MASTER

GPS/NANDC/USB OTG/USB Host2.0/TSP/SFC/GMAC, these masters are used to transfer data between DDR SDRAM and each of their own module.

SLAVE

1.AHB_SLAVE

All ahb slave devices in peri system for register access, also for data transfer such as SD/MMC,SDIO,eMMC, GPS,TSP, SFC, I2S_2ch,I2S_8ch, SPDIF, USB OTG,USB HOST2.0 and NandC.

2. APB_SLAVE

DMAC/PWM/WDT/GPIO/UART/I2C/SPI/TIMER/EFUSE/SARADC/SCR/GMAC, these apb slaves are used to register configuration.

7.3.2 Function description

A Global Programmers View(GPV) module exists to configure some properties of the interconnect. The arbitration scheme of this interconnect is configurable through GPV.

The priority from high to low is as follow:

- CPU AXI interconnect
- DMAC
- USB OTG2.0/USB Host2.0
- PD8_AHBM/ GPS_AHBM

Customers can configure the Qos value through the GPV to change this priority. If you config them to same priority,then the interconnect uses a Least Recently Used (LRU) algorithm

7.3.3 Register description

1. Registers Summary

Name	Offset	Size	Reset Value	Description
PERI_RQos_M0	0x42100	4bits	0x0003	PERI_AXI Port Read channel QoS value.
PERI_WQos_M0	0x42104	4bits	0x0003	PERI_AXI Port Write channel quality value.
PERI_RQos_M1	0x43100	4bits	0x0002	DMAC1 Port Read channel QoS value.
PERI_WQos_M1	0x43104	4bits	0x0002	DMAC1 Port Write channel quality value.
PERI_RQos_M2	0x44100	4bits	0x0001	USBM Port Read channel QoS value.
PERI_WQos_M2	0x44104	4bits	0x0001	USBM Port Write channel quality value.

PERI_RQos_M3	0x45100	4bits	0x0000	PD8_AHB Port Read channel QoS value.
PERI_WQos_M3	0x45104	4bits	0x0000	PD8_AHB Port Write channel quality value.
PERI_RQos_M4	0x46100	4bits	0x0000	GPS_AHB Port Read channel QoS value.
PERI_WQos_M4	0x46104	4bits	0x0000	GPS_AHB Port Write channel quality value.

Notes:

Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** – WORD (32 bits) access

2. Detail Register Description

PERI_RQos_M0

Address: Operational Base+0x42100

PERI_AXI Port Read Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x1	Read channel QoSvalue.Higher value indicates higher priority.

PERI_WQos_M0

Address: Operational Base+0x42104

PERI_AXI Port Write Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x1	Write channel QoSvalue.Higher value indicates higher priority.

PERI_RQos_M1

Address: Operational Base+0x43100

DMAC1 Port Read Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Read channel QoSvalue.Higher value indicates higher priority.

PERI_WQos_M1

Address: Operational Base+0x43104

DMAC1 Port Write Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Write channel QoS value. Higher value indicates higher priority.

PERI_RQos_M2

Address: Operational Base+0x44100

USBM Port Read Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Read channel QoS value. Higher value indicates higher priority.

PERI_WQos_M2

Address: Operational Base+0x44104

USBM Port Write Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Write channel QoS value. Higher value indicates higher priority.

PERI_RQos_M3

Address: Operational Base +0x45100

PD8_AHBM Port Read Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Read channel QoS value. Higher value indicates higher priority.

PERI_WQos_M3

Address: Operational Base +0x45104

PD8_AHBM Port Write Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Write channel QoS value. Higher value indicates higher priority.

PERI_RQos_M4

Address: Operational Base +0x46100

PD8_AHBM Port Read Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Read channel QoS value. Higher value indicates higher priority.

PERI_WQos_M4

Address: Operational Base +0x46104

PD8_AHBM Port Write Qos register

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Write channel QoS value. Higher value indicates higher priority.

7.4 Application Notes

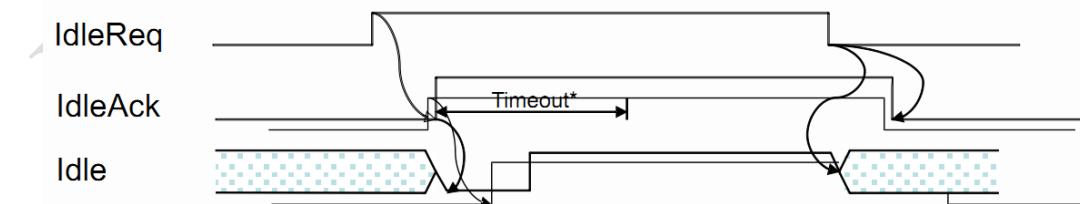
7.4.1 Idle request to flush the transaction flow

The cpu_sys interconnect supports flushing the ongoing transaction when the software needed to do so.

If the GPU power domain need to disconnect from the cpu_sys interconnect, Idle request has to be sent to GPU NIU, the NIU will respond a ack, and when it's ready to be disconnect, one Idle signal will be send out . Then, if GPU still have transaction to be sent to cpu_sys, it will be stalled by the NIU.

If the CPU system power domain is disconnected as the above flow, then GPU want to access to the CPU system, it will response error to GPU module.

The sequence is like following figure shows:



The idle request is set by GRF register.

7.4.2 QoS and more

This device, the QoS information is generated inside the cpu_sys interconnect. This is done through the QoS generator.

The QoSgenerator offers 4 modes:

- None:QoSGenerator is disabled, and priority information are stuck at 0.
- Fixed:QoSGenerator drives applies a fixed urgency to read transactions, and a (possibly different) urgency to write transactions.
- Limiter:QoSGenerator behaves as in fixed mode, but limits the traffic bandwidth coming from that socket, possibly stalling requests if the initiator attempts to exceed its budget.
- Regulator:QoSGenerator promotes are demotes hurry, depending the bandwidth obtained by the initiator is below or beyond a bandwidth budget. As transactions exceeding the bandwidth limit are sent (even though demoted), the regulator mode may be considered as a softer version of the limiter mode.

Limiter Behavior

When configured in bandwidth limiter, the unit uses a 23 bit counter to measure the average bandwidth. This counter has a 1/256 byte resolution and works as follows:

- Adds the number of byte rounded up to 16 (1 -> 16) and then multiplied by 256 to the current value, each time a request is sent.
- Subtracts the Bandwidth register value every cycle. If the Counter becomes negative, force it to 0.
 - If the Counter value is greater than the Saturation register value multiplied by 16*256, any incoming request is stalled until this condition disappears. Note that the Counter cannot wrap-around because the maximum value it can reach is: $\text{SaturationMax} \times 16 \times 256 + \text{BurstMax} \times 256 = 1023 \times 4K + 4K \times 256 = 5116K$ or $223 = 8192K$.

The following example will show the Counter behavior: 32 byte bursts, F=400MHz, BW=200MB/s, T=0.32us. The Bandwidth register will be set to $256 \times 200 / 400 = 128$, and the Saturation register to $128 \times 0.32 \times 400 / 4096 = 4$ (which corresponds to 64 bytes).

Regulator Behavior

When configured in bandwidth regulator, the unit uses a 23 bit counter to measure the average bandwidth. This counter has a 1/256 byte resolution and works as follows:

- Adds the number of byte rounded up to 16 and then multiplied by 256 to the current value, each time a response is received. If the result is greater than the Saturation register value multiplied by 16*256, saturation to this value is applied.
- Subtracts the Bandwidth register value every cycle. If the Counter becomes negative, force it to 0.
 - If the Counter value is less than or equal to the Saturation register value multiplied by $16 \times 256 / 2$, the SocketMst Hurry signal will be set to the HurryHigh register, and HurryLow otherwise. Note that Urgency and Press will be also set to the same value.

The following example will show the Counter behavior: 1Kbyte bursts,

F=500MHz, BW=2GB/s, T=2.048us. The Bandwidth register will be set to $256*2000/500 = 1024$, and the Saturation register to $1024*2.048*500/4096 = 256$ (which corresponds to 4 Kbytes).

QoS Generator Programming

Bandwidth: This $\log_2(\text{socket.wData}/8) + 8$ bits register defines the bandwidth in 1/256th byte per cycle unit. This allows a 2 MByte/s resolution at 500MHz. When the bandwidth is given in MByte/s, the value of this register will be equal to $256 * \text{BWMB/s} / \text{FMHz}$.

Saturation: This 10 bits register defines the number of bytes used for bandwidth measurement. It is expressed in 16bytes unit (up to 16 Kbyte). Usually the integration window is given in us or in cycle: the value of this register will be equal to $\text{Bandwidth} * \text{Tus} * \text{FMHz} / (256 * 16)$ or $\text{Bandwidth} * \text{Ncycle} / (256 * 16)$.

Chapter 8 DMA Controller (DMAC)

8.1 Overview

DMAC does not support TrustZone technology and work under non-secure state only.

DMAC is mainly used for data transfer of the following slaves: I2S0, SD/MMC, SDIO, eMMC, UART0, UART1, UART2, SPI0.

Following table shows the DMAC request mapping scheme.

Table 8-1DMAC Request Mapping Table

Req number	Source	Polarity
0	I2S_2ch_tx	High level
1	I2S_2ch_rx	High level
2	UART0 tx	High level
3	UART0 rx	High level
4	UART1 tx	High level
5	UART1 rx	High level
6	UART2 tx	High level
7	UART2 rx	High level
8	SPI tx	High level
9	SPI rx	High level
10	SD/MMC	High level
11	SDIO	High level
12	eMMC	High level
13	SPDIF	High level
14	I2S_8ch_tx	High level
15	I2S_8ch_rx	High level

DMAC supports the following features:

- Supports 15 peripheral request.
- Up to 64bits data size.
- 8 channel at the same time.
- Up to burst 16.
- 1 interrupts output and 1 abort output.
- Supports 32 MFIFO depth.

8.2 Block Diagram

Figure 8-1 shows the block diagram of DMAC.

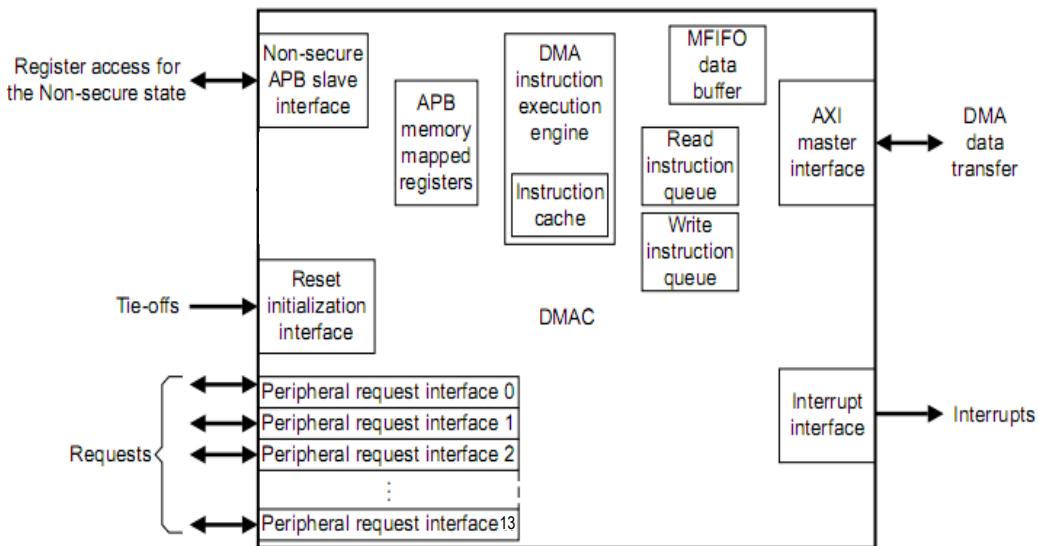


Fig. 8-1 Block diagram of dmac

8.3 Function Description

8.3.1 Introduction

The DMAC contains an instruction processing block that enables it to process program code that controls a DMA transfer. The program code is stored in a region of system memory that the DMAC accesses using its AXI interface. The DMAC stores instructions temporarily in a cache.

DMAC supports 8 channels, each channel capable of supporting a single concurrent thread of DMA operation. In addition, a single DMA manager thread exists, and you can use it to initialize the DMA channel threads. The DMAC executes up to one instruction for each AXI clock cycle. To ensure that it regularly executes each active thread, it alternates by processing the DMA manager thread and then a DMA channel thread. It uses a round-robin process when selecting the next active DMA channel thread to execute.

The DMAC uses variable-length instructions that consist of one to six bytes. It provides a separate Program Counter (PC) register for each DMA channel. When a thread requests an instruction from an address, the cache performs a look-up. If a cache hit occurs, then the cache immediately provides the data. Otherwise, the thread is stalled while the DMAC uses the AXI interface to perform a cache line fill. If an instruction is greater than 4 bytes, or spans the end of a cache line, the DMAC performs multiple cache accesses to fetch the instruction.

When a cache line fill is in progress, the DMAC enables other threads to access the cache, but if another cache miss occurs, this stalls the pipeline until the first line fill is complete.

When a DMA channel thread executes a load or store instruction, the DMAC adds the instruction to the relevant read or write queue. The DMAC uses these queues as an instruction storage buffer prior to it issuing the instructions on the AXI bus. The DMAC also contains a Multi First-In-First-Out (MFIFO) data buffer that it uses to store data that it reads, or writes, during a DMA transfer.

8.3.2 Operating states

Figure shows the operating states for the DMA manager thread and DMA channel threads.

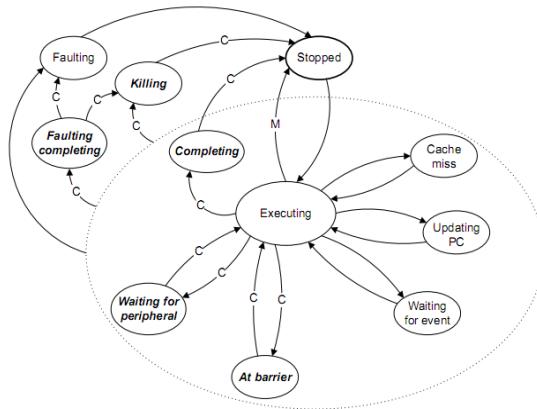


Fig. 8-2 DMAC operation states

Note:

arcs with no letter designator indicate state transitions for the DMA manager and DMA channel threads, otherwise use is restricted as follows:

C DMA channel threads only.

M DMA manager thread only.

After the DMAC exits from reset, it sets all DMA channel threads to the stopped state, and the status of boot_from_pc(tie-off interface of dmac) controls the DMA manager thread state:

boot_from_pc is LOW :DMA manager thread moves to the Stopped state.
boot_from_pc is HIGH :DMA manager thread moves to the Executing state.

8.4 Register Description

8.4.1 Register summary

Name	Offset	Size	Reset Value	Description
DMAC_DSR	0x0000	W	0x0	DMA Status Register.
DMAC_DPC	0x0004	W	0x0	DMA Program Counter Register.
-	-	-	-	reserved
DMAC_INTEN	0x0020	W	0x0	Interrupt Enable Register
DMAC_EVENT_RIS	0x0024	W	0x0	Event Status Register.
DMAC_INTMIS	0x0028	W	0x0	Interrupt Status Register
DMAC_INTCLR	0x002C	W	0x0	Interrupt Clear Register
DMAC_FSRD	0x0030	W	0x0	Fault Status DMA Manager Register.
DMAC_FSRC	0x0034	W	0x0	Fault Status DMA Channel Register.
DMAC_FTRD	0x0038	W	0x0	Fault Type DMA Manager Register.
-	-	-	-	reserved

DMAC_FTR0	0x0040	W	0x0	Fault type for DMA Channel 0
DMAC_FTR1	0x0044	W	0x0	Fault type for DMA Channel 1
DMAC_FTR2	0x0048	W	0x0	Fault type for DMA Channel 2
DMAC_FTR3	0x004C	W	0x0	Fault type for DMA Channel 3
DMAC_FTR4	0x0050	W	0x0	Fault type for DMA Channel 4
DMAC_FTR5	0x0054	W	0x0	Fault type for DMA Channel 5
DMAC_FTR6	0x0058	W	0x0	Fault type for DMA Channel 6
DMAC_FTR7	0x005C	W	0x0	Fault type for DMA Channel 7
-	-	-	-	reserved
DMAC_CSR0	0x0100	W	0x0	Channel Status for DMA Channel 0
DMAC_CSR1	0x0108	W	0x0	Channel Status for DMA Channel 1
DMAC_CSR2	0x0110	W	0x0	Channel Status for DMA Channel 2
DMAC_CSR3	0x0118	W	0x0	Channel Status for DMA Channel 3
DMAC_CSR4	0x0120	W	0x0	Channel Status for DMA Channel 4
DMAC_CSR5	0x0128	W	0x0	Channel Status for DMA Channel 5
DMAC_CSR6	0x0130	W	0x0	Channel Status for DMA Channel 6
DMAC_CSR7	0x0138	W	0x0	Channel Status for DMA Channel 7
DMAC_CPC0	0x0104	W	0x0	Channel PC for DMA Channel 0
DMAC_CPC1	0x010c	W	0x0	Channel PC for DMA Channel 1
DMAC_CPC2	0x0114	W	0x0	Channel PC for DMA Channel 2
DMAC_CPC3	0x011c	W	0x0	Channel PC for DMA Channel 3
DMAC_CPC4	0x0124	W	0x0	Channel PC for DMA Channel 4
DMAC_CPC5	0x012c	W	0x0	Channel PC for DMA Channel 5
DMAC_CPC6	0x0134	W	0x0	Channel PC for DMA Channel 6
DMAC_CPC7	0x013c	W	0x0	Channel PC for DMA Channel 7
DMAC_SAR0	0x0400	W	0x0	Source Address for DMA Channel 0
DMAC_SAR1	0x0420	W	0x0	Source Address for DMA Channel 1
DMAC_SAR2	0x0440	W	0x0	Source Address for DMA Channel 2
DMAC_SAR3	0x0460	W	0x0	Source Address for DMA Channel 3
DMAC_SAR4	0x0480	W	0x0	Source Address for DMA Channel 4
DMAC_SAR5	0x04A0	W	0x0	Source Address for DMA Channel 5
DMAC_SAR6	0x04C0	W	0x0	Source Address for DMA Channel 6
DMAC_SAR7	0x04E0	W	0x0	Source Address for DMA Channel 7
DMAC_DAR0	0x0404	W	0x0	Dest Address for DMAChannel 0
DMAC_DAR1	0x0424	W	0x0	Dest Address for DMAChannel 1

DMAC_DAR2	0x0444	W	0x0	Dest Address for DMAChannel 2
DMAC_DAR3	0x0464	W	0x0	Dest Address for DMAChannel 3
DMAC_DAR4	0x0484	W	0x0	Dest Address for DMAChannel 4
DMAC_DAR5	0x04A4	W	0x0	Dest Address for DMAChannel 5
DMAC_DAR6	0x04C4	W	0x0	Dest Address for DMAChannel 6
DMAC_DAR7	0x04E4	W	0x0	Dest Address for DMAChannel 7
DMAC_CCR0	0x0408	W	0x0	Channel Control for DMA Channel 0
DMAC_CCR1	0x0428	W	0x0	Channel Control for DMA Channel 1
DMAC_CCR2	0x0448	W	0x0	Channel Control for DMA Channel 2
DMAC_CCR3	0x0468	W	0x0	Channel Control for DMA Channel 3
DMAC_CCR4	0x0488	W	0x0	Channel Control for DMA Channel 4
DMAC_CCR5	0x04a8	W	0x0	Channel Control for DMA Channel 5
DMAC_CCR6	0x04c8	W	0x0	Channel Control for DMA Channel 6
DMAC_CCR7	0x04e8	W	0x0	Channel Control for DMA Channel 7
DMAC_LC0_0	0x040C	W	0x0	Loop Counter 0 for DMA Channel 0
DMAC_LC0_1	0x042C	W	0x0	Loop Counter 0 for DMA Channel 1
DMAC_LC0_2	0x044C	W	0x0	Loop Counter 0 for DMA Channel 2
DMAC_LC0_3	0x046C	W	0x0	Loop Counter 0 for DMA Channel 3
DMAC_LC0_4	0x048C	W	0x0	Loop Counter 0 for DMA Channel 4
DMAC_LC0_5	0x04AC	W	0x0	Loop Counter 0 for DMA Channel 5
DMAC_LC0_6	0x04CC	W	0x0	Loop Counter 0 for DMA Channel 6
DMAC_LC0_7	0x04EC	W	0x0	Loop Counter 0 for DMA Channel 7
DMAC_LC1_0	0x0410	W	0x0	Loop Counter 1 for DMA Channel 0
DMAC_LC1_1	0x0430	W	0x0	Loop Counter 1 for DMA Channel 1
DMAC_LC1_2	0x0450	W	0x0	Loop Counter 1 for DMA Channel 2
DMAC_LC1_3	0x0470	W	0x0	Loop Counter 1 for DMA Channel 3
DMAC_LC1_4	0x0490	W	0x0	Loop Counter 1 for DMA

				Channel 4
DMAC_LC1_5	0x04B0	W	0x0	Loop Counter 1 for DMA Channel 5
DMAC_LC1_6	0x04D0	W	0x0	Loop Counter 1 for DMA Channel 6
DMAC_LC1_7	0x04F0	W	0x0	Loop Counter 1 for DMA Channel 7
-	-	-	-	reserved
DMAC_DBGST DMACATUS	0x0D00	W	0x0	Debug Status Register.
DMAC_DBGCMD	0x0D04	W	0x0	Debug Command Register.
DMAC_DBGINST0	0x0D08	W	0x0	Debug Instruction-0 Register.
DMAC_DBGINST1	0x0D0C	W	0x0	Debug Instruction-1 Register.
DMAC_CR0	0x0E00	W		Configuration Register 0.
DMAC_CR1	0x0E04	W		Configuration Register 1.
DMAC_CR2	0x0E08	W		Configuration Register 2.
DMAC_CR3	0x0E0C	W		Configuration Register 3.
DMAC_CR4	0x0E10	W		Configuration Register 4.
DMAC_CRDn	0x0E14	W		Configuration Register Dn.
DMAC_WD	0X0E80	W		Watchdog Register

Notes:

Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** – WORD (32 bits) access

8.4.2 Detail Register Description

DMAC_DSR

Address: Operational Base+0x0

DMA Manager Status Register

Bit	Attr	Reset Value	Description
31:10	-	-	Reserved
9	R	0x0	Provides the security status of the DMA manager thread: 0 = DMA manager operates in the Secure state 1 = DMA manager operates in the Non-secure state.
8:4	R	0x0	When the DMA manager thread executes a DMAWFE instruction, it waits for the following event to occur: b00000 = event[0] b00001 = event[1] b00010 = event[2] ... b11111 = event[31].
3:0	R	0x0	The operating state of the DMA manager: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event

			b0101-b1110 = reserved b1111 = Faulting.
--	--	--	---

DMAC_DPC

Address:Operational Base+0x4

DMA Program Counter Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA manager thread

DMAC_INTEN

Address:Operational Base+0x20

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x0	<p>Program the appropriate bit to control how the DMAC responds when it executes DMASEV:</p> <p>Bit [N] = 0 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC signals event N to all of the threads. Set bit [N] to 0 if your system design does not use irq[N] to signal an interrupt request.</p> <p>Bit [N] = 1 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC sets irq[N] HIGH. Set bit [N] to 1 if your system designer requires irq[N] to signal an interrupt request.</p>

DMAC_EVENT_RIS

Address:Operational Base+0x24

Event-Interrupt Raw Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	<p>Returns the status of the event-interrupt resources:</p> <p>Bit [N] = 0 Event N is inactive or irq[N] is LOW.</p> <p>Bit [N] = 1 Event N is active or irq[N] is HIGH.</p>

DMAC_INTMIS

Address:Operational Base+0x28

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	<p>Provides the status of the interrupts that are active in the DMAC:</p> <p>Bit [N] = 0 Interrupt N is inactive and therefore irq[N] is LOW.</p> <p>Bit [N] = 1 Interrupt N is active and therefore irq[N] is HIGH</p>

DMAC_INTCLR

Address:Operational Base+0x2c

Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:0	W	0x0	Controls the clearing of the irq outputs:

Bit [N] = 0 The status of irq[N] does not

			<p>change.</p> <p>Bit [N] = 1 The DMAC sets irq[N] LOW if the INTEN Register programs the DMAC to signal an interrupt.</p> <p>Otherwise, the status of irq[N] does not change.</p>
--	--	--	--

DMAC_FSRD

Address: Operational Base+0x30

Fault Status DMA Manager Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	<p>Provides the fault status of the DMA manager.</p> <p>Read as:</p> <p>0 = the DMA manager thread is not in the Faulting state</p> <p>1 = the DMA manager thread is in the Faulting state.</p>

DMAC_FSRC

Address: Operational Base+0x34

Fault Status DMA Channel Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	<p>Each bit provides the fault status of the corresponding channel. Read as:</p> <p>Bit [N] = 0 No fault is present on DMA channel N.</p> <p>Bit [N] = 1 DMA channel N is in the Faulting or Faulting completing state.</p>

DMAC_FTRD

Address: Operational Base+0x38

Fault Type DMA Manager Register

Bit	Attr	Reset Value	Description
31	-	-	reserved
30	R	0x0	<p>If the DMA manager aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface:</p> <p>0 = instruction that generated an abort was read from system memory</p> <p>1 = instruction that generated an abort was read from the debug interface.</p>
29:17	-	-	reserved
16	R	0x0	<p>Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA manager performs an instruction fetch:</p> <p>0 = OKAY response</p> <p>1 = EXOKAY, SLVERR, or DECERR response</p>
15:6	-	-	reserved
5	R	0x0	<p>Indicates if the DMA manager was attempting to execute DMAWFE or DMASEV with inappropriate security permissions:</p> <p>0 = DMA manager has appropriate security to execute DMAWFE or DMASEV</p> <p>1 = a DMA manager thread in the Non-secure state attempted to execute either:</p>

			<ul style="list-style-type: none"> • DMAWFE to wait for a secure event • DMASEV to create a secure event or secure interrupt
4	R	0x0	<p>Indicates if the DMA manager was attempting to execute DMAGO with inappropriate security permissions:</p> <p>0 = DMA manager has appropriate security to execute DMAGO</p> <p>1 = a DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state.</p>
3:2	-	-	reserved
1	R	0x0	<p>Indicates if the DMA manager was attempting to execute an instruction operand that was not valid for the configuration of the DMAC:</p> <p>0 = valid operand</p> <p>1 = invalid operand.</p>
0	R	0x0	<p>Indicates if the DMA manager was attempting to execute an undefined instruction:</p> <p>0 = defined instruction</p> <p>1 = undefined instruction.</p>

DMAC_FTR0~DMAC_FTR7

Address: Operational Base+0x40

Operational Base+0x44
 Operational Base+0x48
 Operational Base+0x4C
 Operational Base+0x50
 Operational Base+0x54
 Operational Base+0x58
 Operational Base+0x5C

Fault Type DMA Channel Register

Bit	Attr	Reset Value	Description
31	R	0x0	<p>Indicates if the DMA channel has locked-up because of resource starvation:</p> <p>0 = DMA channel has adequate resources</p> <p>1 = DMA channel has locked-up because of insufficient resources.</p> <p>This fault is an imprecise abort</p>
30	R	0x0	<p>If the DMA channel aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface:</p> <p>0 = instruction that generated an abort was read from system memory</p> <p>1 = instruction that generated an abort was read from the debug interface.</p> <p>This fault is an imprecise abort but the bit is only valid when a precise abort occurs.</p>
29:19	-	-	reserved
18	R	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs a data read:

			1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	R	0x0	Indicates the AXI response that the DMAC receives on the BRESP bus, after the DMA channel thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort.
16	R	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort.
15:14	-	-	reserved
13	R	0x0	Indicates if the MFIFO did not contain the data to enable the DMAC to perform the DMAST: 0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort.
12	R	0x0	Indicates if the MFIFO prevented the DMA channel thread from executing DMALD or DMAST. Depending on the instruction: DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	-	-	reserved
7	R	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to program the CCRn Register to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	R	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to execute DMAWFP, DMALDP, DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: <ul style="list-style-type: none">• DMAWFP to wait for a secure peripheral• DMALDP or DMASTP to notify a secure peripheral

			<ul style="list-style-type: none"> • DMAFLUSHP to flush a secure peripheral. <p>This fault is a precise abort.</p>
5	R	0x0	<p>Indicates if the DMA channel thread attempts to execute DMAWFE or DMASEV with inappropriate security permissions:</p> <p>0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either:</p> <ul style="list-style-type: none"> • DMAWFE to wait for a secure event • DMASEV to create a secure event or secure interrupt. <p>This fault is a precise abort.</p>
4:2	-	-	reserved
1	R	0x0	<p>Indicates if the DMA channel thread was attempting to execute an instruction operand that was not valid for the configuration of the DMAC:</p> <p>0 = valid operand 1 = invalid operand.</p> <p>This fault is a precise abort.</p>
0	R	0x0	<p>Indicates if the DMA channel thread was attempting to execute an undefined instruction:</p> <p>0 = defined instruction 1 = undefined instruction.</p> <p>This fault is a precise abort</p>

DMAC_CSR0~DMAC_CSR7

Address: Operational Base+0x100
 Operational Base+0x108
 Operational Base+0x110
 Operational Base+0x118
 Operational Base+0x120
 Operational Base+0x128
 Operational Base+0x130
 Operational Base+0x138

Channel Status Registers

Bit	Attr	Reset Value	Description
31:22	-	-	reserved
21	R	0x0	<p>The channel non-secure bit provides the security of the DMA channel:</p> <p>0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state</p>
20:16	-	-	reserved
15	R	0x0	<p>When the DMA channel thread executes DMAWFP this bit indicates if the periph operand was set:</p> <p>0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set</p>
14	R	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the burst or single

			operand were set: 0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set.
13:9	-	-	reserved
8:4	R	0x0	If the DMA channel is in the Waiting for event state or the Waiting for peripheral state then these bits indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31
3:0	R	0x0	The channel status encoding is: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event b0101 = At barrier b0110 = reserved b0111 = Waiting for peripheral b1000 = Killing b1001 = Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMAC_CPC0~DMAC_CPC7

Address: Operational Base+0x104

Operational Base+0x10C
 Operational Base+0x114
 Operational Base+0x11c
 Operational Base+0x124
 Operational Base+0x12C
 Operational Base+0x134
 Operational Base+0x13C

Channel Program Counter Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA channel n thread

DMAC_SAR0~DMAC_SAR7

Address: Operational Base+0x400

Operational Base+0x420
 Operational Base+0x440
 Operational Base+0x460

Operational Base+0x480
 Operational Base+0x4A0
 Operational Base+0x4C0
 Operational Base+0x4E0

Source Address Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the source data for DMA channel n

DMAC_DAR0~DMAC_DAR7

Address: Operational Base+0x404
 Operational Base+0x424
 Operational Base+0x444
 Operational Base+0x464
 Operational Base+0x484
 Operational Base+0x4A4
 Operational Base+0x4C4
 Operational Base+0x4E4

DestinationAddress Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the Destinationdata for DMA channel n

DMAC_CCR0~DMAC_CCR7

Address: Operational Base+0x408
 Operational Base+0x428
 Operational Base+0x448
 Operational Base+0x468
 Operational Base+0x488
 Operational Base+0x4A8
 Operational Base+0x4C8
 Operational Base+0x4E8

Channel Control Registers

Bit	Attr	Reset Value	Description
31:28	-	-	reserved
27:25	R	0x0	Programs the state of AWCACHE[3,1:0]a when the DMAC writes the destination data. Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	R	0x0	Programs the state of AWPROT[2:0]a when the DMAC writes the destination data. Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	R	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it writes the destination data: b0000 = 1 data transfer

			b0001 = 2 data transfers b0010 = 3 data transfers . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size
17:15	R	0x0	For each beat within a burst, it programs the number of bytes that the DMAC writes to the destination: b000 = writes 1 byte per beat b001 = writes 2 bytes per beat b010 = writes 4 bytes per beat b011 = writes 8 bytes per beat b100 = writes 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	R	0x0	Programs the burst type that the DMAC performs when it writes the destination data: 0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	R	0x0	Set the bits to control the state of ARCACHE[2:0]a when the DMAC reads the source data. Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH.
10:8	R	0x0	Programs the state of ARPROT[2:0]a when the DMAC reads the source data. Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH.
7:4	R	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it reads the source data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers .

			b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMA LD instruction is the product of src_burst_len and src_burst_size
3:1	R	0x0	For each beat within a burst, it programs the number of bytes that the DMAC reads from the source: b000 = reads 1 byte per beat b001 = reads 2 bytes per beat b010 = reads 4 bytes per beat b011 = reads 8 bytes per beat b100 = reads 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMA LD instruction is the product of src_burst_len and src_burst_size
0	R	0x0	Programs the burst type that the DMAC performs when it reads the source data: 0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMAC_LC0_0~DMAC_LC0_7

Address: Operational Base+0x40c
 Operational Base+0x42C
 Operational Base+0x44C
 Operational Base+0x46C
 Operational Base+0x48C
 Operational Base+0x4AC
 Operational Base+0x4CC
 Operational Base+0x4EC

Loop Counter 0 Registers

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	R	0x0	Loop counter 0 iterations

DMAC_LC1_0~DMAC_LC1_7

Address: Operational Base+0x410
 Operational Base+0x430
 Operational Base+0x450
 Operational Base+0x470
 Operational Base+0x490
 Operational Base+0x4B0
 Operational Base+0x4D0
 Operational Base+0x4F0

Loop Counter 1 Registers

Bit	Attr	Reset Value	Description
31:8	-	-	reserved

7:0	R	0x0	Loop counter 1 iterations
-----	---	-----	---------------------------

DMAC_DBGSTATUS

Address: Operational Base+0xD00

Debug Status Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved
1:0	R	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved.

DMAC_DBGCMD

Address: Operational Base+0xD04

Debug Command Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved
1:0	W	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved

DMAC_DBGINST0

Address: Operational Base+0xD08

Debug Instruction-0 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 1
23:16	W	0x0	Instruction byte 0
17:11	-	-	reserved
10:8	W	0x0	DMA channel number: b000 = DMA channel 0 b001 = DMA channel 1 b010 = DMA channel 2 ... b111 = DMA channel 7
7:1	-	-	reserved
0	W	0x0	The debug thread encoding is as follows: 0 = DMA manager thread 1 = DMA channel.

DMAC_DBGINST1

Address: Operational Base+0xD0C

Debug Instruction-1 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 5
23:16	W	0x0	Instruction byte 4
15:8	W	0x0	Instruction byte 3
7:0	W	0x0	Instruction byte 2

DMAC_CR0

Address: Operational Base+0xE00

Configuration Register 0

Bit	Attr	Reset Value	Description
31:22	-	-	reserved
21:17	R	0x2	<p>Number of interrupt outputs that the DMAC provides:</p> <p>b00000 = 1 interrupt output, irq[0] b00001 = 2 interrupt outputs, irq[1:0] b00010 = 3 interrupt outputs, irq[2:0]</p> <p>.</p> <p>.</p> <p>.</p> <p>b11111 = 32 interrupt outputs, irq[31:0].</p>
16:12	R	0x7	<p>Number of peripheral request interfaces that the DMAC provides:</p> <p>b00000 = 1 peripheral request interface b00001 = 2 peripheral request interfaces b00010 = 3 peripheral request interfaces</p> <p>.</p> <p>.</p> <p>.</p> <p>b11111 = 32 peripheral request interfaces.</p>
11:7	-	-	reserved
6:4	R	0x5	<p>Number of DMA channels that the DMAC supports:</p> <p>b000 = 1 DMA channel b001 = 2 DMA channels b010 = 3 DMA channels</p> <p>.</p> <p>.</p> <p>.</p> <p>b111 = 8 DMA channels.</p>
3	-	-	reserved
2	R	0x0	<p>Indicates the status of the boot_manager_ns signal when the DMAC exited from reset:</p> <p>0 = boot_manager_ns was LOW 1 = boot_manager_ns was HIGH.</p>
1	R	0x0	<p>Indicates the status of the boot_from_pc signal when the DMAC exited from reset:</p> <p>0 = boot_from_pc was LOW 1 = boot_from_pc was HIGH</p>
0	R	0x1	<p>Supports peripheral requests:</p> <p>0 = the DMAC does not provide a peripheral request interface 1 = the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies.</p>

DMAC_CR1

Address: Operational Base+0xE04

Configuration Register 1

Bit	Attr	Reset Value	Description
31:8	-	-	reserved

7:4	R	0x5	[7:4] num_i-cache_lines Number of i-cache lines: b0000 = 1 i-cache line b0001 = 2 i-cache lines b0010 = 3 i-cache lines ... b1111 = 16 i-cache lines.
3	-	-	reserved
2:0	R	0x7	The length of an i-cache line: b000-b001 = reserved b010 = 4 bytes b011 = 8 bytes b100 = 16 bytes b101 = 32 bytes b110-b111 = reserved

DMAC_CR2

Address: Operational Base+0xE08

Configuration Register 2

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMAC_CR3

Address: Operational Base+0xE0C

Configuration Register 3

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the security state of an event-interrupt resource: Bit [N] = 0 Assigns event<N> or irq[N] to the Secure state. Bit [N] = 1 Assigns event<N> or irq[N] to the Non-secure state.

DMAC_CR4

Address: Operational Base+0xE10

Configuration Register 4

Bit	Attr	Reset Value	Description
31:0	R	0x6	Provides the security state of the peripheral request interfaces: Bit [N] = 0 Assigns peripheral request interface N to the Secure state. Bit [N] = 1 Assigns peripheral request interface N to the Non-secure state

DMAC_CRDn

Address: Operational Base+0xE14

DMA Configuration Register

Bit	Attr	Reset Value	Description
31:30	-	-	reserved
29:20	R	0x20	The number of lines that the data buffer contains: b000000000 = 1 line b000000001 = 2 lines

			... b111111111 = 1024 lines
19:16	R	0x9	The depth of the read queue: b0000 = 1 line b0001 = 2 lines . . . b1111 = 16 lines.
15	-	-	reserved
14:12	R	0x4	Read issuing capability that programs the number of outstanding read transactions: b000 = 1 b001 = 2 . . . b111 = 8
11:8	R	0x7	The depth of the write queue: b0000 = 1 line b0001 = 2 lines . . . b1111 = 16 lines.
7	-	-	reserved
6:4	R	0x3	Write issuing capability that programs the number of outstanding write transactions: b000 = 1 b001 = 2 . . . b111 = 8
3	-	-	reserved
2:0		0x3	The data bus width of the AXI interface: b000 = reserved b001 = reserved b010 = 32-bit b011 = 64-bit b100 = 128-bit b101-b111 = reserved.

DMAC_WD

Address: Operational Base+0xE80

DMA Watchdog Register

Bit	Attr	Reset Value	Description
31:1	-	-	reserved
0	RW	0x0	Controls how the DMAC responds when it detects a lock-up condition: 0 = the DMAC aborts all of the contributing DMA channels and sets irq_abort HIGH 1 = the DMAC sets irq_abort HIGH.

8.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

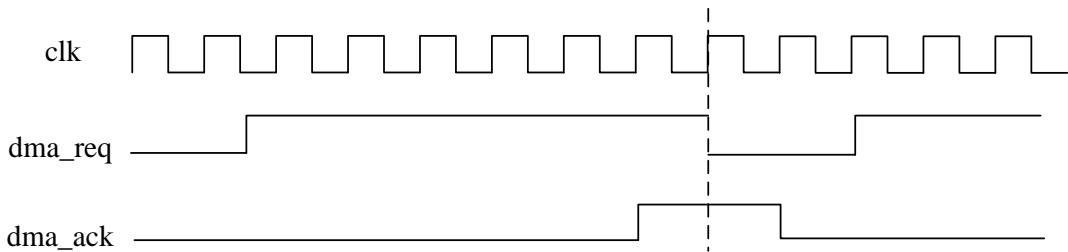


Fig. 8-3 DMAC0 request and acknowledge timing

8.6 Interface Description

DMAC has the following tie-off signals. It can be configured by GRF register.(Please refer to the chapter to find how to configure)

DMAC

interface	Reset value	Control source
boot_addr	0x0	GRF
boot_from_pc	0x0	GRF
boot_manager_ns	0x0	GRF
boot_irq_ns	0xf	GRF
boot_periph_ns	0xfffff	GRF

boot_addr

Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

boot_from_pc

Controls the location in which the DMAC executes its initial instruction, after it exits from reset:

0 = DMAC waits for an instruction from either APB interface

1 = DMA manager thread executes the instruction that is located at the address that

boot_manager_ns

When the DMAC exits from reset, this signal controls the security state of the DMA manager thread:

0 = assigns DMA manager to the Secure state

1 = assigns DMA manager to the Non-secure state.

boot_irq_ns

Controls the security state of an event-interrupt resource, when the DMAC exits from reset:

boot_irq_ns[x] is LOW

The DMAC assigns event<x> or irq[x] to the Secure state.

boot_irq_ns[x] is HIGH

The DMAC assigns event<x> or irq[x] to the Non-secure state.

boot_periph_ns

Controls the security state of a peripheral request interface, when the DMAC exits from reset:

boot_periph_ns[x] is LOW

The DMAC assigns peripheral request interface x to the Secure state.
boot_periph_ns[x] is HIGH
The DMAC assigns peripheral request interface x to the Non-secure state.

8.7 Application Notes

8.7.1 Using the APB slave interfaces

You must ensure that you use the appropriate APB interface, depending on the security state in which the boot_manager_ns initializes the DMAC to operate. For example, if the DMAC is in the secure state, you must issue the instruction using the secure APB interface, otherwise the DMAC ignores the instruction. You can use the secure APB interface, or the non-secure APB interface, to start or restart a DMA channel when the DMAC is in the Non-secure state.

The necessary steps to start a DMA channel thread using the debug instruction registers as following:

1. Create a program for the DMA channel.
2. Store the program in a region of system memory.
3. Poll the DBGSTATUS Register to ensure that debug is idle, that is, the dbgstatus bit is 0.
4. Write to the DBGINST0 Register and enter the:
 - Instruction byte 0 encoding for DMAGO.
 - Instruction byte 1 encoding for DMAGO.
 - Debug thread bit to 0. This selects the DMA manager thread.
5. Write to the DBGINST1 Register with the DMAGO instruction byte [5:2] data, see Debug Instruction-1 Register o. You must set these four bytes to the address of the first instruction in the program, that was written to system memory in step 2.
6. Writing zero to the DBGCMD Register. The DMAC starts the DMA channel thread and sets the dbgstatus bit to 1.

8.7.2 Security usage

When the DMAC exits from reset, the status of the configuration signals that tie-off signals which described in chapter 10.6.

DMA manager thread is in the secure state

If the DNS bit is 0, the DMA manager thread operates in the secure state and it only performs secure instruction fetches. When a DMA manager thread in the secure state processes:

DMAGO

It uses the status of the ns bit, to set the security state of the DMAchannel thread by writing to the CNS bit for that channel.

DMAWFE

It halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit.

DMASEV

It sets the corresponding bit in the INT_EVENT_RIS Register, irrespective of the security state of the corresponding INS bit.

DMA manager thread is in the Non-secure state

If the DNS bit is 1, the DMA manager thread operates in the Non-secure state, and it only performs non-secure instruction fetches. When a DMA manager thread in the Non-secure state processes:

DMAGO

The DMAC uses the status of the ns bit, to control if it starts a DMA channel thread. If:

ns = 0

The DMAC does not start a DMA channel thread and instead it:

1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager
3. Sets the dmago_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

ns = 1

The DMAC starts a DMA channel thread in the Non-secure state and programs the CNS bit to be non-secure.

DMAWFE

The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:

INS = 0

The event is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

INS = 1

The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.

DMASEV

The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it creates the event-interrupt. If:

INS = 0

The event-interrupt resource is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

INS = 1

The event-interrupt resource is in the Non-secure state. The DMAC creates the

event-interrupt.

DMA channel thread is in the secure state

When the CNS bit is 0, the DMA channel thread is programmed to operate in the Secure state and it only performs secure instruction fetches.

When a DMA channel thread in the secure state processes the following instructions:

DMAWFE

The DMAC halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit, in the CR3 Register.

DMASEV

The DMAC creates the event-interrupt, irrespective of the security state of the corresponding INS bit, in the CR3 Register.

DMAWFP

The DMAC halts execution of the thread until the peripheral signals a DMA request. When this occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

DMALDP, DMASTP

The DMAC sends a message to the peripheral to communicate that data transfer is complete, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

DMAFLUSHP

The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

When a DMA channel thread is in the Secure state, it enables the DMAC to perform secure and non-secure AXI accesses

DMA channel thread is in the Non-secure state

When the CNS bit is 1, the DMA channel thread is programmed to operate in the Non-secure state and it only performs non-secure instruction fetches.

When a DMA channel thread in the Non-secure state processes the following instructions:

DMAWFE

The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:

INS = 0

The event is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

INS = 1

The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.

DMASEV

The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it creates the event. If:

INS = 0

The event-interrupt resource is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers .
4. Moves the DMA channel to the Faulting completing state.

INS = 1

The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.

DMAWFP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it waits for the peripheral to signal a request. If:

PNS = 0

The peripheral is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC halts execution of the thread and waits for the peripheral to signal a request.

DMALDP, DMASTP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends an acknowledgement to the peripheral. If:

PNS = 0

The peripheral is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC sends a message to the peripheral to communicate when the data transfer is complete.

DMAFLUSHP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends a flush request to the peripheral. If:

PNS = 0

The peripheral is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status.

When a DMA channel thread is in the Non-secure state, and a DMAMOV CCR instruction attempts to program the channel to perform a secure AXI transaction, the DMAC:

1. Executes a DMANOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_rdwr_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel thread to the Faulting completing state.

8.7.3 Programming restrictions

Fixed unaligned bursts

The DMAC does not support fixed unaligned bursts. If you program the following conditions, the DMAC treats this as a programming error:

Unaligned read

- src_inc field is 0 in the CCRn Register
- the SARn Register contains an address that is not aligned to the size of data that the src_burst_size field contain

Unaligned write

- dst_inc field is 0 in the CCRn Register
- the DARn Register contains an address that is not aligned to the size of data that the dst_burst_size field contains

Endian swap size restrictions

If you program the endian_swap_size field in the CCRn Register, to enable a DMA channel to perform an endian swap then you must set the corresponding SARn Register and the corresponding DARn Register to contain an address that is aligned to the value that the endian_swap_size field contains.

Updating DMA channel control registers during a DMA cycle restrictions

Prior to the DMAC executing a sequence of DMALD and DMAST instructions, the values you program in to the CCRn Register, SARn Register, and DARn Register control the data byte lane manipulation that the DMAC performs when it

transfers the data from the source address to the destination address. You'd better not update these registers during a DMA cycle.

Resource sharing between DMA channels

DMA channel programs share the MFIFO data storage resource. You must not start a set of concurrently running DMA channel programs with a resource requirement that exceeds the configured size of the MFIFO. If you exceed this limit then the DMAC might lock up and generate a Watchdog abort.

8.7.4 Unaligned transfers may be corrupted

For a configuration with more than one channel, if any of channels 1 to 7 is performing transfers between certain types of misaligned source and destination addresses, then the output data may be corrupted by the action of channel 0.

Data corruption might occur if all of the following are true:

1. Two beats of AXI read data are received for one of channels 1 to 7.
2. Source and destination address alignments mean that each read data beat is split across two lines in the data buffer (see Splitting data, below).
3. There is one idle cycle between the two read data beats .
4. Channel 0 performs an operation that updates channel control information during this idle cycle (see Updates to channel control information, below)

Splitting data

Depending upon the programmed values for the DMA transfer, one beat of read data from the AXI interface may need to be split across two lines in the internal data buffer. This occurs when the read data beat contains datbytes which will be written to addresses that wrap around at the AXI interface data width, so that these bytes could not be transferred by a single AXI write data beat of the full interface width.

Most applications of DMA-330 do not split data in this way, so are NOT vulnerable to data corruption from this defect.

The following cases are NOT vulnerable to data corruption because they do not split data:

- Byte lane offset between source and destination addresses is 0 When source and destination addresses have the same byte lane alignment, the offset is 0 and a wrap operation that splits data cannot occur.
- Byte lane offset between source and destination addresses is a multiple of source size

Source size in CCRn	Allowed offset between SARn and DARn
SS8	any offset allowed.
SS16	0,2,4,6,8,10,12,14
SS32	0,4,8,12
SS64	0,8

8.7.5 Interrupt shares between channel.

As the DMAC0 does not record which channel (or list of channels) have asserted an interrupt. So it will depend on your program and whether any of the visible information for that program can be used to determine progress, and help

identify the interrupt source.

There are 4 likely information sources that can be used to determine the progress made by a program:

- Program counter (PC)
- Source address
- Destination address
- Loop counters (LC)

For example, a program might emit an interrupt each time that it iterates around a loop. In this case, the interrupt service routine (ISR) would need to store the loop value of each channel when it is called, and then compare against the new value when it is next called. A change in value would indicate that the program has progressed.

The ISR must be carefully written to ensure that no interrupts are lost. The sequence of operations is as follows:

1. Disable interrupts
2. Immediately clear the interrupt in DMA-330
3. Check the relevant registers for both channels to determine which must be serviced
4. Take appropriate action for the channels
5. Re-enable interrupts and exit ISR

8.7.6 Instruction sets

Table 8-2 DMAC Instruction sets

Mnemonic	Instruction	Thread usage: • M = DMA manager • C = DMA channel
DMAADDH	Add Halfword	C
DMAEND	End	M/C
DMAFLUSHP	Flush and notify Peripheral	C
DMAGO	Go	M
DMAKILL	Kill	C
DMALD	Load	C
DMALDP	Load Peripheral	C
DMALP	Loop	C
DMALPEND	Loop End	C
DMALPFE	Loop Forever	C
DMAMOV	Move	C
DMANOP	No operation	M/C
DMARMB	Read Memory Barrier	C
DMASEV	Send Event	M/C
DMAST	Store	C
DMASTP	Store and notify Peripheral	C
DMASTZ	Store Zero	C
DMAWFE	Wait For Event M	M/C
DMAWFP	Wait For Peripheral	C
DMAWMB	Write Memory Barrier	C
DMAADNH	Add Negative Halfword	C

8.7.7 Assembler directives

In this document, only DMMADNH instruction is took as an example to show the way the instruction assembled. *For the other instructions , please refer to pl330_trm.pdf.*

DMAADNH

Add Negative Halfword adds an immediate negative 16-bit value to the SARn Register or DARn Register, for the DMA channel thread. This enables the DMA C to support 2D DMA operations, or reading or writing an area of memory in a different order to naturally incrementing addresses. See Source Address Registers and Destination Address Registers.

The immediate unsigned 16-bit value is one-extended to 32 bits, to create a value that is the two's complement representation of a negative number between -65536 and -1, before the DMAC adds it to the address using 32-bit addition. The DMAC discards the carry bit so that addresses wrap from 0xFFFFFFFF to 0x00000000. The net effect is to subtract between 65536 and 1 from the current value in the Source or Destination Address Register.

Following table shows the instruction encoding.

Imm[15:8]	Imm[7:0]	0	1	0	1	1	1	ra	0
-----------	----------	---	---	---	---	---	---	----	---

Assembler syntax

DMAADNH <address_register>, <16-bit immediate>

where:

<address_register>

Selects the address register to use. It must be either:

SAR

SARn Register and sets ra to 0.

DAR

DARn Register and sets ra to 1.

<16-bit immediate>

The immediate value to be added to the <address_register>.

You should specify the 16-bit immediate as the number that is to be represented in the instruction encoding. For example, DMAADNH DAR, 0xFFFF causes the value 0xFFFFFFFF to be added to the current value of the Destination Address Register, effectively subtracting 16 from the DAR.

You can only use this instruction in a DMA channel thread.

8.7.8 MFIFO usage

For MFIFO usage , please refer to pl330_trm.pdf

Chapter 9 Generic Interrupt Controller (GIC)

9.1 Overview

The generic interrupt controller(GIC) in this device has two interfaces, the distributor interface connects to the interrupt source, and the CPU interface connects to Cortex-A7.

It supports the following features:

- Supports 128hardware interrupt inputs
- Masking of any interrupts
- Prioritization of interrupts
- Distribution of the interrupts to the target Cortex-A7 processor(s)
- Generation of interrupts by software
- Supports Security Extensions

9.2 Block Diagram

Fig.12-1 shows the block diagram of GIC.

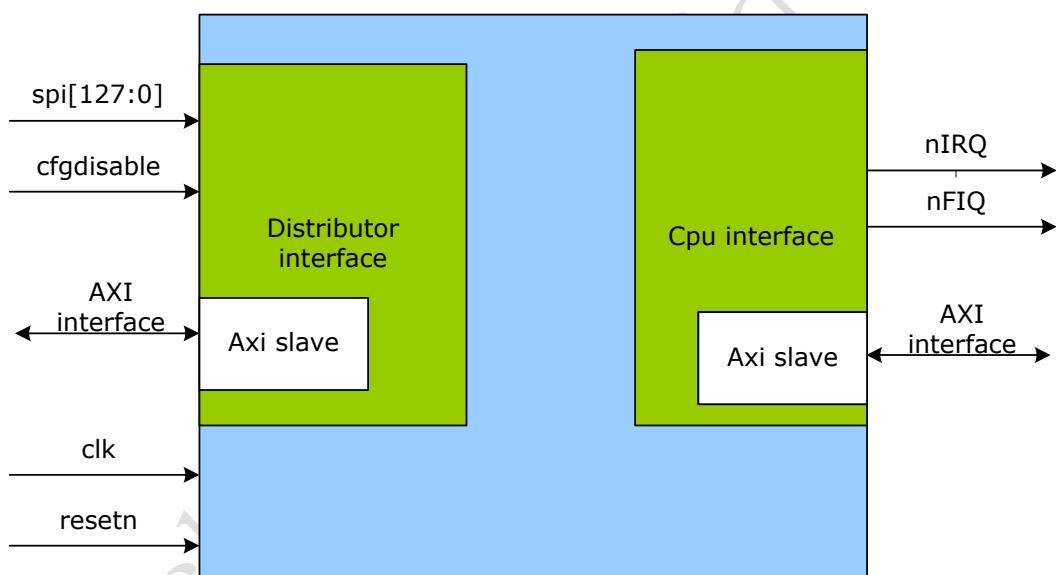


Fig. 9-1Block diagram of GIC

The diagram shows that GIC has two AXI interfaces independently and has two base address for these two interfaces.

9.3 Function Description

This GIC architecture splits logically into a Distributor block and one CPU interface block, as Figure 12-1 shows.

Distributor

This performs interrupt prioritization and distribution to the CPU interface that connect to the processor in the system.

CPU interface

CPU interface performs priority masking and preemption handling for a connected processor in the system.

9.3.1 The Distributor

The Distributor centralizes all interrupt sources, determines the priority of each interrupt, and for CPU interface dispatches the interrupt with the highest priority to the interface for priority masking and preemption handling.

The Distributor provides a programming interface for:

- Globally enabling the forwarding of interrupts to the CPU interface
- Enabling or disabling each interrupt
- Setting the priority level of each interrupt
- Setting the target processor list of each interrupt
- Setting each peripheral interrupt to be level-sensitive or edge-triggered
- If the GIC implements the Security Extensions, setting each interrupt as either
- Secure or Non-secure
- Sending a Software-generated interrupt (SGI) to processor.
- Visibility of the state of each interrupt
- A mechanism for software to set or clear the pending state of a peripheral interrupt.

Interrupt ID

Interrupts from sources are identified using ID numbers. CPU interface can see up to 160 interrupts.

The GIC assigns interrupt these 128 ID numbers as follows:

- Interrupt numbers ID32-ID127 are used for SPIs(shared peripheral interrupts).
- ID0-ID15 are used for SGI.
- ID16-ID31 are used for Private peripheral interrupt (PPI).

The GIC architecture reserves interrupt ID numbers 1022-1023 for special purposes.

ID1022

The GIC returns this value to a processor in response to an interrupt acknowledge only when the following apply:

- The interrupt acknowledge is a Secure read
- The highest priority pending interrupt is Non-secure
- The AckCtl bit in the Secure ICCICR is set to 0
- The priority of the interrupt is sufficient for it to be signalled to the processor.

Interrupt ID 1022 informs secure software that there is a Non-secure interrupt of sufficient priority to be signalled to the processor, that must be handled by Non-secure software. In this situation the secure software might alter its schedule to permit Non-secure software to handle the interrupt, to minimize the interrupt latency.

ID1023

This value is returned to a processor, in response to an interrupt acknowledge, if there is no pending interrupt with sufficient priority for it to be signalled to the processor.

On a processor that implements the Security Extensions, Secure software treats values of 1022 and 1023 as spurious interrupts.

9.3.2 CPU interface

CPU interface block provides the interface for a processor that operates with the GIC. CPU interface provides a programming interface for:

- Enabling the signal of interrupt requests by the CPU interface
- Acknowledging an interrupt
- Indicating completion of the processing of an interrupt
- Setting an interrupt priority mask for the processor
- Defining the preemption policy for the processor
- Determining the highest priority pending interrupt for the processor.

When enabled, CPU interface takes the highest priority pending interrupt for its connected processor and determines whether the interrupt has sufficient priority for it to signal the interrupt request to the processor.

To determine whether to signal the interrupt request to the processor the CPU interface considers the interrupt priority mask and the preemption settings for the processor. At any time, the connected processor can read the priority of its highest priority active interrupt from a CPU interface register.

The processor acknowledges the interrupt request by reading the CPU interface Interrupt Acknowledge register. The CPU interface returns one of:

The ID number of the highest priority pending interrupt, if that interrupt is of sufficient priority to generate an interrupt exception on the processor. This is the normal response to an interrupt acknowledge.

Exceptionally, an ID number that indicates a spurious interrupt.

When the processor acknowledges the interrupt at the CPU interface, the Distributor changes the status of the interrupt from pending to either active, or active and pending. At this point the CPU interface can signal another interrupt to the processor, to preempt interrupts that are active on the processor. If there is no pending interrupt with sufficient priority for signaling to the processor, the interface de-asserts the interrupt request signal to the processor.

When the interrupt handler on the processor has completed the processing of an interrupt, it writes to the CPU interface to indicate interrupt completion. When this happens, the distributor changes the status of the interrupt either:

- From active to inactive
- From active and pending to pending.

9.3.3 Interrupt handling state machine

The distributor maintains a state machine for each supported interrupt on CPU interface. Following figure shows an instance of this state machine, and the possible state transitions.

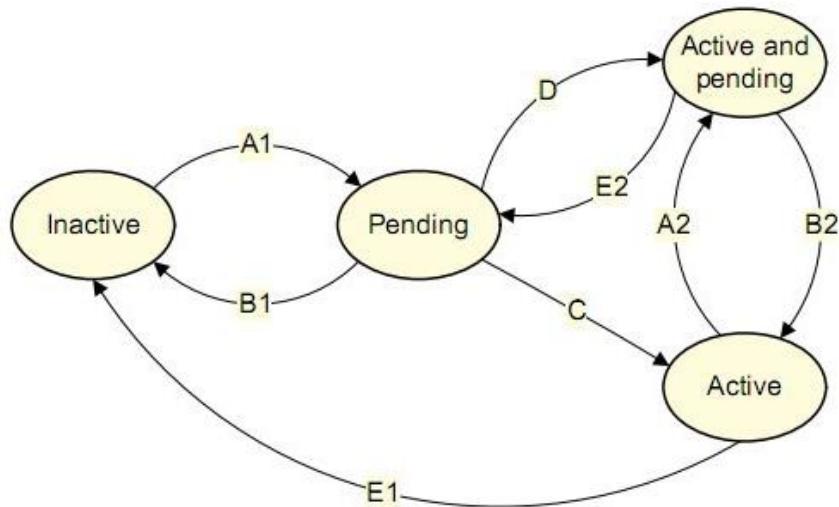


Fig. 9-2 GIC Interrupt handling state machine

Transition A1 or A2, add pending status

For an SGI:

- Occurs on a write to an ICDSGIR that specifies the processor as a target.
- If the GIC implements the Security Extensions and the write to the ICDSGIR is Secure, the transition occurs only if the security configuration of the specified SGI, for the CPU interface, corresponds to the ICDSGIR.SATT bit value.

For an SPI, occurs if either:

- a peripheral asserts an interrupt signal
- software writes to an ICDISPR.

Transition B1 or B2, remove pending status

Not applicable to SGIs:

- a pending SGI must transition through the active state, or reset, to remove its pending status.
- an active and pending SGI must transition through the pending state, or reset, to remove its pending status.

For an SPI, occurs if either:

- the level-sensitive interrupt is pending only because of the assertion of an input signal, and that signal is deasserted
- the interrupt is pending only because of the assertion of an edge-triggered interrupt signal, or a write to an ICDISPR, and software writes to the corresponding ICDICPR.

Transition C

If the interrupt is enabled and of sufficient priority to be signalled to the processor, occurs when software reads from the ICCIAR.

Transition D

For an SGI, occurs if the associated SGI is enabled and the Distributor forwards it to the CPU interface at the same time that the processor reads the ICCIAR to acknowledge a previous instance of the SGI. Whether this transition occurs

depends on the timing of the read of the ICCIAR relative to the reforwarding of the SGI.

For an SPI:

- Occurs if all the following apply:
 - The interrupt is enabled.
 - Software reads from the ICCIAR. This read adds the active state to the interrupt.
 - For a level-sensitive interrupt, the interrupt signal remains asserted. This is usually the case, because the peripheral does not deassert the interrupt until the processor has serviced the interrupt.
- For an edge-triggered interrupt, whether this transition occurs depends on the timing of the read of the ICCIAR relative to the detection of the reassertion of the interrupt. Otherwise the read of the ICCIAR causes transition C, possibly followed by transition A2.

Transition E1 or E2, remove active status

Occurs when software writes to the ICCEOIR.

9.4 Register Description

9.4.1 GIC Distributor interface register summary

Name	Offset	Size	Reset	Description
GICD_ICDDCR	0x000	W	0x0	Distributor Control Register
GICD_ICDICTR	0x004	W		Interrupt Controller Type Register
GICD_ICDIIDR	0x008	W		Distributor Implementer Identification Register
GICD_ICDISR	0x080	W		Interrupt Security Registers
-	-	-	-	reserved
GICD_ICDISER	0x100-0x17C	W		Interrupt Set-Enable Registers
GICD_ICDICER	0x180-0x1FC	W		Interrupt Clear-Enable Registers
GICD_ICDISPR	0x200-0x27C	W	0x0	Interrupt Set-Pending Registers
GICD_ICDICPR	0x280-0x2FC	W	0x0	Interrupt Clear-Pending Registers
GICD_ICDABR	0x300-0x37C	W	0x0	Active Bit Registers
-	-	-	-	reserved
GICD_ICDIPR	0x400-0x7F8	B	0x0	Interrupt Priority Registers
-	-	-	-	reserved
GICD_ICDIPTR	0x800-0x81C	B		Interrupt Processor Targets
-	-	-	-	reserved
GICD_ICDICFR	0xC00-0xCFC	W		Interrupt Configuration Registers
-	-	-	-	Reserved
GICD_ICPPISR		W		PPI Status Register
GICD_ICSPISR	0xD04-0xD1C	W		SPI Status Registers
-	-	-	-	Reserved
GICD_ICDSGIR	0xF00	W		Software Generated Interrupt Register

Notes:

Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** – WORD (32 bits) access

9.4.2 GIC Distributor interface detail register description

GICD_ICDDCR

Address: Operational Base+0x0

Distributor Control Register

Bit	Attr	Reset Value	Description
31:1	-	-	reserved
1	RW	0x0	1'b0: disables all Non-secure interrupts control bits in the distributor from changing state because of any external stimulus change that occurs on the corresponding SPI or PPI signals 1'b1: enables the distributor to update register locations for Non-secure interrupts
0	RW	0x0	1'b0: disables all Secure interrupt control bits in the distributor from changing state because of any external stimulus change that occurs on the corresponding SPI or PPI signals. 1'b1: enables the distributor to update register locations for Secure interrupts.

GICD_ICDICTR

Address: Operational Base+0x4

Interrupt Controller Type Register

Bit	Attr	Reset Value	Description
31:11	-	-	reserved
15:11	R	0x0	Returns the number of Lockable Shared Peripheral Interrupts (LSPIs) that the controller contains. The encoding is: 5'b11111: 31 LSPIs, that are the interrupts of IDs 32-62. When CFGSDISABLE is HIGH, the interrupt controller prevents writes to any register location that controls the operating state of an LSPI.
10	R	0x1	Indicates whether the GIC implements the Security Extensions. 1'b0: Security Extensions not implemented. 1'b1: Security Extensions implemented
9:8	-	-	reserved
7:5	R	0x0	Indicates the number of implemented CPU interface. The number of implemented CPU interface is one more than the value of this field, for example if this field is 3'b011, there are four CPU interface. In this product ,only one CPU interface is implemented.
4:0	R	0x2	Indicates the mAXImum number of interrupts that the GIC supportsa. If the value of this field is N, the mAXImum number of interrupts is 32(N+1). The interrupt ID range is from 0 to one less than the number of IDs. For example:

			5'b00011: Up to 128 interrupt lines, interrupt IDs 0-127. The mAXIum number of interrupts is 1020 (5'b11111).
--	--	--	--

GICD_ICDIIDR

Address: Operational Base+0x8

Distributor Implementer Identification Register

Bit	Attr	Reset Value	Description
31:24	R	0x0	product identifier.
23:20	-	-	reserved
19:16	R	0x0	variant number. Typically, this field is used to distinguish product variants, or major revisions of a product
15:12	R	0x0	revision number. Typically, this field is used to distinguish minor revisions of a product
11:0	R	0x0	Contains the JEP106 code of the company that implemented the GIC Distributor: Bits [11:8]: The JEP106 continuation code of the implementer. Bits [7]: Always 0. Bits [6:0]: The JEP106 identity code of the implementer.

GICD_ICDISR

Address: Operational Base+0x80

Interrupt Security Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For each bit: 1'b0: The corresponding interrupt is Secure. 1'b1: The corresponding interrupt is Non-secure.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDISR number, M, is given by $M = N \text{ DIV } 32$
- the offset of the required ICDISR is $(0x080 + (4*M))$
- the bit number of the required Security status bit in this register is $N \text{ MOD } 32$.

GICD_ICDISER

Address: Operational Base+0x100

Interrupt Set-Enable Registers

Bit	Attr	Reset Value	Description
31:0	RW		For SPIs, for each bit: Reads 1'b0: The corresponding interrupt is disabled. 1'b1: The corresponding interrupt is enabled. Writes 1'b0: Has no effect. 1'b1: Enables the corresponding interrupt. A subsequent read of this bit returns the value 1.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDISER number, M, is given by $M = N \text{ DIV } 32$
- the offset of the required ICDISER is $(0x100 + (4*M))$

- the bit number of the required Set-enable bit in this register is N MOD 32.

GICD_ICDICER

Address: Operational Base+0x180

Interrupt Clear-Enable Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	<p>For SPI, for each bit:</p> <p>Reads</p> <p>1'b0: The corresponding interrupt is disabled. 1'b1: The corresponding interrupt is enabled.</p> <p>Writes</p> <p>1'b0: Has no effect. 1'b1: Disables the corresponding interrupt. A subsequent read of this bit returns the value 0.</p>

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDICER number, M, is given by $M = N \text{ DIV } 32$
- the offset of the required ICDICER is $(0x180 + (4*M))$
- the bit number of the required Clear-enable bit in this register is $N \text{ MOD } 32$.

GICD_ICDISPR

Address: Operational Base+0x200

Interrupt Set-Pending Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	<p>For each bit:</p> <p>Reads</p> <p>1'b0: The corresponding interrupt is not pending on any processor. 1'b1:</p> <ul style="list-style-type: none"> For SGIs, the corresponding interrupt is pending on this processor. For SPIs, the corresponding interrupt is pending on at least one processor. <p>Writes For SPIs:</p> <p>1'b0: Has no effect. 1'b1: The effect depends on whether the interrupt is edge-triggered or level-sensitive:</p> <p>Edge-triggered Changes the status of the corresponding interrupt to:</p> <ul style="list-style-type: none"> pending if it was previously inactive active and pending if it was previously active. Has no effect if the interrupt is already pending. <p>Level sensitive If the corresponding interrupt is not pending, changes the status of the corresponding interrupt to:</p> <ul style="list-style-type: none"> pending if it was previously inactive active and pending if it was previously active.

			If the interrupt is already pending: • because of a write to the ICDISPR, the write has no effect • because the corresponding interrupt signal is asserted, the write has no effect on the status of the interrupt, but the interrupt remains pending if the interrupt signal is deasserted.
--	--	--	--

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDISPR number, M, is given by $M = N \text{ DIV } 32$
- the offset of the required ICDISPR is $(0x200 + (4*M))$
- the bit number of the required Set-pending bit in this register is $N \text{ MOD } 32$.

GICD_ICDICPR

Address: Operational Base+0x280

Interrupt Clear-Pending Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	<p>For each bit:</p> <p>Reads</p> <p>1'b0: The corresponding interrupt is not pending on any processor</p> <p>1'b1:</p> <ul style="list-style-type: none"> • For SGIs, the corresponding interrupt is pending on this processor. • For SPIs, the corresponding interrupt is pending on at least one processor. <p>Writes</p> <p>For SPIs:</p> <p>1'b0: Has no effect.</p> <p>1'b1: The effect depends on whether the interrupt is edge-triggered or level-sensitive:</p> <p>Edge-triggered</p> <p>Changes the status of the corresponding interrupt:</p> <ul style="list-style-type: none"> • inactive if it was previously pending • active if it was previously active and pending. Has no effect if the interrupt is not pending. <p>Level-sensitive</p> <p>If the corresponding interrupt is pending only because of a write to the ICDISPR, the write changes the status of the interrupt to:</p> <ul style="list-style-type: none"> • inactive if it was previously pending • active if it was previously active and pending. Otherwise the interrupt remains pending if the interrupt signal remains asserted.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDICPR number, M, is given by $M = N \text{ DIV } 32$
- the offset of the required ICDICPR is $(0x280 + (4*M))$

- the bit number of the required Set-pending bit in this register is N MOD 32.

GICD_ICDABR

Address: Operational Base+0x300

Active Bit Registers

Bit	Attr	Reset Value	Description
31:0	R		For each bit: 1'b0: Corresponding interrupt is not active. 1'b1: Corresponding interrupt is active.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDABR number, M, is given by $M = N \text{ DIV } 32$
- the offset of the required ICDABR is $(0x300 + (4*M))$
- the bit number of the required Active bit in this register is $N \text{ MOD } 32$.

GICD_ICDIPR

Address: Operational Base+0x400

Interrupt Priority Registers

Bit	Attr	Reset Value	Description
7:0	RW	0x0	The lower the value, the greater the priority of the corresponding interrupt.

For interrupt ID N:

- the corresponding ICDIPR number, M, is given by $M = N$
- the offset of the required ICDIPR is $(0x400 + M)$

GICD_ICDIPTR

Address: Operational Base+0x800

Interrupt Processor Targets Registers

Bit	Attr	Reset Value	Description
7:0	RW	0x1	This register is not used. As in our product ,there is only one processor.

GICD_ICDICFR

Address: Operational Base+0xc00

Interrupt Configuration Registers

Bit	Attr	Reset Value	Description
2F+1	RW	0x0	$F=0,1,2,3....15$ The encoding is: 1'b0: Corresponding interrupt is level-sensitive. 1'b1: Corresponding interrupt is edge-triggered.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDICFR number, M, is given by $M = N \text{ DIV } 16$
- the offset of the required ICDIPTR is $(0xC00 + (4*M))$
- the required Priority field in this register, F, is given by $F = N \text{ MOD } 16$, where field 0 refers to register bits [1:0], field 1 refers to bits [3:2], and so on, up to field 15 refers to bits [31:30]

GICD_ICDSGIR

Address: Operational Base+0xf00

Software Generated Interrupt Register

Bit	Attr	Reset Value	Description

31:26	-	-	reserved
25:24	W	0x0	2'b00: Send the interrupt to the CPU interface specified in the CPUTargetListfield. 2'b01: Send the interrupt to all CPU interface except the CPU interface that requested the interrupt. 2'b10: Send the interrupt only to the CPU interface that requested the interrupt. 2'b11: Reserved
23:16	W	0x0	When TargetList Filter = 2'bb00, defines the CPU interface the Distributor must send the interrupt to. Each bit of CPUTargetList[7:0] refers to the corresponding CPU interface, for example CPUTargetList[0] corresponds to CPU interface 0. Setting a bit to 1 sends the interrupt to the corresponding interface.
15	W	0x0	If the GIC implements the Security Extensions, this field is writable only using a Secure access. Any Non-secure write to the ICDSGIR issues an SGI only if the specified SGI is programmed as Non-secure, regardless of the value of bit [15] of the write. Specifies the required security value of the SGI: 1'b0: Send the SGI specified in the SGIIINTID field to a specified CPU interface only if the SGI is configured as Secure on that interface. 1'b1: Send the SGI specified in the SGIIINTID field to a specified CPU interface only if the SGI is configured as Non-secure on that interface
14:4	-	-	reserved
3:0	W	0x0	The Interrupt ID of the SGI to send to the specified CPU interface. The value of this field is the Interrupt ID, in the range 0-15, for example a value of 4'b0011 specifies Interrupt ID 3

9.4.3 GIC CPU interface register summary

Name	Offset	Size	Reset Value	Description
GICC_ICCICR	0x00	W	0x0	CPU Interface Control Register
GICC_ICCPMR	0x04	W	0x0	Interrupt Priority Mask Register
GICC_ICCBPR	0x08	W	0x0	Binary Point Register
GICC_ICCIAR	0x0C	W	0x3ff	Interrupt Acknowledge Register
GICC_ICCEOIR	0x10	W	-	End of Interrupt Register
GICC_ICCRPR	0x14	W	0xff	Running Priority Register
GICC_ICCHPIR	0x18	W	0x3ff	Highest Pending Interrupt Register
GICC_ICCABPR	0x1C	W	0x0	Aliased Binary Point

				Register
GICC_ICCIIDR	0xFC	W	0x0	CPU Interface Identification Register

Notes:

Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** – WORD (32 bits) access

9.4.4 GIC CPU interface detail register description

GICC_ICCICR

Address: Operational Base+0x0

CPU Interface Control Register

Bit	Attr	Reset Value	Description
31:5	-	-	reserved
4	RW	0x0	<p>Controls whether the CPU interface uses the Secure or Non-secure Binary Point Register for preemption.</p> <p>1'b0: To determine any preemption, use: • the Secure Binary Point Register for Secure interrupts • the Non-secure Binary Point Register for Non-secure interrupts.</p> <p>1'b1: To determine any preemption use the Secure Binary Point Register for both Secure and Non-secure interrupts.</p>
3	RW	0x0	<p>Controls whether the GIC signals Secure interrupts to a target processor using the FIQ or the IRQ signal.</p> <p>1'b0: Signal Secure interrupts using the IRQ signal. 1'b1: Signal Secure interrupts using the FIQ signal. The GIC always signals Non-secure interrupts using the IRQ signal.</p>
2	RW	0x0	<p>Controls whether a Secure read of the ICCIAR, when the highest priority pending interrupt is Non-secure, causes the CPU interface to acknowledge the interrupt.</p> <p>1'b0: If the highest priority pending interrupt is Non-secure, a Secure read of the ICCIAR returns an Interrupt ID of 1022. The read does not acknowledge the interrupt, and the pending status of the interrupt is unchanged.</p> <p>1'b1: If the highest priority pending interrupt is Non-secure, a Secure read of the ICCIAR returns the Interrupt ID of the Non-secure interrupt. The read acknowledges the interrupt, and the status of the interrupt becomes active, or active and pending.</p>

1	RW	0x0	An alias of the Enable bit in the Non-secure ICCICR. This alias bit means Secure software can enable the signalling of Non-secure interrupts. 1'b0: Disable signalling of Non-secure interrupts. 1'b1: Enable signalling of Non-secure interrupts
0	RW	0x0	Global enable for the signalling of Secure interrupts by the CPU interface to the connected processors. 1'b0: Disable signalling of Secure interrupts. 1'b1: Enable signalling of Secure interrupts

GICC_ICCPMR

Address: Operational Base+0x4

Interrupt Priority Mask Register

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	RW	0x0	The priority mask level for the CPU interface. If the priority of an interrupt is higher than the value indicated by this field, the interface signals the interrupt to the processor. If the GIC supports fewer than 256 priority levels then some bits are read as zero(RAZ)/write ignore(WI), as follows: 128 supported levels Bit [0] = 1'b0. 64 supported levels Bit [1:0] = 2'b00. 32 supported levels Bit [2:0] = 3'b000. 16 supported levels Bit [3:0] = 4'b0000

GICC_ICCBPR

Address: Operational Base+0x8

Binary Point Register

Bit	Attr	Reset Value	Description
31:3	-	-	reserved
2:0	RW	0x0	The value of this field controls how the 8-bit interrupt priority field is split into a group priority field, used to determine interrupt preemption, and a subpriority field.

GICC_ICCIAR

Address: Operational Base+0xc

Interrupt Acknowledge Register

Bit	Attr	Reset Value	Description
31:13	-	-	reserved
12:10	RO	0x0	For SGIs in a multiprocessor implementation, this field identifies the processor that requested the interrupt. It returns the number of the CPU interface that made the request, for example a value of 3 (3'b011) means the request was generated by a write to the IDCSFGIR on CPU interface 3. For all other interrupts this field is RAZ.

9:0	RO	0x0	The interrupt ID.
-----	----	-----	-------------------

GICC_ICCEOIR

Address: Operational Base+0x10

End of Interrupt Register

Bit	Attr	Reset Value	Description
31:13	-	-	reserved
12:10	WO	0x0	On a multiprocessor implementation, on completion of the processing of an SGI, this field contains the CPUID value from the corresponding ICCIAR access.
9:0	WO	0x0	The ACKINTID value from the corresponding ICCIAR access.

GICC_ICCRPR

Address: Operational Base+0x14

Running Priority Register

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	RO	0x0	The priority value of the highest priority interrupt that is active on the CPU interface.

GICC_ICCABPR

Address: Operational Base+0x18

Aliased Binary Point Register

Bit	Attr	Reset Value	Description
31:3	-	-	reserved
2:0	RW	0x0	Provides an alias of the Non-secure ICCBPR.

GICC_ICCHPIR

Address: Operational Base+0x1c

Highest Pending Interrupt Register

Bit	Attr	Reset Value	Description
31:10	-	-	reserved
9:0	R	0x0	The interrupt ID of the highest priority pending interrupt.

GICC_ICCIIDR

Address: Operational Base+0xfc

CPU Interface Identification Register

Bit	Attr	Reset Value	Description
31:20	R	0x390	An IMPLEMENTATION DEFINED product identifier.
19:16	R	0x1	For an implementation that complies with this specification, the value is 0x1
15:12	R	0x2	An IMPLEMENTATION DEFINED revision number for the CPU interface.
11:0	R	0x43B	Contains the JEP106 code of the company that implemented the GIC CPU interface: b Bits [11:8]: The JEP106 continuation code of the implementer. Bit [7]: Always 0.

		Bits [6:0]: The JEP106 identity code of the implementer.
--	--	--

9.5 Interface Description

Both distributor interface and CPU interface are secure accessed only after reset.

When the signal cfgsdisable is HIGH, it enhances the security of the GIC by preventing write accesses to security-critical configuration registers. This signal is low after reset, it can be configured through TZPC registers.

9.6 Application Notes

9.6.1 General handling of interrupts

The GIC operates on interrupts as follows:

1. The GIC determines whether each interrupt is enabled. An interrupt that is not enabled has no further effect on the GIC.(Enables an interrupt by writing to the appropriate ICDISER bit, disables an interrupt by writing to the appropriate ICDICER bit)
2. For each enabled interrupt that is pending, the Distributor determines the targeted processor.
3. For processor, the Distributor determines the highest priority pending interrupt, based on the priority information it holds for each interrupt, and forwards the interrupt to the CPU interface.
4. The CPU interface compares the interrupt priority with the current interrupt priority for the processor, determined by a combination of the Priority Mask Register, the current preemption settings, and the highest priority active interrupt for the processor. If the interrupt has sufficient priority, the GIC signals an interrupt exception request to the processor.
5. When the processor takes the interrupt exception, it reads the ICCIAR in its CPU interface to acknowledge the interrupt. This read returns an Interrupt ID that the processor uses to select the correct interrupt handler. When it recognizes this read, the GIC changes the state of the interrupt:
 - If the pending state of the interrupt persists when the interrupt becomes active, or if the interrupt is generated again, from pending to active and pending.
 - Otherwise, from pending to active
6. When the processor has completed handling the interrupt, it signals this completion by writing to the ICCEOIR in the GIC

Generating an SGI

A processor generates an SGI by writing to an ICDSGIR.

9.6.2 Interrupt prioritization

Software configures interrupt prioritization in the GIC by assigning a priority value to each interrupt source. Priority values are 8-bit unsigned binary.

In this product, GIC implements 64 priority levels. So only the highest 6 bits are valid, the lower 2 bits read as zero.

In the GIC prioritization scheme, lower numbers have higher priority, that is, the lower the assigned priority value the higher the priority of the interrupt. The highest interrupt priority always has priority field value 0.

The ICDIPRs hold the priority value for each supported interrupt. To determine the number of priority bits implemented write 0xFF to an ICDIPR priority field and read back the value stored.

Preemption

A CPU interface supports forwarding of higher priority pending interrupts to a target processor before an active interrupt completes. A pending interrupt is only forwarded if it has a higher priority than all of:

- the priority of the highest priority active interrupt on the target processor, the running priority for the processor, see Running Priority Register (ICCRPR) .
- The priority mask, see Priority masking.
- The priority group, see Priority grouping.

Preemption occurs at the time when the processor acknowledges the new interrupt, and starts to service it in preference to the previously active interrupt or the currently running process. When this occurs, the initial active interrupt is said to have been preempted. Starting to service an interrupt while another interrupt is still active is sometimes described as interrupt nesting.

Priority masking

The ICCPMR for a CPU interface defines a priority threshold for the target processor, see Interrupt Priority Mask Register. The GIC only signals pending interrupts with a higher priority than this threshold value to the target processor. A value of zero, the register reset value, masks all interrupts to the associated processor.

The GIC always masks an interrupt that has the largest supported priority field value. This provides an additional means of preventing an interrupt being signalled to any processor.

Priority grouping

Priority grouping splits each priority value into two fields, the group priority and the subpriority fields. The GIC uses the group priority field to determine whether a pending interrupt has sufficient priority to preempt a currently active interrupt.

The binary point field in the ICCBPR controls the split of the priority bits into the two parts. This 3-bit field specifies how many of the least significant bits of the 8-bit interrupt priority field are excluded from the group priority field, as following table shows.

Binary point value	Group priority field	Subpriority field	Field with binary point
0	[7:1]	[0]	ggggggg.s
1	[7:2]	[1:0]	gggggg.ss
2	[7:3]	[2:0]	gggggsss
3	[7:4]	[3:0]	gggg.ssss
4	[7:5]	[4:0]	gg.ssssss
5	[7:6]	[5:0]	g.sssssss
6	[7]	[6:0]	.ssssssss
7	No preemption	[7:0]	

Where multiple pending interrupts share the same group priority, the GIC uses the subpriority field to resolve the priority within a group.

9.6.3 The effect of the Security Extensions on interrupt handling

If a GIC CPU interface implements the Security Extensions, it provides two interrupt output signals, IRQ and FIQ:

- The CPU interface always uses the IRQ exception request for Non-secure interrupts
- Software can configure the CPU interface to use either IRQ or FIQ exception requests for Secure interrupts.

Security Extensions support

Software can detect support for the Security Extensions by reading the ICDICTR.SecurityExtn bit, see Interrupt Controller Type Register (ICDICTR).

Secure software makes Secure writes to the ICDISRs to configure each interrupt as Secure or Non-secure, see Interrupt Security Registers (ICDISRn).

In addition:

- The banking of registers provides independent control of Secure and Non-secure interrupts.
- The Secure copy of the ICCICR has additional fields to control the processing of Secure and Non-secure interrupts, see CPU Interface Control Register (ICCICR) These fields are:
 - the SBPR bit, that affects the preemption of Non-secure interrupts.
 - the FIQEn bit, that controls whether the interface signals Secure interrupt to the processor using the IRQ or FIQ interrupt exception requests.
 - the AckCtl bit, that affects the acknowledgment of Non-secure interrupts.
 - the EnableNS bit, that controls whether Non-secure interrupts are signaled to the processor, and is an alias of the Enable bit in the Non-secure ICCICR.
- The Non-secure copy of the ICCBPR is aliased as the ICCABPR, see Aliased Binary Point Register (ICCABPR). This is a Secure register, meaning it is only accessible by Secure accesses.

Effect of the Security Extensions on interrupt acknowledgement

When a processor takes an interrupt, it acknowledges the interrupt by reading the ICCIAR. A read of the ICCIAR always acknowledges the highest priority pending interrupt for the processor performing the read.

If the highest priority pending interrupt is a Secure interrupt, the processor must make a Secure read of the ICCIAR to acknowledge it.

By default, the processor must make a Non-secure read of the ICCIAR to acknowledge a Non-secure interrupt. If the AckCtl bit in the Secure ICCICR is set to 1 the processor can make a Secure read of the ICCIAR to acknowledge a Non-secure interrupt.

If the read of the ICCIAR does not match the security of the interrupt, taking account of the AckCtl bit value for a Non-secure interrupt, the ICCIAR read does

not acknowledge any interrupt and returns the value:

- 1022 for a Secure read when the highest priority interrupt is Non-secure
- 1023 for a Non-secure read when the highest priority interrupt is Secure.

Rockchip Confidential

Chapter 10 DMC (Dynamic Memory Interface)

10.1 Overview

The DMC includes two section: dynamic ram protocol controller(PCTL) and phy controller.

The PCTL SoC application bus interface supports a lowest-latency native application interface (NIF). To maximize data transfer efficiency, NIF commands transfer data without flow control. To simplify command processing, the NIF accepts addresses in rank, bank, row, column format.

The DDR PHY provides control features to ease the customer implementation of digitally controlled features of the PHY such as initialization, DQS gate training, and programmable configuration controls. The DDR PHY has built-in self test features to provide support for production testing of the compatible PHY. It also provides a DFI 2.1 interface to the PHY.

The DMC supports the following features:

- Complete, integrated DDR3, LPDDR2 solution
- DFI 2.1 interface compatibility
- Up to 800 Mbps in 1:1 frequency ratio, using a 400MHz controller clock and 400MHz memory clock.
- Support for x16, x32 DDR3 memories, for a total memory data path width of 32 bits
- Support for x32 LPDDR2 memories, for a total memory data path width of 32 bits
- Up to 2 memory ranks; devices within a rank tie to a common chip select
- Up to 8 open memory banks, maximum of eight per rank
- Per-NIF transaction controllable bank management policies: open-page, close-page
- Low area, low power architecture with minimal buffering on the data, avoiding duplication of storage resources within the system
- PCTL NIF slave interface facilitates easy integration with an external scheduler or standard on-chip buses
- Efficient DDR protocol implementation with in-order column (Read and Write) commands and out-of-order Activate and Precharge commands
- Three clock cycles best case command latency (best case is when a command is to an open page and the shift array in the PCTL is empty)
- 1T or 2T memory command timing
- Automatic power-down and self-refresh entry and exit
- Software and hardware driven self-refresh entry and exit
- Programmable memory initialization
- Partial population of memories, where not all DDR byte lanes are populated with memory chips

- Programmable per rank memory ODT (On-Die Termination) support for reads and writes
- APB interface for controller software-accessible registers
- Programmable data training interface:
 - Assists in training of the data eye of the memory channel
 - Provides a method for testing large sections of memory
- Automatic DQS gate training
- At-speed built-in-self-test (BIST) loopback testing on both the address and data channels for DDR PHYs
- PHY control and configuration registers
- Optional, additional JTAG interface to configure registers

10.2 Block Diagram

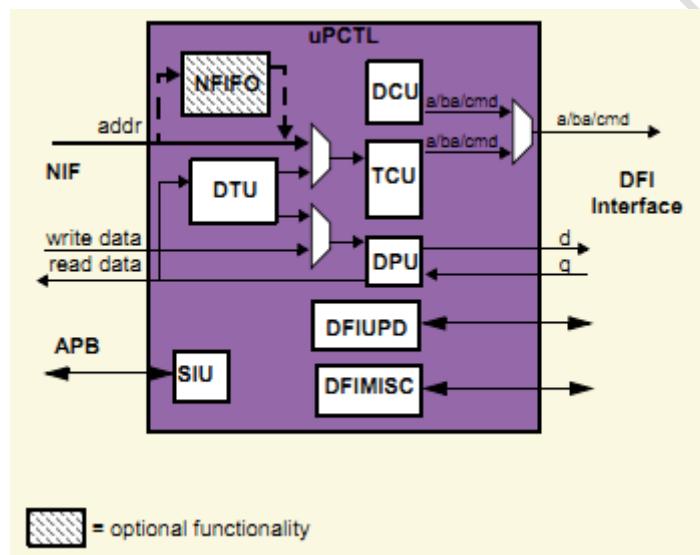


Fig. 11-1 Protocol controller architecture

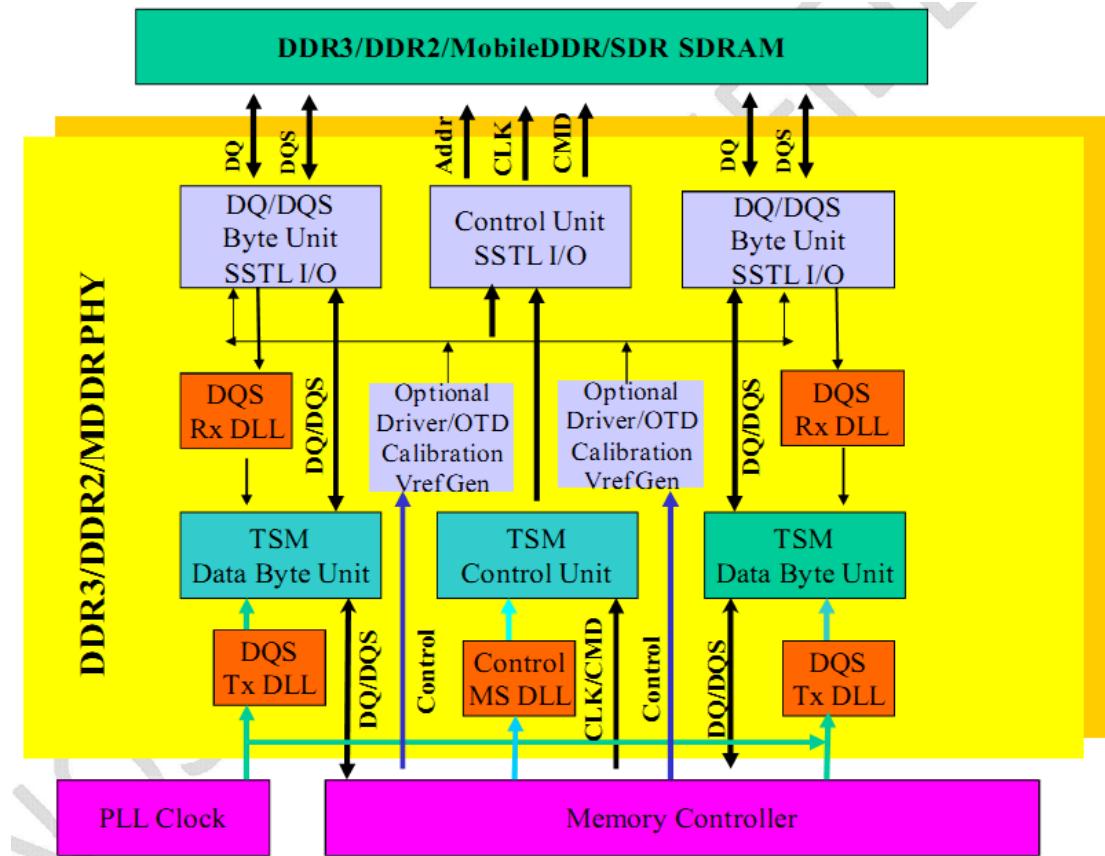


Fig. 10-2 PHY controller architecture

10.3 Function Description

10.3.1 Protocol controller(PCTL)

PCTL operations are defined in terms of the current state of the Operational State Machine. Software can move PCTL in any of the operational states by issuing commands via the SCTL register. Transitions from one operational state to the other occur pass through a “transitional” state. Transitional states are exited automatically by the PCTL after all the necessary actions required to change operational state have been completed. The current operational state of PCTL is reported by the STAT register and is also available from the p_ctl_stat output.

PCTL supports the following operational states:

- Init_mem - This state is the default state entered after reset. All writable registers can be programmed. While in this state software can program PCTL and initialize the PHY and the memories. The memories are not refreshed and data that has previously been written to the memories may be lost as a result. The Init_mem state is also used when it is desirable to stop any automatic PCTL function that directly affects the memories, like Power Down and Refresh, or when a software reset of the memory subsystem has to be executed.
- Config - This state is used to suspend temporarily the normal NIF traffic and allow software to reprogram PCTL and memories if necessary, while still keeping active the periodic generation of Refresh cycles to the memories.

Power Down entry and exit sequences are possible while in Config state.

- Access - This is the operational state where NIF transactions are accepted by the PCTL and converted into memory read and writes. None of the registers can be programmed except SCFG, SCTL, ECCCLR and DTU* registers.
- Low_power - Memories are in self refresh mode. The PCTL does not generate refresh cycles while in this state.

Access and Low_power states can also be entered and exited by the hardware low power signals (c_*). In case of conflicting software and hardware low-power commands, the resulting operational state taken by the controller can be either one of the two conflicting requests.

Figure 10-3 illustrates the operational and transitional states.

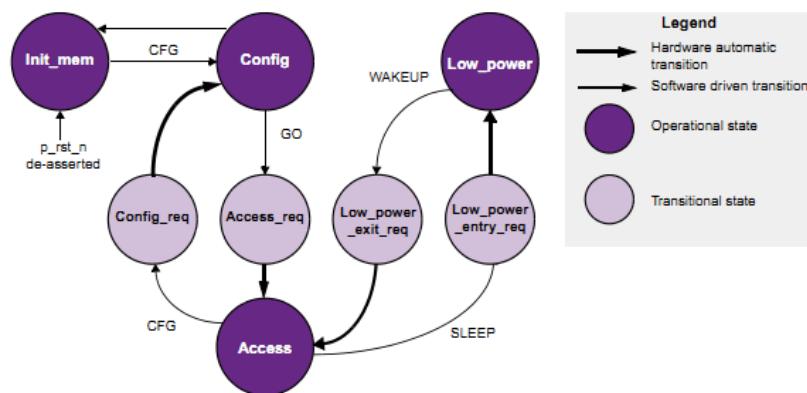


Fig. 11-3 Protocol controller architecture

The controller clock is the same clock driving the memory controller and will be the same frequency as the SDRAM clock (ck). The configuration clock can run at a frequency equal to or less than the controller clock. The configuration clock drives all non-DDR timing logic, such as configuration registers, PHY initialization, output impedance, and so on.

10.3.2 DDR PHY

DDR PHY provides turn key physical interface solutions for chip requiring access to DDR3 SDRAM device. It is optimized for low power and high speed (up to 800Mbps for DDR3 and LPDDR2) applications with robust timing and small silicon area. It supports DDR3 and LPDDR2 SDRAM components in the market. The PHY components contain DDR specialized functional and utility SSTL I/Os up to 800MHz, critical timing synchronization module (TSM) and a low power/jitter DLLs with programmable fine-grain control for any SDRAM interface.

DDR PHY uses a DFI digital interface to connect the memory controller. All interface timing is in 1X SDR clock domain. The controller to PHY interface is running at single data rate (SDR) therefore read/write bus is double width. DDR muxing is done in the PHY block together with all related per-byte lane timing adjustment. The interface is fairly generic and support high performance input and output data flow gearing toward 100Mbps to 800Mbps DDR3 and LPDDR2 SDRAM speed in wide range.

With configurable timing and driving strength and ODT parameters to interface to the wide variety of SDRAMs, the PHY is very flexible with advanced command capability to increase SDRAM operation efficiency.

10.4 Register Description

10.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DDR_PCTL_SCFG	0x0000	W	0x00000300	State Configuration Register
DDR_PCTL_SCTL	0x0004	W	0x00000000	Operational State Control Register
DDR_PCTL_STAT	0x0008	W	0x00000000	Operational State Status Register
DDR_PCTL_INTRSTAT	0x000c	W	0x00000000	Interrupt Status Register
DDR_PCTL_MCMD	0x0040	W	0x00100000	Memory Command Register
DDR_PCTL_POWCTL	0x0044	W	0x00000000	Power Up Control Register
DDR_PCTL_POWSTAT	0x0048	W	0x00000000	Power Up Status Register
DDR_PCTL_CMDTSTA T	0x004c	W	0x00000000	Command Timers Status Register
DDR_PCTL_CMDTSTA TEN	0x0050	W	0x00000000	Command Timers Status Enable Register
DDR_PCTL_MRRCFG0	0x0060	W	0x00000000	Mode Register Read Configuration 0
DDR_PCTL_MRRSTAT 0	0x0064	W	0x00000000	Mode Register Read Status 0 Register
DDR_PCTL_MRRSTAT 1	0x0068	W	0x00000000	Mode Register Read Status 1 Register
DDR_PCTL_MCFG	0x0080	W	0x00040020	Memory Configuration Register
DDR_PCTL_PPCFG	0x0084	W	0x00000000	Partially Populated Memories Configuration Register
DDR_PCTL_MSTAT	0x0088	W	0x00000000	Memory Status Register
DDR_PCTL_MCFG1	0x0090	W	0x00000000	Memory Configuration 1 Register
DDR_PCTL_DTUPDES	0x0094	W	0x00000000	DTU Status Register
DDR_PCTL_DTUNA	0x0098	W	0x00000000	DTU Number of Addresses Created Register
DDR_PCTL_DTUNE	0x009c	W	0x00000000	DTU Number of Errors Register
DDR_PCTL_DTUPRD0	0x00a0	W	0x00000000	DTU Parallel Read 0 Register
DDR_PCTL_DTUPRD1	0x00a4	W	0x00000000	DTU Parallel Read 1 Register
DDR_PCTL_DTUPRD2	0x00a8	W	0x00000000	DTU Parallel Read 2 Register
DDR_PCTL_DTUPRD3	0x00ac	W	0x00000000	DTU Parallel Read 3 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DTUAWDT	0x00b0	W	0x00000290	DTU Address Width Register
DDR_PCTL_TOGCNT1_U	0x00c0	W	0x00000064	Toggle Counter 1us Register
DDR_PCTL_TINIT	0x00c4	W	0x000000c8	t_init Timing Register
DDR_PCTL_TRSTH	0x00c8	W	0x00000000	t_rsth Timing Register
DDR_PCTL_TOGCNT1_00N	0x00cc	W	0x00000001	Toggle Counter 100ns
DDR_PCTL_TREFI	0x00d0	W	0x00000001	t_refi Timing Register
DDR_PCTL_TMRD	0x00d4	W	0x00000001	t_mrd Timing Register
DDR_PCTL_TRFC	0x00d8	W	0x00000001	t_rfc Timing Register
DDR_PCTL_TRP	0x00dc	W	0x00010006	t_trp Timing Register
DDR_PCTL_TRTW	0x00e0	W	0x00000002	t_rtw Timing Register
DDR_PCTL_TAL	0x00e4	W	0x00000000	AL Register
DDR_PCTL_TCL	0x00e8	W	0x00000004	CL Timing Register
DDR_PCTL_TCWL	0x00ec	W	0x00000003	CWL Timing Register
DDR_PCTL_TRAS	0x00f0	W	0x00000010	t_ras Timing Register
DDR_PCTL_TRC	0x00f4	W	0x00000016	t_rc Timing Register
DDR_PCTL_TRCD	0x00f8	W	0x00000006	t_rcd Timing Register
DDR_PCTL_TRRD	0x00fc	W	0x00000004	t_rrd Timing Register
DDR_PCTL_TRTP	0x0100	W	0x00000003	t_rtp Timing Register
DDR_PCTL_TWR	0x0104	W	0x00000006	t_wr Register
DDR_PCTL_TWTR	0x0108	W	0x00000004	t_wtr Timing Register
DDR_PCTL_TEXSR	0x010c	W	0x00000001	t_exsr Timing Register
DDR_PCTL_TXP	0x0110	W	0x00000001	t_xp Timing Register
DDR_PCTL_TXPDLL	0x0114	W	0x00000000	t_xpdll Timing Register
DDR_PCTL_TZQCS	0x0118	W	0x00000000	t_zqcs Timing Register
DDR_PCTL_TZQCSI	0x011c	W	0x00000000	t_zqcsi Timing Register
DDR_PCTL_TDQS	0x0120	W	0x00000001	t_dqs Timing Register
DDR_PCTL_TCKSRE	0x0124	W	0x00000000	t_cksrc Timing Register
DDR_PCTL_TCKSRX	0x0128	W	0x00000000	t_cksrcx Timing Register
DDR_PCTL_TCKE	0x012c	W	0x00000003	t_cke Timing Register
DDR_PCTL_TMOD	0x0130	W	0x00000000	t_mod Timing Register
DDR_PCTL_TRSTL	0x0134	W	0x00000000	Reset Low Timing Register
DDR_PCTL_TZQCL	0x0138	W	0x00000000	t_zqcl Timing Register
DDR_PCTL_TMRR	0x013c	W	0x00000002	t_mrr Timing Register
DDR_PCTL_TCRESR	0x0140	W	0x00000004	t_ckresr Timing Register
DDR_PCTL_DTUWACT_L	0x0200	W	0x00000000	DTU Write Address Control
DDR_PCTL_DTURACT_L	0x0204	W	0x00000000	DTU Read Address Control Register
DDR_PCTL_DTUCFG	0x0208	W	0x00000000	DTU Configuration Control Register
DDR_PCTL_DTUECTL	0x020c	W	0x00000000	DTU Execute Control Register
DDR_PCTL_DTUWD0	0x0210	W	0x00000000	DTU Write Data #0 Register
DDR_PCTL_DTUWD1	0x0214	W	0x00000000	DTU Write Data #1 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DTUWD2	0x0218	W	0x00000000	DTU Write Data #2 Register
DDR_PCTL_DTUWD3	0x021c	W	0x00000000	DTU Write Data #3 Register
DDR_PCTL_DTUWDM	0x0220	W	0x00000000	DTU Write Data Mask Register
DDR_PCTL_DTURD0	0x0224	W	0x00000000	DTU Read Data #0 Register
DDR_PCTL_DTURD1	0x0228	W	0x00000000	DTU Read Data #1 Register
DDR_PCTL_DTURD2	0x022c	W	0x00000000	DTU Read Data #2 Register
DDR_PCTL_DTURD3	0x0230	W	0x00000000	DTU Read Data #3 Register
DDR_PCTL_DTULFSRWD	0x0234	W	0x00000000	DTU LFSR Seed for Write Data Generation Register
DDR_PCTL_DTULFSRRD	0x0238	W	0x00000000	DTU LFSR Seed for Read Data Generation Register
DDR_PCTL_DTUREAF	0x023c	W	0x00000000	DTU Error Address FIFO Register
DDR_PCTL_DFITCTRL_DELAY	0x0240	W	0x00000002	DFI tctrl_delay Register
DDR_PCTL_DFIODTCFG	0x0244	W	0x00000000	DFI ODT Configuration
DDR_PCTL_DFIODTCFG1	0x0248	W	0x06060000	DFI ODT Timing Configuration 1 (for Latency and Length)
DDR_PCTL_DFIODTRA_NKMAP	0x024c	W	0x00008421	DFI ODT Rank Mapping
DDR_PCTL_DFITPHY_WRDATA	0x0250	W	0x00000001	DFI tphy_wrdata Register
DDR_PCTL_DFITPHY_WRLAT	0x0254	W	0x00000001	DFI tphy_wrlat Register
DDR_PCTL_DFITRDDATAEN	0x0260	W	0x00000001	DFI trddata_en Register
DDR_PCTL_DFITPHYR_DLAT	0x0264	W	0x0000000f	DFI tphy_rdlat Register
DDR_PCTL_DFITPHYU_PDTYPE0	0x0270	W	0x00000010	DFI tphyupd_type0 Register
DDR_PCTL_DFITPHYU_PDTYPE1	0x0274	W	0x00000010	DFI tphyupd_type1 Register
DDR_PCTL_DFITPHYU_PDTYPE2	0x0278	W	0x00000010	DFI tphyupd_type2 Register
DDR_PCTL_DFITPHYU_PDTYPE3	0x027c	W	0x00000010	DFI tphyupd_type3 Register
DDR_PCTL_DFITCTRL_UPDMIN	0x0280	W	0x00000010	DFI tctrlupd_min Register
DDR_PCTL_DFITCTRL_UPDMAX	0x0284	W	0x00000040	DFI tctrlupd_max Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFITCTRL_UPDDLY	0x0288	W	0x00000008	DFI tctrlupddly Register
DDR_PCTL_DFIUPDCF_G	0x0290	W	0x00000003	DFI Update Configuration Register
DDR_PCTL_DFITREFMSKI	0x0294	W	0x00000000	DFI Masked Refresh Interval
DDR_PCTL_DFITCTRL_UPDI	0x0298	W	0x00000000	DFI tctrlupd_interval Register
DDR_PCTL_DFITRCFG_0	0x02ac	W	0x00000000	DFI Training Configuration 0 Register
DDR_PCTL_DFITRSTA_T0	0x02b0	W	0x00000000	DFI Training Status 0 Register
DDR_PCTL_DFITRWRL_VLEN	0x02b4	W	0x00000000	DFI Training dfi_wrlvl_en Register
DDR_PCTL_DFITRRDL_VLEN	0x02b8	W	0x00000000	DFI Training dfi_rdlvl_en Register
DDR_PCTL_DFITRRDL_VLGATEEN	0x02bc	W	0x00000000	DFI Training dfi_rdlvl_gate_en Register
DDR_PCTL_DFISTSTA_T0	0x02c0	W	0x00000000	DFI Status Status 0 Register
DDR_PCTL_DFISTCFG_0	0x02c4	W	0x00000000	DFI Status Configuration 0 Register
DDR_PCTL_DFISTCFG_1	0x02c8	W	0x00000000	DFI Status Configuration 1 Register
DDR_PCTL_DFITDRA_MCLKEN	0x02d0	W	0x00000002	DFI t dram_clk_enable Register
DDR_PCTL_DFITDRA_MCLKDIS	0x02d4	W	0x00000002	DFI t dram_clk_disable Register
DDR_PCTL_DFISTCFG_2	0x02d8	W	0x00000000	DFI Status Configuration 2 Register
DDR_PCTL_DFISTPAR_CLR	0x02dc	W	0x00000000	DFI Status Parity Clear Register
DDR_PCTL_DFISTPAR_LOG	0x02e0	W	0x00000000	DFI Status Parity Log Register
DDR_PCTL_DFILPCFG_0	0x02f0	W	0x00070000	DFI Low Power Configuration 0 Register
DDR_PCTL_DFITRWRL_VLRESP0	0x0300	W	0x00000000	DFI Training dfi_wrlvl_resp Status 0 Register
DDR_PCTL_DFITRWRL_VLRESP1	0x0304	W	0x00000000	DFI Training dfi_wrlvl_resp Status 1 Register
DDR_PCTL_DFITRWRL_VLRESP2	0x0308	W	0x00000000	DFI Training dfi_wrlvl_resp Status 2 Register
DDR_PCTL_DFITRRDL_VLRESP0	0x030c	W	0x00000000	DFI Training dfi_rdlvl_resp Status 0 Register
DDR_PCTL_DFITRRDL_VLRESP1	0x0310	W	0x00000000	DFI Training dfi_rdlvl_resp Status 1 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFITRRDL_VLRESP2	0x0314	W	0x00000000	DFI Training dfi_rdlvl_resp Status 2 Register
DDR_PCTL_DFITRWRL_VLDELAY0	0x0318	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 0 Register
DDR_PCTL_DFITRWRL_VLDELAY1	0x031c	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 1 Register
DDR_PCTL_DFITRWRL_VLDELAY2	0x0320	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 2 Register
DDR_PCTL_DFITRRDL_VLDELAY0	0x0324	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 0 Register
DDR_PCTL_DFITRRDL_VLDELAY1	0x0328	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 1 Register
DDR_PCTL_DFITRRDL_VLDELAY2	0x032c	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 2 Register
DDR_PCTL_DFITRRDL_VLGATEDELAY0	0x0330	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 0
DDR_PCTL_DFITRRDL_VLGATEDELAY1	0x0334	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 1
DDR_PCTL_DFITRRDL_VLGATEDELAY2	0x0338	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 2
DDR_PCTL_DFITRCMD	0x033c	W	0x00000000	DFI Training Command Register
DDR_PCTL_IPVR	0x03f8	W	0x00000000	IP Version Register
DDR_PCTL_IPTR	0x03fc	W	0x44574300	IP Type Register

Name	Offset	Size	Reset Value	Description
DDRPHY_REG00	0x0000	W	0x000000ff	DDR PHY register 00
DDRPHY_REG01	0x0004	W	0x00000004	DDR PHY register 01
DDRPHY_REG02	0x0008	W	0x00000000	DDR PHY register 02
DDRPHY_REG03	0x000c	W	0x00000022	DDR PHY register 03
DDRPHY_REG04	0x0010	W	0x00000000	DDR PHY register 04
DDRPHY_REG0B	0x002c	W	0x00000060	DDR PHY register 0B
DDRPHY_REG0C	0x0030	W	0x00000000	DDR PHY register 0C
DDRPHY_REG11	0x0044	W	0x000000aa	DDR PHY register 11
DDRPHY_REG12	0x0048	W	0x00000002	DDR PHY register 12
DDRPHY_REG13	0x004c	W	0x0000000c	DDR PHY register 13
DDRPHY_REG14	0x0050	W	0x00000000	DDR PHY register 14
DDRPHY_REG16	0x0058	W	0x000000aa	DDR PHY register 16
DDRPHY_REG20	0x0080	W	0x000000aa	DDR PHY register 20
DDRPHY_REG21	0x0084	W	0x000000aa	DDR PHY register 21
DDRPHY_REG26	0x0098	W	0x0000000c	DDR PHY register 26

Name	Offset	Size	Reset Value	Description
DDRPHY_REG27	0x009c	W	0x00000000	DDR PHY register 27
DDRPHY_REG28	0x00a0	W	0x00000001	DDR PHY register 28
DDRPHY_REG30	0x00c0	W	0x000000aa	DDR PHY register 30
DDRPHY_REG31	0x00c4	W	0x000000aa	DDR PHY register 31
DDRPHY_REG36	0x00d4	W	0x0000000c	DDR PHY register 36
DDRPHY_REG37	0x00d8	W	0x00000000	DDR PHY register 37
DDRPHY_REG38	0x00dc	W	0x00000001	DDR PHY register 38
DDRPHY_REG40	0x0100	W	0x000000aa	DDR PHY register 40
DDRPHY_REG41	0x0104	W	0x000000aa	DDR PHY register 41
DDRPHY_REG46	0x0118	W	0x0000000c	DDR PHY register 46
DDRPHY_REG47	0x011c	W	0x00000000	DDR PHY register 47
DDRPHY_REG48	0x0120	W	0x00000001	DDR PHY register 48
DDRPHY_REG50	0x0140	W	0x000000aa	DDR PHY register 50
DDRPHY_REG51	0x0144	W	0x000000aa	DDR PHY register 51
DDRPHY_REG56	0x0158	W	0x0000000c	DDR PHY register 56
DDRPHY_REG57	0x015c	W	0x00000000	DDR PHY register 57
DDRPHY_REG58	0x0160	W	0x00000001	DDR PHY register 58
DDRPHY_REGB0	0x02c0	W	0x00000077	DDR PHY register B0
DDRPHY_REGB1	0x02c4	W	0x00000077	DDR PHY register B1
DDRPHY_REGB2	0x02c8	W	0x00000077	DDR PHY register B2
DDRPHY_REGB3	0x02cc	W	0x00000077	DDR PHY register B3
DDRPHY_REGB4	0x02d0	W	0x00000077	DDR PHY register B4
DDRPHY_REGB5	0x02d4	W	0x00000077	DDR PHY register B5
DDRPHY_REGB6	0x02d8	W	0x00000077	DDR PHY register B6
DDRPHY_REGB7	0x02dc	W	0x00000077	DDR PHY register B7
DDRPHY_REGB8	0x02e0	W	0x00000077	DDR PHY register B8
DDRPHY_REGB9	0x02e4	W	0x00000077	DDR PHY register B9
DDRPHY_REGBA	0x02e8	W	0x00000077	DDR PHY register BA
DDRPHY_REGBB	0x02ec	W	0x00000077	DDR PHY register BB
DDRPHY_REGBC	0x02f0	W	0x00000077	DDR PHY register BC
DDRPHY_REGBD	0x02f4	W	0x00000077	DDR PHY register BD
DDRPHY_REGBE	0x02f8	W	0x00000077	DDR PHY register BE
DDRPHY_REGC0	0x0300	W	0x00000077	DDR PHY register C0
DDRPHY_REGC1	0x0304	W	0x00000077	DDR PHY register C1
DDRPHY_REGC2	0x0308	W	0x00000077	DDR PHY register C2
DDRPHY_REGC3	0x030c	W	0x00000077	DDR PHY register C3
DDRPHY_REGC4	0x0310	W	0x00000077	DDR PHY register C4
DDRPHY_REGC5	0x0314	W	0x00000077	DDR PHY register C5
DDRPHY_REGC6	0x0318	W	0x00000077	DDR PHY register C6
DDRPHY_REGC7	0x031c	W	0x00000077	DDR PHY register C7
DDRPHY_REGC8	0x0320	W	0x00000077	DDR PHY register C8
DDRPHY_REGC9	0x0324	W	0x00000077	DDR PHY register C9
DDRPHY_REGCA	0x0328	W	0x00000077	DDR PHY register CA
DDRPHY_REGCB	0x032c	W	0x00000077	DDR PHY register CB
DDRPHY_REGCC	0x0330	W	0x00000077	DDR PHY register CC
DDRPHY_REGCD	0x0334	W	0x00000077	DDR PHY register CD
DDRPHY_REGCE	0x0338	W	0x00000077	DDR PHY register CE
DDRPHY_REGCF	0x033c	W	0x00000077	DDR PHY register CF
DDRPHY_REGD0	0x0340	W	0x00000077	DDR PHY register D0
DDRPHY_REGD1	0x0344	W	0x00000077	DDR PHY register D1

Name	Offset	Size	Reset Value	Description
DDRPHY_REGD2	0x0348	W	0x00000077	DDR PHY register D2
DDRPHY_REGD3	0x034c	W	0x00000077	DDR PHY register D3
DDRPHY_REGD4	0x0350	W	0x00000077	DDR PHY register D4
DDRPHY_REGD5	0x0354	W	0x00000077	DDR PHY register D5
DDRPHY_REGD6	0x0358	W	0x00000077	DDR PHY register D6
DDRPHY_REGD7	0x035c	W	0x00000077	DDR PHY register D7
DDRPHY_REGD8	0x0360	W	0x00000077	DDR PHY register D8
DDRPHY_REGD9	0x0364	W	0x00000077	DDR PHY register D9
DDRPHY_REGDA	0x0368	W	0x00000077	DDR PHY register DA
DDRPHY_REGDB	0x036c	W	0x00000077	DDR PHY register DB
DDRPHY_REGDC	0x0370	W	0x00000077	DDR PHY register DC
DDRPHY_REGDD	0x0374	W	0x00000077	DDR PHY register DD
DDRPHY_REGDE	0x0378	W	0x00000077	DDR PHY register DE
DDRPHY_REGDF	0x037c	W	0x00000077	DDR PHY register DF
DDRPHY_REGE0	0x0380	W	0x00000007	DDR PHY register E0
DDRPHY_REGE1	0x0384	W	0x00000077	DDR PHY register E1
DDRPHY_REGE2	0x0388	W	0x00000077	DDR PHY register E2
DDRPHY_REGE3	0x038c	W	0x00000077	DDR PHY register E3
DDRPHY_REGE4	0x0390	W	0x00000077	DDR PHY register E4
DDRPHY_REGE5	0x0394	W	0x00000077	DDR PHY register E5
DDRPHY_REGE6	0x0398	W	0x00000077	DDR PHY register E6
DDRPHY_REGE7	0x039c	W	0x00000077	DDR PHY register E7
DDRPHY_REGE8	0x03a0	W	0x00000077	DDR PHY register E8
DDRPHY_REGE9	0x03a4	W	0x00000077	DDR PHY register E9
DDRPHY_REGEA	0x03a8	W	0x00000077	DDR PHY register EA
DDRPHY_REGEB	0x03ac	W	0x00000007	DDR PHY register EB
DDRPHY_REGFA	0x03e8	W	0x00000000	DDR PHY register FA
DDRPHY_REGFB	0x03ec	W	0x00000000	DDR PHY register FB
DDRPHY_REGFC	0x03f0	W	0x00000000	DDR PHY register FC
DDRPHY_REGFD	0x03f4	W	0x00000000	DDR PHY register FD
DDRPHY_REGFE	0x03f8	W	0x00000000	DDR PHY register FE
DDRPHY_REGFF	0x03fc	W	0x00000000	DDR PHY register FF

Notes:**Size:** **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

10.4.2 Detail Registers Description

DDR_PCTL_SCFG

Address: Operational Base + offset (0x0000)

State Configuration Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x3	<p>bbflags_timing The n_bbflags is a NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on when precharge, activates, reads/writes are scheduled by the TCU block. It may be possible to de-assert n_bbflags earlier than calculated by the TCU block.</p> <p>Programming bbflags_timing is used to achieve this. The maximum recommended value is: PCTL_TCU_SED_P - TRP.t_rp. The programmed value is the maximum number of "early" cycles that n_bbflags maybe de-asserted. The actual achieved de-assertion depends on the traffic profile.</p> <p>In 1:2 mode the maximum allowed programmable value is 4'b0111</p> <p>In 1:1 mode the value can be 4'b1111</p>
7:1	RO	0x0	reserved
0	RW	0x0	<p>hw_low_power_en Enables the hardware low-power interface. Allows the system to request via hardware (c_sysreq input) to enter the memories into Self-Refresh.</p> <p>The handshaking between the request and acknowledge hardware low power signals (c_sysreq and c_sysack, respectively) is always performed, but the PCTL response depends on the value set on this register field and by the value driven on the c_active_in input pin.</p> <p>1'b0: Disabled. Requests are always denied and PCTL is unaffected by c_sysreq</p> <p>1'b1: Enabled. Requests are accepted or denied, depending on the current operational state of PCTL and on the value of c_active_in.</p>

DDR_PCTL_SCTL

Address: Operational Base + offset (0x0004)

Operational State Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	<p>state_cmd Issues an operational state transition request to the PCTL.</p> <p>3'b000: INIT (move to Init_mem from Config) 3'b001: CFG (move to Config from Init_mem or Access) 3'b010: GO (move to Access from Config) 3'b011: SLEEP (move to Low_power from Access) 3'b100: WAKEUP (move to Access from Low_power) Others: Reserved</p>

DDR_PCTL_STAT

Address: Operational Base + offset (0x0008)

Operational State Status Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RO	0x0	<p>lp_trig Reports the status of what triggered an entry to Low_power state. Is only set if in Low_power state. The individual bits report the following:</p> <ul style="list-style-type: none"> - lp_trig[2]: Software driven due to SCTL.state_cmd==SLEEP. - lp_trig[1]: Hardware driven due to Hardware Low Power Interface. - lp_trig[0]: Hardware driven due to Auto Self Refresh (MCFG1.sr_idle>0). <p>Note, if more than one trigger happens at the exact same time, more than one bit of lp_trig may be asserted high.</p>
3	RO	0x0	reserved
2:0	RO	0x0	<p>ctl_stat Returns the current operational state of the PCTL.</p> <p>3'b000: Init_mem 3'b001: Config 3'b010: Config_req 3'b011: Access 3'b100: Access_req 3'b101: Low_power 3'b110: Low_power_entry_req 3'b111: Low_power_exit_req Others: Reserved</p>

DDR_PCTL_INTRSTAT

Address: Operational Base + offset (0x000c)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	parity_intr Indicates that a DFI parity error has been detected 1'b0: No error 1'b1: Parity error
0	RO	0x0	ecc_intr Indicates that an ECC error has been detected 1'b0: No error 1'b1: Parity error

DDR_PCTL_MCMD

Address: Operational Base + offset (0x0040)
Memory Command Register

Bit	Attr	Reset Value	Description
31	RWSC	0x0	start_cmd Start command. When this bit is set to 1, the command operation defined in the cmd_opcode field is started. This bit is automatically cleared by the PCTL after the command is finished. The application can poll this bit to determine when PCTL is ready to accept another command. This bit cannot be cleared to 1'b0 by software.
30:28	RO	0x0	reserved
27:24	RW	0x0	cmd_add_del Set the additional delay associated with each command to 2^n internal timers clock cycles, where n is the bit field value. If n=0, the delay is 0. Max value is n=10.
23:20	RW	0x1	rank_sel Rank select for the command to be executed. 4'b0001: Rank 0 4'b0010: Rank 1 4'b0100: Rank 2 4'b1000: Rank 3 4'b0000: Reserved Multiple 1'b1s in rank_sel mean multiple ranks are selected, which is useful broadcasting commands in parallel to multiple ranks during initialization and configuration of the memories. If MCMD.cmd_opcode=RSTL, all ranks should be selected as it cannot be performed to individual ranks

Bit	Attr	Reset Value	Description
19:17	RW	0x0	<p>bank_addr Mode Register address driven on the memory bank address bits, BA1, BA0, during a Mode Register Set operation, defined by cmd_opcode=MRS. For other values of cmd_opcode, this field is ignored.</p> <p>3'b000: MR0 3'b001: MR1 3'b010: MR2 3'b011: MR3 Others: Reserved</p>
16:4	RW	0x0000	<p>cmd_addr Mode Register value driven on the memory address bits, A12 to A0, during a Mode Register Set operation defined by cmd_opcode=MRS. For other values of cmd_opcode this field is ignored. Refer to the memory specification for the correct settings of the various bits of this field during a MRS operation.</p>
3:0	RW	0x0	<p>cmd_opcode Command to be issued to the memory. 4'b0000: Deselect. This is only used for timing purpose, no actual direct Deselect command is passed to the memories. 4'b0001: Precharge All (PREA) 4'b0010: Refresh (REF) 4'b0011: Mode Register Set (MRS) 4'b0100: ZQ Calibration Short (ZQCS, only applies to DDR3) 4'b0101: ZQ Calibration Long (ZQCL, only applies to DDR3) 4'b0110: Software Driven Reset (RSTL, only applies to DDR3) 4'b0111: Reserved 4'b1000: Mode Register Read (MRR) - is MPR in DDR3 Others: Reserved</p>

DDR_PCTL_POWCTL

Address: Operational Base + offset (0x0044)

Power Up Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RWSC	0x0	<p>power_up_start Start the memory power up sequence. When this bit is set to 1'b1, PCTL starts the CKE and RESET# power up sequence to the memories. This bit is automatically cleared by PCTL after the sequence is completed. This bit cannot be cleared to 1'b0 by software.</p>

DDR_PCTL_POWSTAT

Address: Operational Base + offset (0x0048)

Power Up Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	power_up_done Returns the status of the memory power-up sequence. 1'b0: Power-up sequence has not been performed. 1'b1: Power-up sequence has been performed.

DDR_PCTL_CMDTSTAT

Address: Operational Base + offset (0x004c)

Command Timers Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	cmd_tstat Returns the status of the timers for memory commands. This ANDs all the command timers together. 1'b0: One or more command timers has not expired. 1'b1: All command timers have expired.

DDR_PCTL_CMDTSTATEN

Address: Operational Base + offset (0x0050)

Command Timers Status Enable Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	cmd_tstat_en Enables the generation of the status of the timers for memory commands. Is enabled before CMDTSTAT register is read. 1'b0: Disabled 1'b1: Enabled

DDR_PCTL_MRRCFG0

Address: Operational Base + offset (0x0060)

Mode Register Read Configuration 0

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	mrr_byte_sel Selects which byte's data to store when performing an MRR command via MCMD. LegalValues: 0 .. 8

DDR_PCTL_MRRSTAT0

Address: Operational Base + offset (0x0064)

Mode Register Read Status 0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	mrrstat_beat3 MRR/MPR read data beat 3
23:16	RO	0x00	mrrstat_beat2 MRR/MPR read data beat 2
15:8	RO	0x00	mrrstat_beat1 MRR/MPR read data beat 1
7:0	RO	0x00	mrrstat_beat0 MRR/MPR read data beat 0

DDR_PCTL_MRRSTAT1

Address: Operational Base + offset (0x0068)

Mode Register Read Status 0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	mrrstat_beat7 MRR/MPR read data beat 7
23:16	RO	0x00	mrrstat_beat6 MRR/MPR read data beat 6
15:8	RO	0x00	mrrstat_beat5 MRR/MPR read data beat 5
7:0	RO	0x00	mrrstat_beat4 MRR/MPR read data beat 4

DDR_PCTL_MCFG

Address: Operational Base + offset (0x0080)

Memory Configuration Register

Bit	Attr	Reset Value	Description
31:20	RO	0x00	Reserved
19:18	RW	0x1	tfa_w_cfg Sets tFAW to be 4, 5 or 6 times tRRD. 2'b00: set tFAW=4*tRRD 2'b01: set tFAW=5*tRRD 2'b10: set tFAW=6*tRRD
17	RW	0x0	pd_exit_mode Selects the mode for Power Down Exit. For DDR3, the power down exit mode setting in PCTL must be consistent with the value programmed into the power down exit mode bit of MRO. 1'b0: slow exit 1'b1: fast exit
16	RW	0x0	pd_type Sets the Power down type. 1'b0: Precharge Power Down 1'b1: Active Power Down

Bit	Attr	Reset Value	Description
15:8	RW	0x00	pd_idle Power-down idle period in n_clk cycles. Memories are placed into power-down mode if the NIF is idle for pd_idle n_clk cycles. The automatic power down function is disabled when pd_idle=0.
7:5	RO	0x0	reserved
4	RW	0x0	stagger_cs For multi-rank commands from the DCU, stagger the assertion of CS_N to odd and even ranks by one n_clk cycle. This is useful when using RDIMMs, when multi-rank commands may be interpreted as writes to control words in the register chip. 1'b0: Do not stagger CS_N 1'b1: Stagger CS_N
3	RW	0x0	two_t_en Enables 2T timing for memory commands. 1'b0: Disabled 1'b1: Enabled
2	RO	0x0	Reserved
1	RW	0x0	cke_or_en This bit is intended to be set for 4-rank RDIMMs, which have a 2-bit CKE input. If set, dfi_cke[0] is asserted to enable either of the even ranks (0 and 2), while dfi_cke[1] is asserted to enable either of the odd ranks (1 and 3). dfi_cke[3:2] are inactive (0) 1'b0: Disabled 1'b1: Enabled
0	RW	0x0	mem_bl DDR Burst Length. The BL setting in DDR3 must be consistent with the value programmed into the BL field of MR0. 1'b0: BL4, Burst length of 4 1'b1: BL8, Burst length of 8 (MR0.BL=2'b00 for DDR3)

DDR_PCTL_PPCFG

Address: Operational Base + offset (0x0084)

Partially Populated Memories Configuration Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:1	RW	0x00	<p>rpmem_dis Reduced Population Disable bits. Setting these bits disables the corresponding NIF/DDR data lanes from writing or reading data. Lane 0 is always present, hence only 8 bits are required for the remaining lanes including the ECC lane. In 1:2 mode bit 0 of rpmem_dis covers n_wdata/n_rdata/m_ctl_d/m_phy_q[63:32], bit 1 [95:64] etc. In 1:1 mode bit 0 of rpmem_dis covers n_wdata/n_rdata/m_ctl_d/m_phy_q[31:16], bit 2 [47:32] etc. There are no restrictions on which byte lanes can be disabled, other than byte lane 0 is required. Gaps between enabled byte lanes are allowed For each bit: 1'b0: lane exists 1'b1: lane is disabled</p>
0	RW	0x0	<p>ppmem_en Partially Population Enable bit. Setting this bit enables the partial population of external memories where the entire application bus is routed to a reduced size memory system. The lower half of the SDRAM data bus, bit 0 up to bit PCTL_M_DW/2-1, is the active portion when Partially Populated memories are enabled. 1'b0: Disabled 1'b1: Enabled</p>

DDR_PCTL_MSTAT

Address: Operational Base + offset (0x0088)

Memory Status Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	<p>self_refresh Indicates if PCTL, through auto self refresh, has placed the memories in Self Refresh. 1'b0: Memory is not in Self Refresh 1'b1: Memory is in Self Refresh</p>
1	RO	0x0	<p>clock_stop Indicates if PCTL has placed the memories in Clock Stop. 1'b0: Memory is not in Clock Stop 1'b1: Memory is in Clock Stop</p>
0	RO	0x0	<p>power_down Indicates if PCTL has placed the memories in Power Down. 1'b0: Memory is not in Power Down 1'b1: Memory is in Power-Down</p>

DDR_PCTL_MCFG1

Address: Operational Base + offset (0x0090)

Memory Configuration 1 Register

Bit	Attr	Reset Value	Description
31	RW	0x0	hw_exit_idle_en When this bit is programmed to 1'b1 the c_active_in pin can be used to exit from the automatic clock stop , power down or self-refresh modes.
30:24	RO	0x0	reserved
23:16	RW	0x00	hw_idle Hardware idle period. The c_active output is driven high if the NIF is idle in Access state for hw_idle * 32 * n_clk cycles. The hardware idle function is disabled when hw_idle=0.
15:8	RO	0x0	reserved
7:0	RW	0x00	sr_idle Self Refresh idle period. Memories are placed into Self-Refresh mode if the NIF is idle in Access state for sr_idle * 32 * n_clk cycles. The automatic self refresh function is disabled when sr_idle=0.

DDR_PCTL_DTUPDES

Address: Operational Base + offset (0x0094)

DTU Status Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RO	0x0	dtu_rd_missing Indicates if one or more read beats of data did not return from memory.
12:9	RO	0x0	dtu_eaffl Indicates the number of entries in the FIFO that is holding the log of error addresses for data comparison
8	RO	0x0	dtu_random_error Indicates that the random data generated had some failures when written and read to the memories
7	RO	0x0	dtu_err_b7 Detected at least 1 bit error for bit 7 in the programmable data buffers
6	RO	0x0	dtu_err_b6 Detected at least 1 bit error for bit 6 in the programmable data buffers
5	RO	0x0	dtu_err_b5 Detected at least 1 bit error for bit 5 in the programmable data buffers
4	RO	0x0	dtu_err_b4 Detected at least 1 bit error for bit 4 in the programmable data buffers

Bit	Attr	Reset Value	Description
3	RO	0x0	dtu_err_b3 Detected at least 1 bit error for bit 3 in the programmable data buffers
2	RO	0x0	dtu_err_b2 Detected at least 1 bit error for bit 2 in the programmable data buffers
1	RO	0x0	dtu_err_b1 Detected at least 1 bit error for bit 1 in the programmable data buffers
0	RO	0x0	dtu_err_b0 Detected at least 1 bit error for bit 0 in the programmable data buffers

DDR_PCTL_DTUNA

Address: Operational Base + offset (0x0098)

DTU Number of Addresses Created Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dtu_num_address Indicates the number of addresses that were created on the NIF interface during random data generation.

DDR_PCTL_DTUNE

Address: Operational Base + offset (0x009c)

DTU Number of Errors Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dtu_num_errors Indicates the number of errors that were detected on the readback of the NIF data during random data generation.

DDR_PCTL_DTUPRDO

Address: Operational Base + offset (0x00a0)

DTU Parallel Read 0 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_1 Allows all the bit ones from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_0 Allows all the bit zeros from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD1

Address: Operational Base + offset (0x00a4)

DTU Parallel Read 1 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_3 Allows all the bit threes from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_2 Allows all the bit twos from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD2

Address: Operational Base + offset (0x00a8)

DTU Parallel Read 2 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_5 Allows all the bit fives from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_4 Allows all the bit fours from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD3

Address: Operational Base + offset (0x00ac)

DTU Parallel Read 3 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_7 Allows all the bit sevens from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_6 Allows all the bit sixes from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUAWDT

Address: Operational Base + offset (0x00b0)
 DTU Address Width Register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:9	RW	0x1	number_ranks Number of supported memory ranks. 2'b00: 1 rank 2'b01: 2 ranks 2'b10: 3 ranks 2'b11: 4 ranks
8	RO	0x0	reserved
7:6	RW	0x2	row_addr_width Width of the memory row address bits. 2'b00: 13 bits wide 2'b01: 14 bits wide 2'b10: 15 bits wide 2'b11: 16 bits wide
5	RO	0x0	reserved
4:3	RW	0x2	bank_addr_width Width of the memory bank address bits. 2'b00: 2 bits wide (4 banks) 2'b01: 3 bits wide (8 banks) Others: Reserved
2	RO	0x0	reserved
1:0	RW	0x0	column_addr_width Width of the memory column address bits. 2'b00: 7 bits wide 2'b01: 8 bits wide 2'b10: 9 bits wide 2'b11: 10 bits wide

DDR_PCTL_TOGCNT1U

Address: Operational Base + offset (0x00c0)
 Toggle Counter 1us Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x064	toggle_counter_1u The number of internal timers clock cycles

DDR_PCTL_TINIT

Address: Operational Base + offset (0x00c4)
 t_init Timing Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:0	RW	0x0c8	t_init Defines the time period (in us) to hold dfi_cke and dfi_reset_n stable during the memory power up sequence. The value programmed must correspond to at least 200us. The actual time period defined is TINIT * TOGCNT1U * internal timers clock .period

DDR_PCTL_TRSTH

Address: Operational Base + offset (0x00c8)

t_rsth Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	t_rsth Defines the time period (in us) to hold the dfi_reset_n signal high after it is de-asserted during the DDR3 Power Up/Reset sequence. The value programmed for DDR3 must correspond to minimum 500us of delay. The actual time period defined is TRSTH * TOGCNT1U * internal timers clock period.

DDR_PCTL_TOGCNT100N

Address: Operational Base + offset (0x00cc)

Toggle Counter 100ns

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x01	toggle_counter_100n The number of internal timers clock cycles.

DDR_PCTL_TREFI

Address: Operational Base + offset (0x00d0)

t_refi Timing Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x01	t_refi Defines the time period (in 100ns units) of the Refresh interval. The actual time period defined is TREFI * TOGCNT100N * internal timers clock period.

DDR_PCTL_TMRD

Address: Operational Base + offset (0x00d4)

t_mrd Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x1	t_mrd Mode Register Set command cycle time in memory clock cycles. DDR3: Time from MRS to MRS command. DDR3 Legal Values: 2..4

DDR_PCTL_TRFC

Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x001	t_rfc Refresh to Active/Refresh command time in memory clock cycles. DDR3 Legal Values: 36.. 374

DDR_PCTL_TRP

Address: Operational Base + offset (0x00dc)

t_trp Timing Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	RW	0x1	prea_extra Additional cycles required for a Precharge All (PREA) command - in addition to t_rp. In terms of memory clock cycles DDR3 Value: 0
15:4	RO	0x0	reserved
3:0	RW	0x6	t_rp Precharge period in memory clock cycles. DDR3 Legal Values: 5..14

DDR_PCTL_TRTW

Address: Operational Base + offset (0x00e0)

t_rtw Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	t_rtw Read to Write turnaround time in memory clock cycles. DDR3 Legal Values: 2..10

DDR_PCTL_TAL

Address: Operational Base + offset (0x00e4)

AL Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	t_al Additive Latency in memory clock cycles. For DDR3 this must be 0, CL-1, CL-2 depending weather the AL value in MR1 is 0,1, or 2 respectively. CL is the CAS latency programmed into MR0. DDR3 Legal Values: 0, CL-1, CL-2 (depending on AL=0,1,2 in MR1)

DDR_PCTL_TCL

Address: Operational Base + offset (0x00e8)

CL Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_cl CAS Latency in memory clock cycles. If DDR3, the PCTL setting must match the value programmed into the CL field of MR0. DDR3 Legal Value: CL

DDR_PCTL_TCWL

Address: Operational Base + offset (0x00ec)

CWL Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x3	t_cwl CAS Write Latency in memory clock cycles. For DDR3, the setting must match the value programmed in the memory CWL field of MR2. DDR3 Legal Value: CWL

DDR_PCTL_TRAS

Address: Operational Base + offset (0x00f0)

t_ras Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x10	t_ras Activate to Precharge command time in memory clock cycles. DDR3 Legal Values: 15..38

DDR_PCTL_TRC

Address: Operational Base + offset (0x00f4)

t_rc Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x16	t_rc Row Cycle time in memory clock cycles. Specifies the minimum Activate to Activate distance for accesses to same bank. DDR3 Legal Values: 20..52

DDR_PCTL_TRCD

Address: Operational Base + offset (0x00f8)

t_rcd Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x6	t_rcd Row to Column delay in memory clock cycles. Specifies the minimum Activate to Column distance. DDR3 Legal Values: 5..14

DDR_PCTL_TRRD

Address: Operational Base + offset (0x00fc)

t_rrd Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_rrd Row-to-Row delay in memory clock cycles. Specifies the minimum Activate-to-Activate distance for consecutive accesses to different banks in the same rank. DDR3 Legal Values: 4..8

DDR_PCTL_TRTP

Address: Operational Base + offset (0x0100)

t_rtp Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x3	t_rtp Read to Precharge time in memory clock cycles. Specifies the minimum distance Read to Precharge for consecutive accesses to same bank. DDR3 Legal Values: 3..8

DDR_PCTL_TWR

Address: Operational Base + offset (0x0104)

t_wr Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x06	t_wr Write recovery time in memory clock cycles. When using close page the PCTL setting must be consistent with the WR field setting of MR0. DDR3 Legal Values: 6..16

DDR_PCTL_TWTR

Address: Operational Base + offset (0x0108)

t_wtr Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_wtr Write to Read turnaround time, in memory clock cycles. DDR3 Legal Values: 3..8

DDR_PCTL_TEXSR

Address: Operational Base + offset (0x010c)

t_exsr Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x001	t_exsr Exit Self Refresh to first valid command delay, in memory clock cycles. For DDR3, this should be programmed to match tXSDLL (SRE to a command requiring DLL locked) as defined by the memory device specification. DDR3 Typical Value: 512

DDR_PCTL_TXP

Address: Operational Base + offset (0x0110)

t_xp Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x1	t_xp Exit Power Down to first valid command delay when DLL is on (fast exit), measured in memory clock cycles. Legal Values: 1..7

DDR_PCTL_TXPDLL

Address: Operational Base + offset (0x0114)

t_xpdll Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	t_xpdll Exit Power Down to first valid command delay when DLL is off (slow exit), measured in memory clock cycles. DDR3 Legal Values: 3..63

DDR_PCTL_TZQCS

Address: Operational Base + offset (0x0118)

t_zqcs Timing Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	t_zqcs SDRAM ZQ Calibration Short period, in memory clock cycles. Should be programmed to match the tZQCS timing value as defined in the memory specification. DDR3 Typical Value: 64

DDR_PCTL_TZQCSI

Address: Operational Base + offset (0x011c)

t_zqcsi Timing Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t_zqcsi SDRAM ZQCS interval, measured in Refresh interval units. The total time period defined is TZQCSI*TREFI * TOGCNT100N * internal timers clock period. Programming a value of 0 in t_zqcsi disables the auto-ZQCS functionality in PCTL. DDR3 Legal Values: 0..4294967295

DDR_PCTL_TDQS

Address: Operational Base + offset (0x0120)

t_dqs Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x1	t_dqs Additional data turnaround time in memory clock cycles for accesses to different ranks. Used to increase the distance between column commands to different ranks, allowing more tolerance as the driver source changes on the bidirectional DQS and/or DQ signals. DDR3 Legal Values: 1..7

DDR_PCTL_TCKSRE

Address: Operational Base + offset (0x0124)

t_cksre Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_cksrc In DDR3, this is the time after Self Refresh Entry that CKE is held high before going low. In memory clock cycles. Specifies the clock disable delay after SRE. This should be programmed to match the greatest value between 10ns and 5 memory clock periods. DDR3 Legal Values: 5..15

DDR_PCTL_TCKSRX

Address: Operational Base + offset (0x0128)

t_cksrc Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_cksrc In DDR3, this is the time (before Self Refresh Exit) that CKE is maintained high before issuing SRX. In memory clock cycles. Specifies the clock stable time before SRX. This should be programmed to match the greatest value between 10ns and 5 memory clock periods. DDR3 Legal Values: 5..15

DDR_PCTL_TCKE

Address: Operational Base + offset (0x012c)

t_cke Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x3	t_cke CKE minimum pulse width in memory clock cycles. DDR3 Legal Values: 3..6

DDR_PCTL_TMOD

Address: Operational Base + offset (0x0130)

t_mod Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_mod In DDR3 mode, this is the time from MRS to any valid non-MRS command (except DESELECT or NOP) in memory clock cycles. DDR3 Legal Values: 0..31

DDR_PCTL_TRSTL

Address: Operational Base + offset (0x0134)

Reset Low Timing Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	t_rstl Memory Reset Low time, in memory clock cycles. Defines the time period to hold dfi_reset_n signal low during a software driven DDR3 Reset Operation. The value programmed must correspond to at least 100ns of delay. DDR3 Legal Values: 1..127

DDR_PCTL_TZQCL

Address: Operational Base + offset (0x0138)

t_zqcl Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	t_zqcl SDRAM ZQ Calibration Long period in memory clock cycles. If DDR3, should be programmed to match the memory tZQinit timing value for the first ZQCL command during memory initialization; should be programmed to match tZQoper timing value after reset and initialization. DDR3 Legal Values: 0..1023

DDR_PCTL_TMRR

Address: Operational Base + offset (0x013c)

t_mrr Timing Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x02	t_mrr Time for a Mode Register Read (MRR command from MCMD).

DDR_PCTL_TCLESR

Address: Operational Base + offset (0x0140)

t_ckesr Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_ckesr Minimum CKE low width for Self Refresh entry to exit timing in memory clock cycles. Recommended settings: DDR3 : t_ckesr = t_cke + 1 DDR3 Legal Values: 4..7

DDR_PCTL_DTUWACTL

Address: Operational Base + offset (0x0200)

DTU Write Address Control

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dtu_wr_rank Write rank to where data is to be targeted
29	RO	0x0	reserved
28:13	RW	0x0000	dtu_wr_row Write row to where data is to be targeted
12:10	RW	0x0	dtu_wr_bank Write bank to where data is to be targeted
9:0	RW	0x000	dtu_wr_col FWrite column to where data is to be targeted

DDR_PCTL_DTRACTL

Address: Operational Base + offset (0x0204)

DTU Read Address Control Register

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dtu_rd_rank Read rank from where data comes
29	RO	0x0	reserved
28:13	RW	0x0000	dtu_rd_row Read row from where data comes
12:10	RW	0x0	dtu_rd_bank Read bank from where data comes
9:0	RW	0x000	dtu_rd_col Read column from where data comes

DDR_PCTL_DTUCFG

Address: Operational Base + offset (0x0208)

DTU Configuration Control Register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:16	RW	0x00	dtu_row_increments Number of times to increment the row address when generating random data, up to a maximum of 127 times.
15	RW	0x0	dtu_wr_multi_rd When set puts the DTU into write once multiple reads mode.
14	RW	0x0	dtu_data_mask_en Controls whether random generated data masks are transmitted. Unless enabled all data bytes are written to memory and expected to be read from memory.
13:10	RW	0x0	dtu_target_lane Selects one of the byte lanes for data comparison into the programmable read data buffer.
9	RW	0x0	dtu_generate_random Generate transfers using random data, otherwise generate transfers from the programmable write data buffers.

Bit	Attr	Reset Value	Description
8	RW	0x0	dtu_incr_banks When the column address rolls over increment the bank address until we reach and conclude bank 7.
7	RW	0x0	dtu_incr_cols Increment the column address until we saturate. Return to zero if DTUCFG.dtu_incr_banks is set to 1 and we are not at bank 7.
6:1	RW	0x00	dtu_nalen Length of the NIF transfer sequence that is passed through the PCTL for each created address.
0	RW	0x0	dtu_enable When set, allows the DTU module to take ownership of the NIF interface: 1'b1: DTU enabled 1'b0: DTU disabled

DDR_PCTL_DTUectl

Address: Operational Base + offset (0x020c)

DTU Execute Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RWSC	0x0	wr_multi_rd_rst When set, resets the DTU in write once multiple reads mode, to allow a new write to be performed. This bit automatically clears.
1	RWSC	0x0	run_error_reports When set, initiates the calculation of the error status bits. This bit automatically clears when the re-calculation is done. This is only used in debug mode to verify the comparison logic.
0	RWSC	0x0	run_dtu When set, initiates the running of the DTU read and write transfer. This bit automatically clears when the transfers are completed

DDR_PCTL_DTUwdo

Address: Operational Base + offset (0x0210)

DTU Write Data #0 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte3 Write data byte
23:16	RW	0x00	dtu_wr_byte2 Write data byte
15:8	RW	0x00	dtu_wr_byte1 Write data byte
7:0	RW	0x00	dtu_wr_byte0 Write data byte

DDR_PCTL_DTUWD1

Address: Operational Base + offset (0x0214)

DTU Write Data #1 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte7 Write data byte
23:16	RW	0x00	dtu_wr_byte6 Write data byte
15:8	RW	0x00	dtu_wr_byte5 Write data byte
7:0	RW	0x00	dtu_wr_byte4 Write data byte

DDR_PCTL_DTUWD2

Address: Operational Base + offset (0x0218)

DTU Write Data #2 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte11 Write data byte
23:16	RW	0x00	dtu_wr_byte10 Write data byte
15:8	RW	0x00	dtu_wr_byte9 Write data byte
7:0	RW	0x00	dtu_wr_byte8 Write data byte

DDR_PCTL_DTUWD3

Address: Operational Base + offset (0x021c)

DTU Write Data #3 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte15 Write data byte
23:16	RW	0x00	dtu_wr_byte14 Write data byte
15:8	RW	0x00	dtu_wr_byte13 Write data byte
7:0	RW	0x00	dtu_wr_byte12 Write data byte

DDR_PCTL_DTUWDM

Address: Operational Base + offset (0x0220)

DTU Write Data Mask Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	dm_wr_byte0 Write data mask bit, one bit for each byte. Each bit should be 0 for a byte lane that contains valid write data.

DDR_PCTL_DTURD0

Address: Operational Base + offset (0x0224)

DTU Read Data #0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte3 Read byte
23:16	RO	0x00	dtu_rd_byte2 Read byte
15:8	RO	0x00	dtu_rd_byte1 Read byte
7:0	RO	0x00	dtu_rd_byte0 Read byte

DDR_PCTL_DTURD1

Address: Operational Base + offset (0x0228)

DTU Read Data #1 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte7 Read byte
23:16	RO	0x00	dtu_rd_byte6 Read byte
15:8	RO	0x00	dtu_rd_byte5 Read byte
7:0	RO	0x00	dtu_rd_byte4 Read byte

DDR_PCTL_DTURD2

Address: Operational Base + offset (0x022c)

DTU Read Data #2 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte11 Read byte
23:16	RO	0x00	dtu_rd_byte10 Read byte
15:8	RO	0x00	dtu_rd_byte9 Read byte
7:0	RO	0x00	dtu_rd_byte8 Read byte

DDR_PCTL_DTURD3

Address: Operational Base + offset (0x0230)

DTU Read Data #3 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte15 Read byte
23:16	RO	0x00	dtu_rd_byte14 Read byte

Bit	Attr	Reset Value	Description
15:8	RO	0x00	dtu_rd_byte13 Read byte
7:0	RO	0x00	dtu_rd_byte12 Read byte

DDR_PCTL_DTULFSRWD

Address: Operational Base + offset (0x0234)

DTU LFSR Seed for Write Data Generation Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dtu_lfsr_wseed This is the initial seed for the random write data generation LFSR (linear feedback shift register), shared with the write mask generation.

DDR_PCTL_DTULFSRRD

Address: Operational Base + offset (0x0238)

DTU LFSR Seed for Read Data Generation Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dtu_lfsr_rseed This is the initial seed for the random read data generation LFSR (linear feedback shift register), this is shared with the read mask generation. The read data mask is reconstructed the same as the write data mask was created, allowing the "on the fly comparison" ignore bytes which were not written.

DDR_PCTL_DTUEAF

Address: Operational Base + offset (0x023c)

DTU Error Address FIFO Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	ea_rank Indicates the rank that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.
29	RO	0x0	reserved
28:13	RO	0x0000	ea_row Indicates the row that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.
12:10	RO	0x0	ea_bank Indicates the bank that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes

Bit	Attr	Reset Value	Description
9:0	RO	0x000	ea_column Indicates the column address that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.

DDR_PCTL_DFITCTRLDELAY

Address: Operational Base + offset (0x0240)

DFI tctrl_delay Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tctrl_delay Specifies the number of DFI clock cycles after an assertion or deassertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFIODTCFG

Address: Operational Base + offset (0x0244)

DFI ODT Configuration

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RW	0x0	rank3_odt_default Default ODT value of rank 3 when there is no read/write activity
27	RW	0x0	rank3_odt_write_sel Enable/disable ODT for rank 3 when a write access is occurring on this rank
26	RW	0x0	rank3_odt_write_nse Enable/disable ODT for rank 3 when a write access is occurring on a different rank
25	RW	0x0	rank3_odt_read_sel Enable/disable ODT for rank 3 when a read access is occurring on this rank
24	RW	0x0	rank3_odt_read_nsel Enable/disable ODT for rank 3 when a read access is occurring on a different rank
23:21	RO	0x0	reserved
20	RW	0x0	rank2_odt_default Default ODT value of rank 2 when there is no read/write activity
19	RW	0x0	rank2_odt_write_sel Enable/disable ODT for rank 2 when a write access is occurring on this rank

Bit	Attr	Reset Value	Description
18	RW	0x0	rank2_odt_write_nse Enable/disable ODT for rank 2 when a write access is occurring on a different rank
17	RW	0x0	rank2_odt_read_sel Enable/disable ODT for rank 2 when a read access is occurring on this rank
16	RW	0x0	rank2_odt_read_nsel Enable/disable ODT for rank 2 when a read access is occurring on a different rank
15:13	RO	0x0	reserved
12	RW	0x0	rank1_odt_default Default ODT value of rank 1 when there is no read/write activity
11	RW	0x0	rank1_odt_write_sel Enable/disable ODT for rank 1 when a write access is occurring on this rank
10	RW	0x0	rank1_odt_write_nse Enable/disable ODT for rank 1 when a write access is occurring on a different rank
9	RW	0x0	rank1_odt_read_sel Enable/disable ODT for rank 1 when a read access is occurring on this rank
8	RW	0x0	rank1_odt_read_nsel Enable/disable ODT for rank 1 when a read access is occurring on a different rank
7:5	RO	0x0	reserved
4	RW	0x0	rank0_odt_default Default ODT value of rank 0 when there is no read/write activity
3	RW	0x0	rank0_odt_write_sel Enable/disable ODT for rank 0 when a write access is occurring on this rank
2	RW	0x0	rank0_odt_write_nse Enable/disable ODT for rank 0 when a write access is occurring on a different rank
1	RW	0x0	rank0_odt_read_sel Enable/disable ODT for rank 0 when a read access is occurring on this rank
0	RW	0x0	rank0_odt_read_nsel Enable/disable ODT for rank 0 when a read access is occurring on a different rank

DDR_PCTL_DFIODTCFG1

Address: Operational Base + offset (0x0248)

DFI ODT Timing Configuration 1 (for Latency and Length)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26:24	RW	0x6	ODT length for BL8 read transfers Length of dfi_odt signal for BL8 reads. This is in terms of SDR cycles. For BL4 reads, the length of dfi_odt is always 2 cycles shorter than the value in this register field.
23:19	RO	0x0	reserved
18:16	RW	0x6	ODT length for BL8 write transfers Length of dfi_odt signal for BL8 writes. This is in terms of SDR cycles. For BL4 writes, the length of dfi_odt is always 2 cycles shorter than the value in this register field.
15:13	RO	0x0	reserved
12:8	RW	0x00	ODT latency for reads Latency after a read command that dfi_odt is set. This is in terms of SDR cycles.
7:5	RO	0x0	reserved
4:0	RW	0x00	ODT latency for writes Latency after a write command that dfi_odt is set. This is in terms of SDR cycles

DDR_PCTL_DFIODTRANKMAP

Address: Operational Base + offset (0x024c)

DFI ODT Rank Mapping

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:12	RW	0x8	Rank mapping for dfi_odt[3] Determines whether dfi_odt[3] should be asserted when the PCTL requires to terminate each rank Bit 15 = 1: dfi_odt[3] will be asserted to terminate rank 3 Bit 14 = 1: dfi_odt[3] will be asserted to terminate rank 2 Bit 13 = 1: dfi_odt[3] will be asserted to terminate rank 1 Bit 12 = 1: dfi_odt[3] will be asserted to terminate rank 0 This field exists only if PCTL_M_NRANKS = 4
11:8	RW	0x4	Rank mapping for dfi_odt[2] Determines which rank access(es) will cause dfi_odt[2] to be asserted Bit 11 = 1: dfi_odt[2] will be asserted to terminate rank 3 Bit 10 = 1: dfi_odt[2] will be asserted to terminate rank 2 Bit 9 = 1: dfi_odt[2] will be asserted to terminate rank 1 Bit 8 = 1: dfi_odt[2] will be asserted to terminate rank 0 This field exists only if PCTL_M_NRANKS = 4

Bit	Attr	Reset Value	Description
7:4	RW	0x2	<p>Rank mapping for dfi_odt[1]</p> <p>Determines which rank access(es) will cause dfi_odt[1] to be asserted</p> <p>Bit 7= 1: dfi_odt[1] will be asserted to terminate rank 3</p> <p>Bit 6= 1: dfi_odt[1] will be asserted to terminate rank 2</p> <p>Bit 5= 1: dfi_odt[1] will be asserted to terminate rank 1</p> <p>Bit 4= 1: dfi_odt[1] will be asserted to terminate rank 0</p> <p>This field exists only if PCTL_M_NRANKS ></p>
3:0	RW	0x1	<p>Rank mapping for dfi_odt[0]</p> <p>Determines which rank access(es) will cause dfi_odt[0] to be asserted</p> <p>Bit 3= 1: dfi_odt[0] will be asserted to terminate rank 3</p> <p>Bit 2= 1: dfi_odt[0] will be asserted to terminate rank 2</p> <p>Bit 1= 1: dfi_odt[0] will be asserted to terminate rank 1</p> <p>Bit 0= 1: dfi_odt[0] will be asserted to terminate rank 0</p>

DDR_PCTL_DFITPHYWRDATA

Address: Operational Base + offset (0x0250)

DFI tphy_wrdata Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	tphy_wrdata Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted to when the associated write data is driven on the dfi_wrdata signal. This has no impact on performance, only adjusts the relative time between enable and data transfer.

DDR_PCTL_DFITPHYWRLAT

Address: Operational Base + offset (0x0254)

DFI tphy_wrlat Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	tphy_wrlat Specifies the number of DFI clock cycles between when a write command is sent on the DFI control interface and when the dfi_wrdata_en signal is asserted.

DDR_PCTL_DFITRDDATAEN

Address: Operational Base + offset (0x0260)

DFI trddata_en Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	trddata_en Specifies the number of DFI clock cycles from the assertion of a read command on the DFI to the assertion of the dfi_rddata_en signal.

DDR_PCTL_DFITPHYRDLAT

Address: Operational Base + offset (0x0264)

DFI tphy_rdlat Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x0f	tphy_rdlat Specifies the maximum number of DFI clock cycles allowed from the assertion of the dfi_rddata_en signal to the assertion of the dfi_rddata_valid signal.

DDR_PCTL_DFITPHYUPDTYPE0

Address: Operational Base + offset (0x0270)

DFI tphyupd_type0 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type0 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x0. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE1

Address: Operational Base + offset (0x0274)

DFI tphyupd_type1 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type1 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x1. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE2

Address: Operational Base + offset (0x0278)

DFI tphyupd_type2 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type2 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x2. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE3

Address: Operational Base + offset (0x027c)

DFI tphyupd_type3 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type3 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x3. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITCTRLUPDMIN

Address: Operational Base + offset (0x0280)

DFI tctrlupd_min Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0010	tctrlupd_min Specifies the minimum number of DFI clock cycles that the dfi_ctrlupd_req signal must be asserted.

DDR_PCTL_DFITCTRLUPDMAX

Address: Operational Base + offset (0x0284)

DFI tctrlupd_max Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0040	tctrlupd_max Specifies the maximum number of DFI clock cycles that the dfi_ctrlupd_req signal can assert.

DDR_PCTL_DFITCTRLUPDDLY

Address: Operational Base + offset (0x0288)

DFI tcctrlupddly Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x8	tctrlupd_dly Delay in DFI clock cycles between time a PCTL-initiated update could be started and time PCTL-initiated update actually starts (dfi_ctrlupd_req going high).

DDR_PCTL_DFIUPDCFG

Address: Operational Base + offset (0x0290)

DFI Update Configuration Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x1	dfi_phyupd_en Enables the support for acknowledging PHY-initiated updates: 1'b0: Disabled 1'b1: Enabled
0	RW	0x1	dfi_ctrlupd_en Enables the generation of PCTL-initiated updates: 1'b0: Disabled 1'b1: Enabled

DDR_PCTL_DFITREFMSKI

Address: Operational Base + offset (0x0294)

DFI Masked Refresh Interval

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	trefmski Time period of the masked Refresh interval This value is only used if TREFI==0. Defines the time period (in 100ns units) of the masked Refresh (REFMSK) interval. The actual time period defined is DFITREFMSKI* TOGCNT100N * internal timers clock period.

DDR_PCTL_DFITCTRLUPDI

Address: Operational Base + offset (0x0298)

DFI tcctrlupd_interval Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	tctrlupd_interval DFI PCTL-initiated updates interval, measured in terms of Refresh interval units. If TREFI!=0, the time period is defined as DFITCTRLUPDI*TREFI * TOGCNT100N * internal timers clock period. If TREFI==0 and DFITREFMSKI!=0, the period changes to DFITCTRLUPDI*DFITREFMSKI* * TOGCNT100N * internal timers clock period. Programming a value of 0 is the same as programming a value of 1; for instance, a PCTL-initiated update occurs every Refresh interval.

DDR_PCTL_DFITRCFG0

Address: Operational Base + offset (0x02ac)

DFI Training Configuration 0 Register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	dfi_wrlvl_rank_sel Determines the value to drive on the output signal dfi_wrlvl_cs_n. The value on dfi_wrlvl_cs_n is the inverse of the setting in this field.
15:13	RO	0x0	reserved
12:4	RW	0x000	dfi_rdlvl_edge Determines the value to drive on the output signal dfi_rdlvl_edge. The value on dfi_rdlvl_edge is the same as the setting in this field.
3:0	RW	0x0	dfi_rdlvl_rank_sel Determines the value to drive on the output signal dfi_rdlvl_cs_n. The value on dfi_rdlvl_cs_n is the inverse of the setting in this field.

DDR_PCTL_DFITRSTAT0

Address: Operational Base + offset (0x02b0)

DFI Training Status 0 Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	RO	0x0	dfi_wrlvl_mode Reports the value of the input signal dfi_wrlvl_mode.
15:10	RO	0x0	reserved
9:8	RO	0x0	dfi_rdlvl_gate_mode Reports the value of the input signal dfi_rdlvl_gate_mode.
7:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RO	0x0	dfi_rdlvl_mode Reports the value of the input signal dfi_rdlvl_mode.

DDR_PCTL_DFITRWRLVLEN

Address: Operational Base + offset (0x02b4)

DFI Training dfi_wrlvl_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_wrlvl_en Determines the value to drive on the output signal dfi_wrlvl_en.

DDR_PCTL_DFITRRDLVLEN

Address: Operational Base + offset (0x02b8)

DFI Training dfi_rdlvl_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_rdlvl_en Determines the value to drive on the output signal dfi_rdlvl_en.

DDR_PCTL_DFITRRDLVLGATEEN

Address: Operational Base + offset (0x02bc)

DFI Training dfi_rdlvl_gate_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_rdlvl_gate_en Determines the value to drive on the output signal dfi_rdlvl_gate_en.

DDR_PCTL_DFISTSTAT0

Address: Operational Base + offset (0x02c0)

DFI Status Status 0 Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:16	RO	0x000	dfi_data_byte_disable Reports the value of the output signal dfi_data_byte_disable.
15:6	RO	0x0	reserved
5:4	RO	0x0	dfi_freq_ratio Reports the value of the output signal dfi_freq_ratio.
3:2	RO	0x0	reserved
1	RO	0x0	dfi_init_start Reports the value of the output signal dfi_init_start.

Bit	Attr	Reset Value	Description
0	RO	0x0	dfi_init_complete Reports the value of the input signal dfi_init_complete.

DDR_PCTL_DFISTCFG0

Address: Operational Base + offset (0x02c4)

DFI Status Configuration 0 Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	dfi_data_byte_disable_en Enables the driving of the dfi_data_byte_disable signal. The value driven on dfi_data_byte_disable is dependent on the setting of PPCFG register. 1'b0: Drive dfi_data_byte_disable to default value of all zeroes. 1'b1: Drive dfi_data_byte_disable according to value as defined by PPCFG register setting. Note: should be set to 1only after PPCFG is correctly set.
1	RW	0x0	dfi_freq_ratio_en Enables the driving of the dfi_freq_ratio signal. When enabled, the dfi_freq_ratio value driven is dependent on configuration parameter PCTL_FREQ_RATIO: 2'b00 is driven when PCTL_FREQ_RATIO=1; 2'b01 is driven when PCTL_FREQ_RATIO=2. 1'b0: Drive dfi_freq_ratio to default value of 2'b00. 1'b1: Drive dfi_freq_ratio value according to how configuration parameter is set.
0	RW	0x0	dfi_init_start Sets the value of the dfi_init_start signal. 1'b0: dfi_init_start is driven low 1'b1: dfi_init_start is driven high

DDR_PCTL_DFISTCFG1

Address: Operational Base + offset (0x02c8)

DFI Status Configuration 1 Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	dfi_dram_clk_disable_en Enables support of the dfi_dram_clk_disable signal with Self Refresh (SR). 1'b0: Disable dfi_dram_clk_disable support in relation to SR 1'b1: Enable dfi_dram_clk_disable support in relation to SR

DDR_PCTL_DFITDRAMCLKEN

Address: Operational Base + offset (0x02d0)

DFI tdram_clk_enable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tdram_clk_enable Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phasealigned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFITDRAMCLKDIS

Address: Operational Base + offset (0x02d4)

DFI tdram_clk_disable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tdram_clk_disable Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phasealigned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFISTCFG2

Address: Operational Base + offset (0x02d8)

DFI Status Configuration 2 Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	parity_en Enables the DFI parity generation feature (driven on output signal dfi_parity_in) 1'b0: Disable DFI parity generation 1'b1: Enable DFI parity generation
0	RW	0x0	parity_intr_en Enable interrupt generation for DFI parity error (from input signal dfi_parity_error). 1'b0: Disable interrupt 1'b1: Enable interrupt

DDR_PCTL_DFISTPARCLR

Address: Operational Base + offset (0x02dc)

DFI Status Parity Clear Register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RWSC	0x0	parity_log_clr Set this bit to 1'b1 to clear the DFI Status Parity Log register (DFISTPARLOG). 1'b0: Do not clear DFI status Parity Log register 1'b1: Clear DFI status Parity Log register
0	RWSC	0x0	parity_intr_clr Set this bit to 1'b1 to clear the interrupt generated by an DFI parity error (as enabled by DFISTCFG2.parity_intr_en). It also clears the INTRSTAT.parity_intr register field. It is automatically cleared by hardware when the interrupt has been cleared.

DDR_PCTL_DFISTPARLOG

Address: Operational Base + offset (0x02e0)

DFI Status Parity Log Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	parity_err_cnt Increments any time the DFI parity logic detects a parity error(s) (on dfi_parity_error).

DDR_PCTL_DFILPCFG0

Address: Operational Base + offset (0x02f0)

DFI Low Power Configuration 0 Register

Bit	Attr	Reset Value	Description
31:28	RW	0x0	dfi_lp_wakeup_dpd Value to drive on dfi_lp_wakeup signal when Deep Power Down mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000: 16 cycles 4'b0001: 32 cycles 4'b0010: 64 cycles 4'b0011: 128 cycles 4'b0100: 256 cycles 4'b0101: 512 cycles 4'b0110: 1024 cycles 4'b0111: 2048 cycles 4'b1000: 4096 cycles 4'b1001: 8192 cycles 4'b1010: 16384 cycles 4'b1011: 32768 cycles 4'b1100: 65536 cycles 4'b1101: 131072 cycles 4'b1110: 262144 cycles 4'b1111: Unlimited
27:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24	RW	0x0	dfi_lp_en_dpd Enables DFI Low Power interface handshaking during Deep Power Down Entry/Exit. 1'b0: Disabled 1'b1: Enabled
23:20	RO	0x0	reserved
19:16	RW	0x7	dfi_tlp_resp Setting for tlp_resp time. Same value is used for both Power Down and Self refresh and Deep Power Down modes. DFI 2.1 specification, recommends using value of 7 always.
15:12	RW	0x0	dfi_lp_wakeup_sr Value to drive on dfi_lp_wakeup signal when Self Refresh mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000: 16 cycles 4'b0001: 32 cycles 4'b0010: 64 cycles 4'b0011: 128 cycles 4'b0100: 256 cycles 4'b0101: 512 cycles 4'b0110: 1024 cycles 4'b0111: 2048 cycles 4'b1000: 4096 cycles 4'b1001: 8192 cycles 4'b1010: 16384 cycles 4'b1011: 32768 cycles 4'b1100: 65536 cycles 4'b1101: 131072 cycles 4'b1110: 262144 cycles 4'b1111: Unlimited
11:9	RO	0x0	reserved
8	RW	0x0	dfi_lp_en_sr Enables DFI Low Power interface handshaking during Self Refresh Entry/Exit. 1'b0: Disabled 1'b1: Enabled

Bit	Attr	Reset Value	Description
7:4	RW	0x0	<p>dfi_ip_wakeup_pd Value to drive on dfi_ip_wakeup signal when Power Down mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000: 16 cycles 4'b0001: 32 cycles 4'b0010: 64 cycles 4'b0011: 128 cycles 4'b0100: 256 cycles 4'b0101: 512 cycles 4'b0110: 1024 cycles 4'b0111: 2048 cycles 4'b1000: 4096 cycles 4'b1001: 8192 cycles 4'b1010: 16384 cycles 4'b1011: 32768 cycles 4'b1100: 65536 cycles 4'b1101: 131072 cycles 4'b1110: 262144 cycles 4'b1111: Unlimited</p>
3:1	RO	0x0	reserved
0	RW	0x0	<p>dfi_ip_en_pd Enables DFI Low Power interface handshaking during Power Down Entry/Exit. 1'b0: Disabled 1'b1: Enabled</p>

DDR_PCTL_DFITRWRLVLRESP0

Address: Operational Base + offset (0x0300)

DFI Training dfi_wrlvl_resp Status 0 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>dfi_wrlvl_resp0 Reports the status of the dif_wrlvl_resp[31:0] signal.</p>

DDR_PCTL_DFITRWRLVLRESP1

Address: Operational Base + offset (0x0304)

DFI Training dfi_wrlvl_resp Status 1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>dfi_wrlvl_resp1 Reports the status of the dif_wrlvl_resp[63:32] signal.</p>

DDR_PCTL_DFITRWRLVLRESP2

Address: Operational Base + offset (0x0308)

DFI Training dfi_wrlvl_resp Status 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	dfi_wrlvl_resp2 Reports the status of the dif_wrlvl_resp[71:64] signal.

DDR_PCTL_DFITRRDLVLRESP0

Address: Operational Base + offset (0x030c)

DFI Training dfi_rdlvl_resp Status 0 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rdlvl_resp0 Reports the status of the dif_rdlvl_resp[31:0] signal.

DDR_PCTL_DFITRRDLVLRESP1

Address: Operational Base + offset (0x0310)

DFI Training dfi_rdlvl_resp Status 1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rdlvl_resp1 Reports the status of the dif_rdlvl_resp[63:32] signal.

DDR_PCTL_DFITRRDLVLRESP2

Address: Operational Base + offset (0x0314)

DFI Training dfi_rdlvl_resp Status 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	dfi_rdlvl_resp2 Reports the status of the dif_rdlvl_resp[71:64] signal.

DDR_PCTL_DFITWRWLVLDELAY0

Address: Operational Base + offset (0x0318)

DFI Training dfi_wrlvl_delay Configuration 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_wrlvl_delay0 Sets the value to be driven on the signal dfi_wrlvl_delay_x[31:0].

DDR_PCTL_DFITWRWLVLDELAY1

Address: Operational Base + offset (0x031c)

DFI Training dfi_wrlvl_delay Configuration 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_wrlvl_delay1 Sets the value to be driven on the signal dfi_wrlvl_delay_x[63:32].

DDR_PCTL_DFITWRWLVLDELAY2

Address: Operational Base + offset (0x0320)

DFI Training dfi_wrlvl_delay Configuration 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_wrlvl_delay2 Sets the value to be driven on the signal dfi_wrlvl_delay_x[71:64].

DDR_PCTL_DFITRRDLVLDELAY0

Address: Operational Base + offset (0x0324)

DFI Training dfi_rdlvl_delay Configuration 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_delay0 Sets the value to be driven on the signal dfi_rdlvl_delay_x[31:0].

DDR_PCTL_DFITRRDLVLDELAY1

Address: Operational Base + offset (0x0328)

DFI Training dfi_rdlvl_delay Configuration 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_delay1 Sets the value to be driven on the signal dfi_rdlvl_delay_x[63:32].

DDR_PCTL_DFITRRDLVLDELAY2

Address: Operational Base + offset (0x032c)

DFI Training dfi_rdlvl_delay Configuration 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_rdlvl_delay2 Sets the value to be driven on the signal dfi_rdlvl_delay_x[71:64].

DDR_PCTL_DFITRRDLVLGATEDELAY0

Address: Operational Base + offset (0x0330)

DFI Training dfi_rdlvl_gate_delay Configuration 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_gate_delay0 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[31:0].

DDR_PCTL_DFITRRDLVLGATEDELAY1

Address: Operational Base + offset (0x0334)

DFI Training dfi_rdlvl_gate_delay Configuration 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_gate_delay1 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[63:32].

DDR_PCTL_DFITRRDLVLGATEDELAY2

Address: Operational Base + offset (0x0338)

DFI Training dfi_rdlvl_gate_delay Configuration 2

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_rdlvl_gate_delay2 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[71:64].

DDR_PCTL_DFITRCMD

Address: Operational Base + offset (0x033c)

DFI Training Command Register

Bit	Attr	Reset Value	Description
31	RWSC	0x0	dfitrcmd_start DFI Training Command Start. When this bit is set to 1, the command operation defined in the dfitrcmd_opcode field is started. This bit is automatically cleared by the PCTL after the command is finished. The application can poll this bit to determine when PCTL is ready to accept another command. This bit cannot be cleared to 1'b0 by software.
30:13	RO	0x0	reserved
12:4	RW	0x000	dfitrcmd_en DFI Training Command Enable. Selects which bits of chosen DFI Training command to drive to 1'b1.
3:2	RO	0x0	reserved
1:0	RW	0x0	dfitrcmd_opcode DFI Training Command Opcode. Select which DFI Training command to generate for one n_clk cycle: 2'b00: dfi_wrlvl_load 2'b01: dfi_wrlvl_strobe 2'b10: dfi_rdlvl_load 2'b11: Reserved.

DDR_PCTL_IPVR

Address: Operational Base + offset (0x03f8)

IP Version Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ip_version ASCII value for each number in the version, followed by a *.

DDR_PCTL_IPTR

Address: Operational Base + offset (0x03fc)

IP Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44574300	ip_type Contains the IP's identification code, which is an ASCII value to identify the component and it is currently set to the string "DWC". This value never changes.

DDRPHY_REG00

Address: Operational Base + offset (0x0000)

DDR PHY register 00

Bit	Attr	Reset Value	Description
31:4	RO	0xf	reserved
3	RW	0x1	soft reset 1, active low
2	RW	0x1	soft reset 0, active low
1:0	RO	0x3	reserved

DDRPHY_REG01

Address: Operational Base + offset (0x0004)

DDR PHY register 01

Bit	Attr	Reset Value	Description
31:2	RO	0x1	reserved
1:0	RW	0x0	PHY working mode: 0x0: ddr3 PHY mode 0x1: ddr2 PHY mode 0x2: lpddr2 PHY mode 0x3: reserved

DDRPHY_REG02

Address: Operational Base + offset (0x0008)

DDR PHY register 02

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	RXMEM calibration control, active high

DDRPHY_REG03

Address: Operational Base + offset (0x000c)

DDR PHY register 03

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x2	right channel A read ODT delay
2:0	RW	0x2	left channel A read ODT delay

DDRPHY_REG04

Address: Operational Base + offset (0x0010)

DDR PHY register 04

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x2	right channel B read ODT delay

Bit	Attr	Reset Value	Description
2:0	RW	0x2	left channel B read ODT delay

DDRPHY_REG0B

Address: Operational Base + offset (0x002c)

DDR PHY register 0B

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x6	CL value DDR3/LPDDR2 CAS Latency
3:0	RW	0x0	AL value DDR3/LPDDR2 additive latency

DDRPHY_REG0C

Address: Operational Base + offset (0x0030)

DDR PHY register 0C

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	CWL value DDR3/LPDDR2 write latency

DDRPHY_REG11

Address: Operational Base + offset (0x0044)

DDR PHY register 11

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	CMD P RCOMP, except for CK/CKB. The larger the value, the stronger the drive strength
3:0	RW	0xa	CMD N RCOMP, except for CK/CKB. The larger the value, the stronger the drive strength

DDRPHY_REG12

Address: Operational Base + offset (0x0048)

DDR PHY register 12

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x1	CMD weak pull up enable, active LOW
0	RW	0x0	CMD weak pull down enable, active HIGH

DDRPHY_REG13

Address: Operational Base + offset (0x0048)

DDR PHY register 12

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	CMD DLL clock phase select in bypass mode 1'b0: no delay 1'b1: 180 deg delay

Bit	Attr	Reset Value	Description
3	RW	0x1	CMD DLL enable 1'b0: disable 1'b1: enable
2:0	RW	0x4	CMD and ADDRESS DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG14

Address: Operational Base + offset (0x0050)

DDR PHY register 14

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	CK DLL clock phase select in bypass mode 1'b0: no delay 1'b1: 180 deg delay
2:0	RW	0x0	CK DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG16

Address: Operational Base + offset (0x0058)

DDR PHY register 16

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	CK/CKB PRCOMP. The larger the value, the stronger the drive strength
3:0	RW	0xa	CK/CKB NRCOMP. The larger the value, the stronger the drive strength

DDRPHY_REG20

Address: Operational Base + offset (0x0080)

DDR PHY register 20

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	Left channel A PRCOMP. The larger the value, the stronger the drive strength in the scope from A_DQ0 to A_DQ7

Bit	Attr	Reset Value	Description
3:0	RW	0xa	Left channel A NRCOMP. The larger the value, the stronger the drive strength in the scope from A_DQ0 to A_DQ7

DDRPHY_REG21

Address: Operational Base + offset (0x0084)

DDR PHY register 21

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	Left channel A read pull-up ODT. The larger the value, the smaller the pull-up resistance in the scope from A_DQ0 to A_DQ7
3:0	RW	0xa	Left channel A read pull-down ODT. The larger the value, the smaller the pull-down resistance in the scope from A_DQ0 to A_DQ7

DDRPHY_REG26

Address: Operational Base + offset (0x0098)

DDR PHY register 26

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	left channel A write DQ DLL phase select 1'b0: no delay 1'b1: 180 deg delay
3	RW	0x1	left channel A write DQ DLL enable 1'b0: disable 1'b1: enable
2:0	RW	0x4	left channel A write DQ DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG27

Address: Operational Base + offset (0x009c)

DDR PHY register 27

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	left channel A write DQS DLL phase select 1'b0: no delay 1'b1: 180 deg delay

Bit	Attr	Reset Value	Description
2:0	RW	0x0	left channel A write DQS DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG28

Address: Operational Base + offset (0x00a0)
 DDR PHY register 28

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	left channel A read DQS DLL delay 2'd0: no delay 2'd1: 22.5 deg delay 2'd2: 45 deg delay 2'd3: 67.5 deg delay

DDRPHY_REG30

Address: Operational Base + offset (0x00c0)
 DDR PHY register 30

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	Right channel A PRCOMP. The larger the value, the stronger the drive strength in the scope from A_DQ8 to A_DQ15
3:0	RW	0xa	Right channel A NRCOMP. The larger the value, the stronger the drive strength in the scope from A_DQ8 to A_DQ15

DDRPHY_REG31

Address: Operational Base + offset (0x00c4)
 DDR PHY register 31

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	Right channel A read pull-up ODT. The larger the value, the smaller the pull-up resistance in the scope from A_DQ8 to A_DQ15
3:0	RW	0xa	Right channel A read pull-down ODT. The larger the value, the smaller the pull-down resistance in the scope from A_DQ8 to A_DQ15

DDRPHY_REG36

Address: Operational Base + offset (0x00d4)

DDR PHY register 36

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	Right channel A write DQ DLL phase select 1'b0: no delay 1'b1: 180 deg delay
3	RW	0x1	Right channel A write DQ DLL enable 1'b0: disable 1'b1: enable
2:0	RW	0x4	Right channel A write DQ DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG37

Address: Operational Base + offset (0x00d8)

DDR PHY register 37

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	Right channel A write DQS DLL phase select 1'b0: no delay 1'b1: 180 deg delay
2:0	RW	0x0	Right channel A write DQS DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG38

Address: Operational Base + offset (0x00dc)

DDR PHY register 38

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	Right channel A read DQS DLL delay 2'd0: no delay 2'd1: 22.5 deg delay 2'd2: 45 deg delay 2'd3: 67.5 deg delay

DDRPHY_REG40

Address: Operational Base + offset (0x0100)

DDR PHY register 40

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	Left channel B PRCOMP. The larger the value, the stronger the drive strength in the scope from B_DQ0 to B_DQ7
3:0	RW	0xa	Left channel B NRCOMP. The larger the value, the stronger the drive strength in the scope from B_DQ0 to B_DQ7

DDRPHY_REG41

Address: Operational Base + offset (0x0104)

DDR PHY register 41

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	Left channel B read pull-up ODT. The larger the value, the smaller the pull-up resistance in the scope from B_DQ0 to B_DQ7
3:0	RW	0xa	Left channel B read pull-down ODT. The larger the value, the smaller the pull-down resistance in the scope from B_DQ0 to B_DQ7

DDRPHY_REG46

Address: Operational Base + offset (0x0118)

DDR PHY register 46

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	left channel B write DQ DLL phase select 1'b0: no delay 1'b1: 180 deg delay
3	RW	0x1	left channel B write DQ DLL enable 1'b0: disable 1'b1: enable
2:0	RW	0x4	left channel B write DQ DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG47

Address: Operational Base + offset (0x011c)

DDR PHY register 47

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	left channel B write DQS DLL phase select 1'b0: no delay 1'b1: 180 deg delay
2:0	RW	0x0	left channel B write DQS DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG48

Address: Operational Base + offset (0x0120)

DDR PHY register 48

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	left channel B read DQS DLL delay 2'd0: no delay 2'd1: 22.5 deg delay 2'd2: 45 deg delay 2'd3: 67.5 deg delay

DDRPHY_REG50

Address: Operational Base + offset (0x0140)

DDR PHY register 50

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	Right channel B PRCOMP. The larger the value, the stronger the drive strength in the scope from B_DQ8 to B_DQ15
3:0	RW	0xa	Right channel B NRCOMP. The larger the value, the stronger the drive strength in the scope from B_DQ8 to B_DQ15

DDRPHY_REG51

Address: Operational Base + offset (0x0144)

DDR PHY register 51

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0xa	Right channel B read pull-up ODT. The larger the value, the smaller the pull-up resistance in the scope from B_DQ8 to B_DQ15
3:0	RW	0xa	Right channel B read pull-down ODT. The larger the value, the smaller the pull-down resistance in the scope from B_DQ8 to B_DQ15

DDRPHY_REG56

Address: Operational Base + offset (0x0158)
 DDR PHY register 56

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	Right channel B write DQ DLL phase select 1'b0: no delay 1'b1: 180 deg delay
3	RW	0x1	Right channel B write DQ DLL enable 1'b0: disable 1'b1: enable
2:0	RW	0x4	Right channel B write DQ DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG57

Address: Operational Base + offset (0x015c)
 DDR PHY register 57

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	Right channel B write DQS DLL phase select 1'b0: no delay 1'b1: 180 deg delay
2:0	RW	0x0	Right channel B write DQS DLL delay 3'd0: no delay 3'd1: 22.5 deg delay 3'd2: 45 deg delay 3'd3: 67.5 deg delay 3'd4: 90 deg delay 3'd5: 112.5 deg delay 3'd6: 135 deg delay 3'd7: 157.5 deg delay

DDRPHY_REG58

Address: Operational Base + offset (0x0160)
 DDR PHY register 58

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	Right channel B read DQS DLL delay 2'd0: no delay 2'd1: 22.5 deg delay 2'd2: 45 deg delay 2'd3: 67.5 deg delay

DDRPHY_REGB0

Address: Operational Base + offset (0x02c0)
 DDR PHY register B0

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A1 de-skew
3:0	RW	0x7	A0 de-skew

DDRPHY_REGB1

Address: Operational Base + offset (0x02c4)
 DDR PHY register B1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A3 de-skew
3:0	RW	0x7	A2 de-skew

DDRPHY_REGB2

Address: Operational Base + offset (0x02c8)
 DDR PHY register B2

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A5 de-skew
3:0	RW	0x7	A4 de-skew

DDRPHY_REGB3

Address: Operational Base + offset (0x02cc)
 DDR PHY register B3

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A7 de-skew
3:0	RW	0x7	A6 de-skew

DDRPHY_REGB4

Address: Operational Base + offset (0x02d0)
 DDR PHY register B4

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A9 de-skew
3:0	RW	0x7	A8 de-skew

DDRPHY_REGB5

Address: Operational Base + offset (0x02d4)
 DDR PHY register B5

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A11 de-skew

Bit	Attr	Reset Value	Description
3:0	RW	0x7	A10 de-skew

DDRPHY_REGB6

Address: Operational Base + offset (0x02d8)

DDR PHY register B6

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A13 de-skew
3:0	RW	0x7	A12 de-skew

DDRPHY_REGB7

Address: Operational Base + offset (0x02dc)

DDR PHY register B7

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A15 de-skew
3:0	RW	0x7	A14 de-skew

DDRPHY_REGB8

Address: Operational Base + offset (0x02e0)

DDR PHY register B8

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B1 de-skew
3:0	RW	0x7	B0 de-skew

DDRPHY_REGB9

Address: Operational Base + offset (0x02e4)

DDR PHY register B9

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	RASB de-skew
3:0	RW	0x7	B2 de-skew

DDRPHY_REGBA

Address: Operational Base + offset (0x02e8)

DDR PHY register BA

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	WEB de-skew
3:0	RW	0x7	CASB de-skew

DDRPHY_REGBB

Address: Operational Base + offset (0x02ec)

DDR PHY register BB

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CKB de-skew
3:0	RW	0x7	CK de-skew

DDRPHY_REGBC

Address: Operational Base + offset (0x02f0)

DDR PHY register BC

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CKE de-skew
3:0	RW	0x7	ODT0 de-skew

DDRPHY_REGBD

Address: Operational Base + offset (0x02f4)

DDR PHY register BD

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CSB0 de-skew
3:0	RW	0x7	RESETN de-skew

DDRPHY_REGBE

Address: Operational Base + offset (0x02f8)

DDR PHY register BE

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CSB1 de-skew
3:0	RW	0x7	ODT1 de-skew

DDRPHY_REGC0

Address: Operational Base + offset (0x0300)

DDR PHY register C0

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DM0 RX de-skew
3:0	RW	0x7	A_DM0 TX de-skew

DDRPHY_REGC1

Address: Operational Base + offset (0x0304)

DDR PHY register C1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ0 RX de-skew
3:0	RW	0x7	A_DQ0 TX de-skew

DDRPHY_REGC2

Address: Operational Base + offset (0x0308)
 DDR PHY register C2

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ1 RX de-skew
3:0	RW	0x7	A_DQ1 TX de-skew

DDRPHY_REGC3

Address: Operational Base + offset (0x030c)
 DDR PHY register C3

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ2 RX de-skew
3:0	RW	0x7	A_DQ2 TX de-skew

DDRPHY_REGC4

Address: Operational Base + offset (0x0310)
 DDR PHY register C4

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ3 RX de-skew
3:0	RW	0x7	A_DQ3 TX de-skew

DDRPHY_REGC5

Address: Operational Base + offset (0x0314)
 DDR PHY register C5

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ4 RX de-skew
3:0	RW	0x7	A_DQ4 TX de-skew

DDRPHY_REGC6

Address: Operational Base + offset (0x0318)
 DDR PHY register C6

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ5 RX de-skew
3:0	RW	0x7	A_DQ5 TX de-skew

DDRPHY_REGC7

Address: Operational Base + offset (0x031c)
 DDR PHY register C7

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ6 RX de-skew
3:0	RW	0x7	A_DQ6 TX de-skew

DDRPHY_REGC8

Address: Operational Base + offset (0x0320)
 DDR PHY register C8

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ7 RX de-skew
3:0	RW	0x7	A_DQ7 TX de-skew

DDRPHY_REGC9

Address: Operational Base + offset (0x0324)
 DDR PHY register C9

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQS0 RX de-skew
3:0	RW	0x7	A_DQS0 TX de-skew

DDRPHY_REGCA

Address: Operational Base + offset (0x0328)
 DDR PHY register CA

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x7	A_DQSB0 TX de-skew

DDRPHY_REGCB

Address: Operational Base + offset (0x032c)
 DDR PHY register CB

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DM1 RX de-skew
3:0	RW	0x7	A_DM1 TX de-skew

DDRPHY_REGCC

Address: Operational Base + offset (0x0330)
 DDR PHY register CC

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ8 RX de-skew
3:0	RW	0x7	A_DQ8 TX de-skew

DDRPHY_REGCD

Address: Operational Base + offset (0x0334)
 DDR PHY register CD

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ9 RX de-skew
3:0	RW	0x7	A_DQ9 TX de-skew

DDRPHY_REGCE

Address: Operational Base + offset (0x0338)

DDR PHY register CE

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ10 RX de-skew
3:0	RW	0x7	A_DQ10 TX de-skew

DDRPHY_REGCF

Address: Operational Base + offset (0x033c)

DDR PHY register CF

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ11 RX de-skew
3:0	RW	0x7	A_DQ11 TX de-skew

DDRPHY_REGD0

Address: Operational Base + offset (0x0340)

DDR PHY register D0

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ12 RX de-skew
3:0	RW	0x7	A_DQ12 TX de-skew

DDRPHY_REGD1

Address: Operational Base + offset (0x0344)

DDR PHY register D1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ13 RX de-skew
3:0	RW	0x7	A_DQ13 TX de-skew

DDRPHY_REGD2

Address: Operational Base + offset (0x0348)

DDR PHY register D2

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQ14 RX de-skew
3:0	RW	0x7	A_DQ14 TX de-skew

DDRPHY_REGD3

Address: Operational Base + offset (0x034c)

DDR PHY register D3

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:4	RW	0x7	A_DQ15 RX de-skew
3:0	RW	0x7	A_DQ15 TX de-skew

DDRPHY_REGD4

Address: Operational Base + offset (0x0350)

DDR PHY register D4

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A_DQS1 RX de-skew
3:0	RW	0x7	A_DQS1 TX de-skew

DDRPHY_REGD5

Address: Operational Base + offset (0x0354)

DDR PHY register D5

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x7	A_DQSB1 TX de-skew

DDRPHY_REGD6

Address: Operational Base + offset (0x0358)

DDR PHY register D6

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DM0 RX de-skew
3:0	RW	0x7	B_DM0 TX de-skew

DDRPHY_REGD7

Address: Operational Base + offset (0x035c)

DDR PHY register D7

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ0 RX de-skew
3:0	RW	0x7	B_DQ0 TX de-skew

DDRPHY_REGD8

Address: Operational Base + offset (0x0360)

DDR PHY register D8

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ1 RX de-skew
3:0	RW	0x7	B_DQ1 TX de-skew

DDRPHY_REGD9

Address: Operational Base + offset (0x0364)

DDR PHY register D9

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ2 RX de-skew
3:0	RW	0x7	B_DQ2 TX de-skew

DDRPHY_REGDA

Address: Operational Base + offset (0x0368)

DDR PHY register DA

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ3 RX de-skew
3:0	RW	0x7	B_DQ3 TX de-skew

DDRPHY_REGDB

Address: Operational Base + offset (0x036c)

DDR PHY register DB

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ4 RX de-skew
3:0	RW	0x7	B_DQ4 TX de-skew

DDRPHY_REGDC

Address: Operational Base + offset (0x0370)

DDR PHY register DC

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ5 RX de-skew
3:0	RW	0x7	B_DQ5 TX de-skew

DDRPHY_REGDD

Address: Operational Base + offset (0x0374)

DDR PHY register DD

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ6 RX de-skew
3:0	RW	0x7	B_DQ6 TX de-skew

DDRPHY_REGDE

Address: Operational Base + offset (0x0378)

DDR PHY register DE

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ7 RX de-skew
3:0	RW	0x7	B_DQ7 TX de-skew

DDRPHY_REGDF

Address: Operational Base + offset (0x037c)
 DDR PHY register DF

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQS0 RX de-skew
3:0	RW	0x7	B_DQS0 TX de-skew

DDRPHY_REGE0

Address: Operational Base + offset (0x0380)
 DDR PHY register E0

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x7	B_DQSB0 TX de-skew

DDRPHY_REGE1

Address: Operational Base + offset (0x0384)
 DDR PHY register E1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DM1 RX de-skew
3:0	RW	0x7	B_DM1 TX de-skew

DDRPHY_REGE2

Address: Operational Base + offset (0x0388)
 DDR PHY register E2

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ8 RX de-skew
3:0	RW	0x7	B_DQ8 TX de-skew

DDRPHY_REGE3

Address: Operational Base + offset (0x038c)
 DDR PHY register E3

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ9 RX de-skew
3:0	RW	0x7	B_DQ9 TX de-skew

DDRPHY_REGE4

Address: Operational Base + offset (0x0390)
 DDR PHY register E4

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ10 RX de-skew
3:0	RW	0x7	B_DQ10 TX de-skew

DDRPHY_REGE5

Address: Operational Base + offset (0x0394)

DDR PHY register E5

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ11 RX de-skew
3:0	RW	0x7	B_DQ11 TX de-skew

DDRPHY_REGE6

Address: Operational Base + offset (0x0398)

DDR PHY register E6

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ12 RX de-skew
3:0	RW	0x7	B_DQ12 TX de-skew

DDRPHY_REGE7

Address: Operational Base + offset (0x039c)

DDR PHY register E7

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ13 RX de-skew
3:0	RW	0x7	B_DQ13 TX de-skew

DDRPHY_REGE8

Address: Operational Base + offset (0x03a0)

DDR PHY register E8

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ14 RX de-skew
3:0	RW	0x7	B_DQ14 TX de-skew

DDRPHY_REGE9

Address: Operational Base + offset (0x03a4)

DDR PHY register E9

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQ15 RX de-skew
3:0	RW	0x7	B_DQ15 TX de-skew

DDRPHY_REGEA

Address: Operational Base + offset (0x03a8)

DDR PHY register EA

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B_DQS1 RX de-skew
3:0	RW	0x7	B_DQS1 TX de-skew

DDRPHY_REGEB

Address: Operational Base + offset (0x03ac)

DDR PHY register EB

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x7	B_DQSB1 TX de-skew

DDRPHY_REGFA

Address: Operational Base + offset (0x03e8)

DDR PHY register FA

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	Channel B High 8bit dqs gate sample dqs value(idqs)
2	RO	0x0	Channel B Low 8bit dqs gate sample dqs value(idqs)
1	RO	0x0	Channel A High 8bit dqs gate sample dqs value(idqs)
0	RO	0x0	Channel A Low 8bit dqs gate sample dqs value(idqs)

DDRPHY_REGFB

Address: Operational Base + offset (0x03ec)

DDR PHY register FB

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RO	0x0	Calibration get the dll configure channel A low 8bit
3	RO	0x0	Calibration get the ophsel configure channel A low 8bit
2:0	RO	0x0	Calibration get the cyclesel configure channle A low 8bit

DDRPHY_REGFC

Address: Operational Base + offset (0x03f0)

DDR PHY register FC

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RO	0x0	Calibration get the dll configure channel A high 8bit
3	RO	0x0	Calibration get the ophsel configure channel A high 8bit
2:0	RO	0x0	Calibration get the cyclesel configure channle A high 8bit

DDRPHY_REGFD

Address: Operational Base + offset (0x03f4)

DDR PHY register FD

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RO	0x0	Calibration get the dll configure channel B low 8bit
3	RO	0x0	Calibration get the ophsel configure channel B low 8bit
2:0	RO	0x0	Calibration get the cyclesel configure channe B low 8bit

DDRPHY_REGFE

Address: Operational Base + offset (0x03f8)

DDR PHY register FE

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RO	0x0	Calibration get the dll configure channel B high 8bit
3	RO	0x0	Calibration get the ophsel configure channel B high 8bit
2:0	RO	0x0	Calibration get the cyclesel configure channe B high 8bit

DDRPHY_REGFF

Address: Operational Base + offset (0x03fc)

DDR PHY register FF

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	Channel B High 8bit Calibration done
2	RO	0x0	Channel B Low 8bit Calibration done
1	RO	0x0	Channel A High 8bit Calibration done
0	RO	0x0	Channel A Low 8bit Calibration done

Note:

- (1) left channel A signals: A_DQS0, A_DQSB0, A_DQ7~A_DQ0, A_DM0
right channel A signals: A_DQS1, A_DQSB1, A_DQ15~A_DQ8, A_DM1
left channel B signals: B_DQS0, B_DQSB0, B_DQ7~B_DQ0, B_DM0
right channel B signals: B_DQS1, B_DQSB1, B_DQ15~B_DQ8, B_DM1
The delayed phase is in 1X clock domain, whose frequency is equivalent to SDRAM clock.

(2) per-bit de-skew, for detailed information, refer to previous section 10.6.8.

10.5 Interface Description

DDR IOs are listed as following Table.

Table 10-1 DDR IO description

Pin Name	Description
CK	Active-high clock signal to the memory device.
CK_B	Active-low clock signal to the memory device.
CKE	Active-high clock enable signal to the memory device for two

	chip select.
CS_Bi (i=0,1)	Active-low chip select signal to the memory device. ATThere are two chip select.
RAS_B	Active-low row address strobe to the memory device.
CAS_B	Active-low column address strobe to the memory device.
WE_B	Active-low write enable strobe to the memory device.
BA[2:0]	Bank address signal to the memory device.
A[15:0]	Address signal to the memory device.
DQ[15:0]	Bidirectional data line to the memory device.
DQS[1:0]	Active-high bidirectional data strobes to the memory device.
DQS_B[1:0]	Active-low bidirectional data strobes to the memory device.
DM[1:0]	Active-low data mask signal to the memory device.
ODTi (i=0,1)	On-Die Termination output signal for two chip select.
RESET	DDR3 reset signal.

10.6 Application Notes

10.6.1 State transition of PCTL

To operate PCTL, the programmer must be familiar with the available operational states and how to transition to each state from the current state.

Every software programmable register is accessible only during certain operational states. For information about what registers are accessible in each state, refer to “Software Registers,” which provides this information in each register description. The general rule is that the PCTL must be in the Init_mem or Config states to successfully write most of the registers.

The following tables provide the programming sequences for moving to the various states of the state machine.

Moving to the Init_mem State

Step	Application	PCTL
1	Read STAT register	Returns the current PCTL state.
2	If STAT.ctl_stat = Init_mem, go to END.	
3	If STAT.ctl_stat =Config, go to Step9.	
4	If STAT.ctl_stat =Access, go to Step8.	
5	If STAT.ctl_stat = Low_power, go to Step7.	
6	Goto Step1.	PCTL is in a Transitional state and not in any of the previous operational states.
7	Write WAKEUP to SCTL.state_cmd and poll STAT.ctl_stat = Access.	Issues SRX, moves to the Access state, updates STAT.ctl_stat =Access when complete.
8	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat = Config.	PCTL stalls the NIF; completes any pending transaction; issues PREA if required; moves into the Config state; updates STAT.ctl_stat =Config whencomplete.
9	Write INIT to SCTL.state_cmd and poll STAT.ctl_stat =Init_mem	Moves into the Init_mem state and updates STAT.ctl_stat =Init_mem.

END		PCTL is in Init_mem state.
-----	--	----------------------------

Moving to Config State

Step	Application	PCTL
1	Read STAT register.	Returns the current PCTL state.
2	If STAT.ctl_stat = Config, goto END.	
3	If STAT.ctl_stat = Low_power, go to Step6 .	
4	If STAT.ctl_stat = Init_mem or Access, go to Step7 .	
5	Go to Step1 .	PCTL is in a transitional state and is not in any of the previous operational states.
6	Write WAKEUP to CTL.state_cmd and poll STAT.ctl_stat = Access.	Issues SRX, moves to the Access state, and updates STAT.ctl_stat = Access when complete.
7	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat = Config.	PCTL stalls the NIF; completes any pending transaction; issues PREA if required; moves into the Config state; and updates STAT.ctl_stat = Config when complete.
END		PCTL is in Config state.

Moving to Access State

Step	Application	PCTL
1	Read STAT register	Returns the current PCTL state.
2	If STAT.ctl_stat = Access, go to END.	
3	If STAT.ctl_stat = Config, go to Step9	
4	If STAT.ctl_stat = Init_mem, go to Step8	
5	If STAT.ctl_stat = Low_power, go to Step7 .	
6	Goto Step1 .	PCTL is in a transitional state and is not in any of the previous operational states.
7	Write WAKEUP to SCTL.state_cmd and poll STAT.ctl_stat = Access. Goto END	Issues SRX, moves to the Access state, updates STAT.ctl_stat = Access when complete.
8	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat = Config.	Moves into the Config state, updates STAT.ctl_stat = Config when complete.
9	Write GO to SCTL.state_cmd and poll STAT.ctl_stat = Access.	Moves into the Access state, updates STAT.ctl_stat = Access when complete.
END		PCTL is in Access state.

Moving to Low Power State

Step	Application	PCTL
1	Read STAT register.	Returns current PCTL state.

2	If STAT .ctl_stat =Low_power, go to END.	
3	If STAT .ctl_stat = Access, go to Step9	
4	If STAT .ctl_stat = Config, go to Step8	
5	If STAT .ctl_stat = Init_mem, go to Step7 .	
6	Goto Step1 .	PCTL is in transitional state and is not in any of the previous operational states.
7	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat = Config.	Moves into the Config state, updates STAT.ctl_stat = Config when complete.
8	Write GO to SCTL.state_cmd and poll STAT.ctl_stat = Access.	Moves into the Access state, updates STAT.ctl_stat =Access when complete.
9	Write SLEEP to SCTL.state_cmd and poll STAT.ctl_stat = Low_power.	Issues PDX if necessary; completes any pending transactions; issues PREA command; finally, issues SRE and updates STAT.ctl_stat = Low_power.
END		PCTL is in Low Power state

10.6.2 Initialization

PHY Initialization

DDR PHY power-up reset sequence:

1. PHY Register Reset: reset the PHY register block through presetn;
2. Configure registers : AL ,CL etc. (register address 0x38)
3. System reset: reset the PHY through system_rstn;
4. Soft reset(optional): after system reset, execute the soft reset by changing PHY_REG0 values. Soft reset function is on when PHY_REG0[3:2] =2'b11.
5. Start PHY initialization: after pclk, dfi_clk1x and dfi_clk2x clock signals are stable, initialize the PHY.
6. Start PHY Calibration: after ddr sdram initialization done, you can set PHY calibration start.
7. Close PHY Calibration: after step 5us, you can set close PHY calibration.
8. Start Write and Read.

DDR3 Initialization Sequence

The initialization steps for DDR3 SDRAMs are as follows:

1. Optionally maintain RESET# low for a minimum of either 200 us (power-up initialization) or 100ns (power-on initialization). The DDR PHY drives RESET# low from the beginning of reset assertion and therefore this step may be skipped when DRAM initialization is triggered if enough time may already have expired to satisfy the RESET# low time.
2. After RESET# is de-asserted, wait a minimum of 500 us with CKE low.
3. Apply NOP and drive CKE high.

4. Wait a minimum of tXPR.
5. Issue a load Mode Register 2 (MR2) command.
6. Issue a load Mode Register 3 (MR3) command.
7. Issue a load Mode Register (MR1) command (to set parameters and enable DLL).
8. Issue a load Mode Register (MR0) command to set parameters and reset DLL.
9. Issue ZQ calibration command.
10. Wait 512 SDRAM clock cycles for the DLL to lock (tDLLK) and ZQ calibration (tZQinit) to finish. This wait time is relative to Step 8, i.e. relative to when the DLL reset command was issued onto the SDRAM command bus.

LPDDR2 Initialization Sequence

The initialization steps for LPDDR2 SDRAMs are as follows:

1. Wait a minimum of 100 ns (tINIT1) with CKE driven low.
2. Apply NOP and set CKE high.
3. Wait a minimum of 200 us (tINIT3).
4. Issue a RESET command.
5. Wait a minimum of 1 us + 10 us (tINIT4 + tINIT5).
6. Issue a ZQ calibration command.
7. Wait a minimum of 1 us (tZQINIT).
8. Issue a Write Mode Register to MR1.
9. Issue a Write Mode Register to MR2
10. Issue a Write Mode Register to MR3

10.6.3 Low Power Operation

Low_power state can be entered/exited via following ways:

- Software control of PCTL State machine (highest priority)
- Hardware Low Power Interface (middle priority)
- Auto Self Refresh feature (lowest priority)

Note the priority of requests from Access to Low_power is highlighted above. The STAT.ip_trig register field reports which of the 3 requests caused the entry to Low_power state.

Software control of PCTL State

The application can request via software to enter the memories into Self Refresh state by issuing the SLEEP command by programming SCTL.PCTL responds to the software request by moving into the Low_power operational state and issuing the SRE command to the memories. Note that the Low_power state can only be reached from the Access state.

In a similar fashion, the application requests to exit the memories from Self Refresh by issuing a WAKEUP command by programming SCTL.. PCTL responds to the WAKEUP command issuing SRX and restoring normal NIF address channel operation.

Hardware Low Power Interface

The hardware low power interface can also be used to enter/exit Self Refresh. The functionality is enabled by setting SCFG.hw_low_power_en=1. Once that bit is set, the input c_sysreq has the ability to trigger entry into the Low Power configuration state just like the software methodology (SCTL.state_cmd = SLEEP). A hardware Low Power entry trigger will be ignored/denied if the input c_active_in=1 or n_valid=1. It may be accepted if c_active_in=0 and n_valid=0, depending on the current state of the PCTL. When SCFG.hw_low_power_en=1, the outputs c_sysack and c_active provide feedback as required by the AXI low power interface specification (this interface's operation is defined by the AXI specification). c_sysack acknowledges the request to go into the Low_power state, and c_active indicates when the PCTL is actually in the Low_power state.

The c_active output could also be used by an external Low Power controller to decide when to request a transition to low power. When MCFG1.hw_idle > 0, c_active = 1'b0 indicates that the NIF has been idle for at least MCFG1.hw_idle * 32 * n_clk cycles while in the Access state.

When in low power the c_active output can be used by an external Low Power controller to trigger a low power exit. c_active will be driven high when either c_active_in or n_valid are high. The path from c_active_in and n_valid to c_active is asynchronous so even if the clocks have been removed c_active will assert. The Low Power controller should re-enable the clocks when c_active is driven high while in the Low_power state.

Auto Power Down/Self Refresh

The Power Down and/or Self Refresh sequence is automatically started by PCTL when the NIF address channel is idle for a number of cycles, depending on the programmed value in MCFG.pd_idle and MCFG1.sr_idle.

Following table outlines the effect of these settings in conjunction with NIF being idle.

pd_idle	sr_idle	Memory modes	Memory Type
0	0	none	All
>0	0	Power Down	All
0	>0	Self Refresh	All
>0	>0	Power Down -> Self Refresh ³	All

Note:

1. Power Down is entered if NIF is idle for pd_idle. Following on from that, if NIF continues to be idle for a further sr_idle*32 cycles, Power Down is exited and Self Refresh is entered.
2. Following on from that, if NIF continues to be idle for a further sr_idle*32 cycles, Power Down is exited and Self Refresh is entered.

Removing PCTL's n_clk

In DDR3 and LPDDR2, the relationship between SRE/SRX and stopping/starting the memory clock (CK) are formalized and are accounted for automatically by

PCTL. With DDR3 and LPDDR2, CK should only be stopped after PCTL has reached the Low_power state. The current operational state can be verified by reading STAT.ctl_stat. The CK must be started and stable before the Software or Hardware Low Power Interface attempts to take the memory out of Self Refresh.

PCTL's n_clk can be safely removed when PCTL is in Low Power state. The sequences outlined in following two tables should be followed for safe operation:

Step	Application	PCTL
1	Write SLEEP to SCTL.state_cmd and poll STAT.ctl_stat = LOW_POWER.	Tells PCTL to move memories into Self Refresh and waits until this completes.
2	Write TREFI=0. Also, write DFITCRLUPDI=0 and DFIREFMSKI=0, if they are not already 0.	Stops any MC-driven DFI updates occurring internally with PCTL
3	Wait a minimum interval which is equivalent to the PCTL's Refresh Interval (previous value of TREFI*TOGCNT100N*internal timers clock period;	Ensures any already scheduled PHY/PVT updates have completed successfully.
4	Stop toggling n_clk to PCTL.	n_clk logic inside PCTL is stopped.
end		

Step	Application	PCTL
1	Drive c_active_in low	Confirms that system external to PCTL can accept a Low-power request
2	Drive c_sysreq low	System Low-power request
3	Wait for PCTL to drive c_sysack low	PCTL Low-power request acknowledgement
4	Check value of c_active when Step 3 occurs. - if c_active=1, request denied. Cannot remove n_clk. Go to END. - if c_active=0, request accepted.	PCTL low-power request status response
5	Stop toggling n_clk to PCTL	n_clk logic inside PCTL is stopped
end		

10.6.4 TX DLLs

All high speed IO signals' phase can be adjusted by TX DLLs. Table 10-3 illustrates these DLLs.

Table 10-2 DDR PHY TX DLLs Delay Step

Offset	Bit	Control Signal Phase	Default	Description
0x4c	2~0	CMD	0x4	CMD DLL delay step
0x50	2~0	CK	0x0	CK DLL delay step
0x98	2~0	A_DM0, A_DQ7~A_DQ0	0x4	DM and DQ DLL Signal delay step
0xd8	2~0	A_DM1, A_DQ15~A_DQ8	0x4	
0x118	2~0	B_DM0, B_DQ7~B_DQ0	0x4	
0x158	2~0	B_DM1, B_DQ15~B_DQ8	0x4	

0x9c	2~0	A_DQS0, A_DQSB0	0x0	TX DQS DLL Signal delay step
0xdc	2~0	A_DQS1, A_DQSB1	0x0	
0x11c	2~0	B_DQS0, B_DQSB0	0x0	
0x15c	2~0	B_DQS1, B_DQSB1	0x0	

Step 0x0 values means no phase delay, and 0x4 increases delay phase to 90 deg, 0x7 values corresponds to maximum phase delay. All DLLs having 8 delay steps which can get 90 deg phase delay by setting 0x4.

10.6.5 RX DLLs

The RX DLLs are used for sample RX DQS signals with proper phase delay and pulse edges. The DQS squelch (RXMEM) signal opens a window for passing RX DQS pulses, both RX DQS and DQS squelch signal phase can be adjusted by separate DLLs.

Table 10-3 DDR PHY RX DQS Delay Step

Offset	Bit	Control Signal Phase	Default	Description
0xe0	5~3	Left DQS squelch	0x0	Left DQS squelch delay step
	1~0	DQS0, DQSB0	0x0	
0x120	5~3	Right DQS squelch	0x0	Right DQS squelch delay step
	1~0	DQS1, DQSB1	0x0	
0xa0	1~0	A_DQS0, A_DQSB0	0x1	read DQS delay phase
0xe0	1~0	A_DQS1, A_DQSB1	0x1	
0x120	1~0	B_DQS0, B_DQSB0	0x1	
0x160	1~0	B_DQS1, B_DQSB1	0x1	
0xb0	6~4	left channel A RXMEM	0x1	
0xf0	6~4	right channel A RXMEM	0x1	RXMEM delay, unit x1 clock cycle
0x130	6~4	left channel B RXMEM	0x1	
0x170	6~4	right channel B RXMEM	0x1	
0xb0	3	left channel A RXMEM	0x1	
0xf0	3	right channel A RXMEM	0x1	additive and accumulative RXMEM delay, unit 2x clock cycle
0x130	3	left channel B RXMEM	0x1	
0x170	3	right channel B RXMEM	0x1	
0xb0	2~0	left channel A RXMEM	0x4	
0xf0	2~0	right channel A RXMEM	0x4	additive and accumulative RXMEM delay, unit per DLL step
0x130	2~0	left channel B RXMEM	0x4	
0x170	2~0	right channel B RXMEM	0x4	

10.6.6 High Speed IO Drive Strength

The tuning range of driver resistance is 28ohm to 138ohm. By default, 0x5 is 46ohm for DDR3 CMD driver. When the control bit is set to be larger, the drive strength becomes stronger.

Table 10-4 CK, CMD Signal Drive Strength Register

Offset	Bit	Default	Description
0x44	7~4	0xa	adjustable CMD pull-up resistance
	3~0	0xa	adjustable CMD pull-down resistance

0x58	7~4	0xa	adjustable CK pull-up resistance
	3~0	0xa	adjustable CK pull-down resistance

Table 10-5 DM, DQ Signal Drive Strength Register

Offset	Bit	Default	Description
0x80	7~4	0xa	pull-up driving resistance for A_DQ0~A_DQ7
	3~0	0xa	pull-down driving resistance for A_DQ0~A_DQ7
0xc0	7~4	0xa	pull-up driving resistance for A_DQ8~A_DQ15
	3~0	0xa	pull-down driving resistance for A_DQ8~A_DQ15
0x100	7~4	0xa	pull-up driving resistance for B_DQ0~B_DQ7
	3~0	0xa	pull-down driving resistance for B_DQ0~B_DQ7
0x140	7~4	0xa	pull-up driving resistance for B_DQ8~B_DQ15
	3~0	0xa	pull-down driving resistance for B_DQ8~B_DQ15

The value is larger, the drive strength is stronger.

Table 10-6 DM/DQ/DQS/CMD Driver output resistance with control bit

Control bit	4'b0000	4'b0001	4'b0010	4'b0011	4'b0100	4'b0101	4'b0110	4'b0111
Pull-up resistance	+∞	309ohm	155ohm	103ohm	77ohm	63ohm	52ohm	45ohm
Pull-down resistance	+∞	309ohm	155ohm	103ohm	77ohm	63ohm	52ohm	45ohm
Control bit	4'b1000	4'b1001	4'b1010	4'b1011	4'b1100	4'b1101	4'b1110	4'b1111
Pull-up resistance	77ohm	62ohm	52ohm	44ohm	39ohm	34ohm	31ohm	28ohm
Pull-down resistance	77ohm	62ohm	52ohm	44ohm	39ohm	34ohm	31ohm	28ohm

Table 10-7 DM/DQ/DQS RX ODT resistance with control bit

Control bit	4'b0000	4'b0001	4'b0010	4'b0011	4'b0100	4'b0101	4'b0110	4'b0111
Pull-up resistance	+∞	861ohm	431ohm	287ohm	216ohm	172ohm	145ohm	124ohm
Pull-down resistance	+∞	861ohm	431ohm	287ohm	216ohm	172ohm	145ohm	124ohm
Control bit	4'b1000	4'b1001	4'b1010	4'b1011	4'b1100	4'b1101	4'b1110	4'b1111
Pull-up resistance	215ohm	172ohm	144ohm	123ohm	108ohm	96ohm	86ohm	78ohm
Pull-down resistance	215ohm	172ohm	144ohm	123ohm	108ohm	96ohm	86ohm	78ohm

10.6.7 PHY Low Speed Mode (200MHz)

DDR PHY supports low speed to high speed DDR3 and LPDDR2 by using two operating mode: normal delay line mode up to 800Mbps or more, low power mode where we support any speed up to 533Mbps. If all TX DLLs are bypassed, the PHY will enter low power state.

The DDR PHY enters low power mode when setting DLLs into Bypass mode. Table 10-9 illustrates related register settings.

Table 10-8 Low Power DLL Setting

Offset	Bit	Default	Low power Setting	Description
0x290	4	0x0	0x1	right channel B TX DQ DLL in bypass mode
	3	0x0	0x1	left channel B TX DQ DLL in bypass mode
	2	0x0	0x1	right channel A TX DQ DLL in bypass mode
	1	0x0	0x1	left channel A TX DQ DLL in bypass mode
	0	0x0	0x1	CMD/CK DLL in bypass mode

0x4c	4	0x0	0x1	CMD DLL phase select
0x50	3	0x0	0x0	CK DLL phase select
0x98	4	0x0	0x1	A_DQ0~A_DQ7 TX DLL phase select
0x9c	3	0x0	0x0	A_DQS0/A_DQSB0 TX DLL phase select
0xd8	4	0x0	0x1	A_DQ8~A_DQ15 TX DLL phase select
0xdc	3	0x0	0x0	A_DQS1/A_DQSB1 TX DLL phase select
0x118	4	0x1	0x1	B_DQ0~B_DQ7 TX DLL phase select
0x11c	3	0x0	0x0	B_DQS0/B_DQSB0 TX DLL phase select
0x158	4	0x1	0x1	B_DQ8~B_DQ15 TX DLL phase select
0x15c	3	0x0	0x0	B_DQS1/B_DQSB1 TX DLL phase select

Those bits lead to command DLL in bypass mode. Both DQS0 DLL and DQ7~DQ0 DLLs are bypassed, but only DQS0 DLL should be in inversion mode because DQS0 step delay is 0x00 and DQ7~DQ0 DLL's step delay is 0x4.

10.6.8 Per bit de-skew tuning

Per-bit de-skew is designed for compensating PCB trace mismatch, DDR PHY support skew individually adjustable for all PHY signals. There are eight steps for each bit de-skew adjusting, and the adjust resolution under different corners is shown below:

Table 10-9 per-bit de-skew tuning resolution

	ff	tt	ss
de-skew resolution	15ps	20ps	30ps

Pre-bit de-skew is realized with inverter chain delay, per-bit de-skew control signals select how much inverters are connected to data path, the minimum resolution is determined by the two inverters minimum delay.

TX path deskew and RX path deskew employ same delay line, and they have same deskew tuning resolution. Minimum RX deskew tuning resolution can be about 28ps with SMIC40lp tt corner process, and we can re-design tuning resolution according to system and customer requirement.

10.6.9 DDR PHY Calibration

DDR PHY auto-adjustment function has been implemented in the PHY. The entire training processes only need to modify the register to start and complete enough.

The entire training process is as follows:

1. PHY's register is reset, the setup is complete.
2. Controller to send the initial command and to complete initialization.
3. Set the PHY's register beginning calibration.

Offset	Bit	Default	Description
0x8	7~2	0x0	Others register
	1	0x0	set calibration bypass mode(1:bypass mode; 0:nomal)
	0	0x0	set calibration start (1: start; 0: stop)

4. After setting register in 5us complete calibration, see the calibration register can be read at this time is complete.

5. Normal read and writes operation can begin.

The software training process is as follows:

1. PHY's register is reset, the setup is complete.
2. Controller to send the initial command and to complete initialization.
3. Set the PHY's register calibration bypass mode.
4. Send 4 consecutive burst8/4 read command.
5. Read register address 0x3c4 value idqs.
6. According to the state diagram shown below to change the cycle ophse dll value.
7. Repeat 4,5,6 until you have completed.

Chapter 11 Crypto

11.1 Overview

Crypto is a hardware accelerator of encrypting or decrypting. It supports the most commonly used algorithm: DES/3DES, AES, SHA1, SHA256, MD5 and RSA.

The Crypto supports following features:

- ◆ Support AES 128/192/256 bits key mode, ECB/CBC/CTR chain mode, Slave/FIFO mode
- ◆ Support DES/3DES (ECB and CBC chain mode) , 3DES (EDE/ EEE key mode), Slave/FIFO mode
- ◆ Support SHA1/SHA256/MD5 (with hardware padding) HASH function, FIFO mode only
- ◆ Support 160 bit Pseudo Random Number Generator (PRNG)
- ◆ Support PKA 512/1024/2048 bit Exp Modulator
- ◆ Support up to 150M clock frequency

11.2 Block Diagram

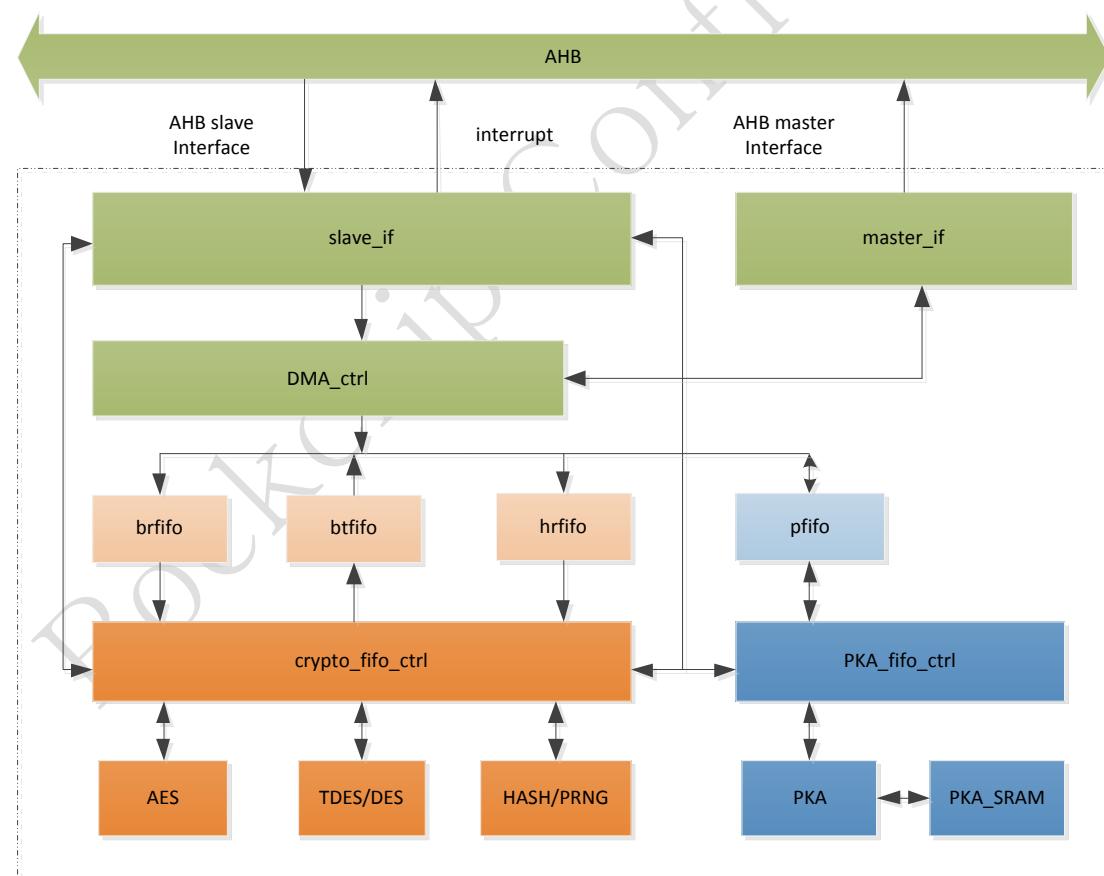


Fig. 11-1 Crypto Architecture

Figure above shows the architecture of Crypto.

11.3 Register description

11.3.1 Register Summary

Name	Offset	Size	Reset Value	Description
CRYPTO_INTSTS	0x0000	W	0x00000000	Interrupt Status Register
CRYPTO_INTENA	0x0004	W	0x00000000	Interrupt Set Register
CRYPTO_CTRL	0x0008	W	0x00000000	Control Register
CRYPTO_CONF	0x000c	W	0x00000000	
CRYPTO_BRDMAS	0x0010	W	0x00000000	Block Receiving DMA Start Address Register
CRYPTO_BTDMAS	0x0014	W	0x00000000	Block Transmiting DMA Start Address Register
CRYPTO_BRDMAL	0x0018	W	0x00000000	Block Receiving DMA Length Register
CRYPTO_HRDMAS	0x001c	W	0x00000000	Hash Receiving DMA Start Address Register
CRYPTO_HRDMAL	0x0020	W	0x00000000	Hash Receiving DMA Length Register
CRYPTO_AES_CTRL	0x0080	W	0x00000000	AES Control Register
CRYPTO_AES_STS	0x0084	W	0x00000000	Status Register
CRYPTO_AES_DIN_0	0x0088	W	0x00000000	AES Input Data 0 Register
CRYPTO_AES_DIN_1	0x008c	W	0x00000000	AES Input Data 1 Register
CRYPTO_AES_DIN_2	0x0090	W	0x00000000	AES Input Data 2 Register
CRYPTO_AES_DIN_3	0x0094	W	0x00000000	AES Input Data 3 Register
CRYPTO_AES_DOUT_0	0x0098	W	0x00000000	AES Output Data 0 Register
CRYPTO_AES_DOUT_1	0x009c	W	0x00000000	AES Output Data 1 Register
CRYPTO_AES_DOUT_2	0x00a0	W	0x00000000	AES Output Data 2 Register
CRYPTO_AES_DOUT_3	0x00a4	W	0x00000000	AES Output Data 3 Register
CRYPTO_AES_IV_0	0x00a8	W	0x00000000	AES IV data 0 Register
CRYPTO_AES_IV_1	0x00ac	W	0x00000000	AES IV data 1 Register
CRYPTO_AES_IV_2	0x00b0	W	0x00000000	AES IV data 2 Register
CRYPTO_AES_IV_3	0x00b4	W	0x00000000	AES IV data 3 Register
CRYPTO_AES_KEY_0	0x00b8	W	0x00000000	AES Key data 0 Register
CRYPTO_AES_KEY_1	0x00bc	W	0x00000000	AES Key data 1 Register
CRYPTO_AES_KEY_2	0x00c0	W	0x00000000	AES Key data 2 Register
CRYPTO_AES_KEY_3	0x00c4	W	0x00000000	AES Key data 3 Register
CRYPTO_AES_KEY_4	0x00c8	W	0x00000000	AES Key data 4 Register
CRYPTO_AES_KEY_5	0x00cc	W	0x00000000	AES Key data 5 Register
CRYPTO_AES_KEY_6	0x00d0	W	0x00000000	AES Key data 6 Register
CRYPTO_AES_KEY_7	0x00d4	W	0x00000000	AES Key data 7 Register
CRYPTO_AES_CNT_0	0x00d8	W	0x00000000	AES Input Counter 0 Register
CRYPTO_AES_CNT_1	0x00dc	W	0x00000000	AES Input Counter 1 Register
CRYPTO_AES_CNT_2	0x00e0	W	0x00000000	AES Input Counter 2 Register
CRYPTO_AES_CNT_3	0x00e4	W	0x00000000	AES Input Counter 3 Register
CRYPTO_TDES_CTRL	0x0100	W	0x00000000	TDES Control Register

Name	Offset	Size	Reset Value	Description
CRYPTO_TDES_STS	0x0104	W	0x00000000	Status Register
CRYPTO_TDES_DIN_0	0x0108	W	0x00000000	TDES Input Data 0 Register
CRYPTO_TDES_DIN_1	0x010c	W	0x00000000	TDES Input Data 1 Register
CRYPTO_TDES_DOU_T_0	0x0110	W	0x00000000	TDES Output Data 0 Register
CRYPTO_TDES_DOU_T_1	0x0114	W	0x00000000	TDES Output Data 1 Register
CRYPTO_TDES_IV_0	0x0118	W	0x00000000	TDES IV data 0 Register
CRYPTO_TDES_IV_1	0x011c	W	0x00000000	TDES IV data 1 Register
CRYPTO_TDES_KEY1_0	0x0120	W	0x00000000	TDES Key1 data 1 Register
CRYPTO_TDES_KEY1_1	0x0124	W	0x00000000	TDES Key1 data 1 Register
CRYPTO_TDES_KEY2_0	0x0128	W	0x00000000	TDES Key2 data 0 Register
CRYPTO_TDES_KEY2_1	0x012c	W	0x00000000	TDES Key2 data 1 Register
CRYPTO_TDES_KEY3_0	0x0130	W	0x00000000	TDES Key3 data 0 Register
CRYPTO_TDES_KEY3_1	0x0134	W	0x00000000	TDES Key3 data 1 Register
CRYPTO_HASH_CTRL	0x0180	W	0x00000000	Hash Control Register
CRYPTO_HASH_STS	0x0184	W	0x00000000	Hash Status Register
CRYPTO_HASH_MSG_LEN	0x0188	W	0x00000000	Hash Message Len
CRYPTO_HASH_DOU_T_0	0x018c	W	0x00000000	Hash Result Register 0
CRYPTO_HASH_DOU_T_1	0x0190	W	0x00000000	Hash Result Register 1
CRYPTO_HASH_DOU_T_2	0x0194	W	0x00000000	Hash Result Register 2
CRYPTO_HASH_DOU_T_3	0x0198	W	0x00000000	Hash Result Register 3
CRYPTO_HASH_DOU_T_4	0x019c	W	0x00000000	Hash Result Register 4
CRYPTO_HASH_DOU_T_5	0x01a0	W	0x00000000	Hash Result Register 4
CRYPTO_HASH_DOU_T_6	0x01a4	W	0x00000000	Hash Result Register 6
CRYPTO_HASH_DOU_T_7	0x01a8	W	0x00000000	Hash Result Register 7
CRYPTO_HASH_SEE_D_0	0x01ac	W	0x00000000	PRNG Seed/HMAC Key Register 0
CRYPTO_HASH_SEE_D_1	0x01b0	W	0x00000000	PRNG Seed/HMAC Key Register 1
CRYPTO_HASH_SEE_D_2	0x01b4	W	0x00000000	PRNG Seed/HMAC Key Register 2

Name	Offset	Size	Reset Value	Description
CRYPTO_HASH_SEE_D_3	0x01b8	W	0x00000000	PRNG Seed/HMAC Key Register 3
CRYPTO_HASH_SEE_D_4	0x01bc	W	0x00000000	PRNG Seed/HMAC Key Register 4
CRYPTO_TRNG_CTRL	0x0200	W	0x00000000	TRNG Control
CRYPTO_TRNG_DOU_T_0	0x0204	W	0x00000000	TRNG Output Data 0
CRYPTO_TRNG_DOU_T_1	0x0208	W	0x00000000	TRNG Output Data 1
CRYPTO_TRNG_DOU_T_2	0x020c	W	0x00000000	TRNG Output Data 2
CRYPTO_TRNG_DOU_T_3	0x0210	W	0x00000000	TRNG Output Data 3
CRYPTO_TRNG_DOU_T_4	0x0214	W	0x00000000	TRNG Output Data 4
CRYPTO_TRNG_DOU_T_5	0x0218	W	0x00000000	TRNG Output Data 5
CRYPTO_TRNG_DOU_T_6	0x021c	W	0x00000000	TRNG Output Data 6
CRYPTO_TRNG_DOU_T_7	0x0220	W	0x00000000	TRNG Output Data 7
CRYPTO_PKA_CTRL	0x0280	W	0x00000000	PKA Control Register
CRYPTO_PKA_M	0x0400	W	0x00000000	
CRYPTO_PKA_C	0x0500	W	0x00000000	
CRYPTO_PKA_N	0x0600	W	0x00000000	
CRYPTO_PKA_E	0x0700	W	0x00000000	

Notes: Size : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** -WORD (32 bits) access

11.3.2 Detail Register Description

CRYPTO_INTSTS

Address: Operational Base + offset (0x0000)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	PKA_DONE_INT PKA Done Interrupt
4	W1C	0x0	HASH_DONE_INT Hash Done Interrupt
3	W1C	0x0	HRDMA_ERR_INT Specifies the interrupt of hash receiving DMA Error
2	W1C	0x0	HRDMA_DONE_INT Specifies the interrupt of hash receiving DMA DONE

Bit	Attr	Reset Value	Description
1	W1C	0x0	BCDMA_ERR_INT Specifies the interrupt of block cipher Error
0	W1C	0x0	BCDMA_DONE_INT Specifies the interrupt of block cipher DONE

CRYPTO_INTENA

Address: Operational Base + offset (0x0004)

Interrupt Set Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	PKA_DONE_ENA Set the interrupt Enable of PKA done 1'b1: enable 1'b0: disable
4	RW	0x0	HASH_DONE_ENA Set the interrupt Enable of hash done 1'b1: enable 1'b0: disable
3	RW	0x0	HRDMA_ERR_ENA Set the interrupt Enable of hash receiving DMA Error 1'b1: enable 1'b0: disable
2	RW	0x0	HRDMA_DONE_ENA Set the interrupt Enable of hash receiving DMA DONE 1'b1: enable 1'b0: disable
1	RW	0x0	BCDMA_ERR_ENA Set the interrupt Enable of block cipher DMA Error 1'b1: enable 1'b0: disable
0	RW	0x0	BCDMA_DONE_ENA Set the interrupt Enable of block cipher DMA DONE 1'b1: enable 1'b0: disable

CRYPTO_CTRL

Address: Operational Base + offset (0x0008)

Control Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	Write_Mask
15:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	TRNG_FLUSH FLUSH TRNG Software write 1 to start. When finishes, the core will clear it.
8	RWSC	0x0	TRNG_START Start TRNG Software write 1 to start. When finishes, the core will clear it.
7	RWSC	0x0	PKA_FLUSH Software write 1 to start Flush Process. The process will clear BRFIFO, BTFIFO, and state machine. Then Software should write 0 to end FLUSH Process
6	RW	0x0	HASH_FLUSH Software write 1 to start Flush Process. The process will clear BRFIFO, BTFIFO, and state machine. Then Software should write 0 to end FLUSH Process
5	RW	0x0	BLOCK_FLUSH Software write 1 to start Flush Process. The process will clear BRFIFO, BTFIFO, and state machine. Then Software should write 0 to end FLUSH Process. It must last for at least 20 cycles to clean registers and FSM
4	RWSC	0x0	PKA_START Starts/initializes PKA Software write 1 to start. When finishes, the core will clear it.
3	RWSC	0x0	HASH_START Starts/initializes HASH/PRNG/HMAC Software write 1 to start. When finishes, the core will clear it.
2	RWSC	0x0	BLOCK_START Starts/initializes Block Cipher Software write 1 to start. When finishes, the core will clear it.
1	RWSC	0x0	TDES_START Starts/initializes TDES Software write 1 to start. When finishes, the core will clear it.
0	RWSC	0x0	AES_START Starts/initializes AES Software write 1 to start. When finishes, the core will clear it. Software can also write 0 to clear it.

CRYPTO_CONF

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	HR_ADDR_MODE Hash Receive DMA Address Mode 1'b1: fix 1'b0: increment
7	RW	0x0	BT_ADDR_MODE Block Transmit DMA Address Mode 1'b1: fix 1'b0: increment
6	RW	0x0	BR_ADDR_MODE Block Receive DMA Address Mode 1'b1: fix 1'b0: increment
5	RW	0x0	Byteswap_HRFIFO If this bit is high, then the data read from the bus is byte-swapped in a word boundary. If this bit is low (default), then the data is handed over to the FIFO without byte-swap. For little endian bus, this bit should be 1'b1.
4	RW	0x0	Byteswap_BTFIFO If this bit is high, then the data read from the bus is byte-swapped in a word boundary. If this bit is low (default), then the data is handed over to the FIFO without byte-swap. For little endian bus, this bit should be 1'b1.
3	RW	0x0	Byteswap_BRFIFO If this bit is high, then the data read from the bus is byte-swapped in a word boundary. If this bit is low (default), then the data is handed over to the FIFO without byte-swap. For little endian bus, this bit should be 1'b1.
2	RW	0x0	DESSEL Specifies the Destination block cipher of FIFO. AES(=0)/DES(=1)
1:0	RW	0x0	HASHINSEL Specifies the following Data from independent source (0) Data from block cipher input (1) Data from block cipher output (2) Reserved (3)

CRYPTO_BRDMAS

Address: Operational Base + offset (0x0010)

Block Receiving DMA Start Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STARTADDR Specifies the Start Address of DMA The address should be aligned by 32-bit.

CRYPTO_BTDMAS

Address: Operational Base + offset (0x0014)

Block Transmitting DMA Start Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STARTADDR Specifies the Start Address of DMA The address needs to be aligned by 32-bit.

CRYPTO_BRDMAL

Address: Operational Base + offset (0x0018)

Block Receiving DMA Length Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	LENGTH Specifies the Block length of DMA. The length unit is WORD.

CRYPTO_HRDMAS

Address: Operational Base + offset (0x001c)

Hash Receiving DMA Start Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STARTADDR Specifies the Start Address of DMA The address needs to be aligned by 32-bit.

CRYPTO_HRDMAL

Address: Operational Base + offset (0x0020)

Hash Receiving DMA Length Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	LENGTH Specifies the Block length of DMA. The length unit is BYTE.

CRYPTO_AES_CTRL

Address: Operational Base + offset (0x0080)

AES Control Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11	RW	0x0	AES_BitSwap_CNT Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Counter data byte swap 1 = Enables Counter data byte swap
10	RW	0x0	AES_BitSwap_Key Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Key byte swap 1 = Enables Key byte swap

Bit	Attr	Reset Value	Description
9	RW	0x0	AES_BitSwap_IV Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Initial value byte swap 1 = Enables Initial value byte swap
8	RW	0x0	AES_BitSwap_DO Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Output data byte swap 1 = Enables Output data byte swap
7	RW	0x0	AES_BitSwap_DI Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Input data byte swap 1 = Enables Input data byte swap
6	RW	0x0	AES_KeyChange Specifies the AES key change mode selection signal. When the bit is asserted, it will not do key-expansion function to calculate new sub-key. So it is a faster way, when several times of calculation use the same key. But if the keys are different, asserting this bit will have the wrong result. 0 = Key is not changed 1 = Key is changed
5:4	RW	0x0	AES_ChainMode Specifies AES chain mode selection 00 = ECB mode 01 = CBC mode 10 = CTR mode
3:2	RW	0x0	AES_KeySize Specifies the AES key size selection signal 00 : 128-bit key 01 : 192-bit key 10 : 256-bit key
1	RW	0x0	AES_FifoMode Specify AES Fifo Mode 1'b0: Slave mode 1'b1: fifo mode
0	RW	0x0	AES_Enc Specifies the Encryption/ Decryption mode selection signal 0 : Encryption 1 : Decryption

CRYPTO_AES_STS

Address: Operational Base + offset (0x0084)

Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	AES_DONE When AES finish, it will be HIGH, And it will not be LOW until it restart . 1: done 0: not done

CRYPTO_AES_DIN_0

Address: Operational Base + offset (0x0088)

AES Input Data 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_DIN_0 Specifies AES Input data [127:96].

CRYPTO_AES_DIN_1

Address: Operational Base + offset (0x008c)

AES Input Data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_DIN_1 Specifies AES Input data [95:64].

CRYPTO_AES_DIN_2

Address: Operational Base + offset (0x0090)

AES Input Data 2 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_DIN_2 Specifies AES Input data [63:32]

CRYPTO_AES_DIN_3

Address: Operational Base + offset (0x0094)

AES Input Data 3 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_DIN_3 Specifies AES Input data [31:0]

CRYPTO_AES_DOUT_0

Address: Operational Base + offset (0x0098)

AES Output Data 0 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	AES_DOUT_0 Specifies AES Output data [127:96].

CRYPTO_AES_DOUT_1

Address: Operational Base + offset (0x009c)

AES Output Data 1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	AES_DOUT_1 Specifies the Output data [95:64].

CRYPTO_AES_DOUT_2

Address: Operational Base + offset (0x00a0)

AES Output Data 2 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	AES_DOUT_2 Specifies AES Output data [63:32].

CRYPTO_AES_DOUT_3

Address: Operational Base + offset (0x00a4)

AES Output Data 3 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	AES_DOUT_3 Specifies AES Output data [31:0].

CRYPTO_AES_IV_0

Address: Operational Base + offset (0x00a8)

AES IV data 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_IV_0 Specifies AES Initialization vector [127:96]

CRYPTO_AES_IV_1

Address: Operational Base + offset (0x00ac)

AES IV data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_IV_1 Specifies AES Initialization vector [95:64]

CRYPTO_AES_IV_2

Address: Operational Base + offset (0x00b0)

AES IV data 2 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_IV_2 Specifies AES Initialization vector [63:32]

CRYPTO_AES_IV_3

Address: Operational Base + offset (0x00b4)
 AES IV data 3 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_IV_3 Specifies AES Initialization vector [31:0]

CRYPTO_AES_KEY_0

Address: Operational Base + offset (0x00b8)
 AES Key data 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY_0 Specifies AES key data [255:224]

CRYPTO_AES_KEY_1

Address: Operational Base + offset (0x00bc)
 AES Key data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY_1 Specifies AES key data [223:192]

CRYPTO_AES_KEY_2

Address: Operational Base + offset (0x00c0)
 AES Key data 2 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY_2 Specifies AES key data [191:160]

CRYPTO_AES_KEY_3

Address: Operational Base + offset (0x00c4)
 AES Key data 3 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY_3 Specifies AES key data [159:128]

CRYPTO_AES_KEY_4

Address: Operational Base + offset (0x00c8)
 AES Key data 4 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY_4 Specifies AES key data [127:96]

CRYPTO_AES_KEY_5

Address: Operational Base + offset (0x00cc)

AES Key data 5 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY_5 Specifies the key data [95:64]

CRYPTO_AES_KEY_6

Address: Operational Base + offset (0x00d0)

AES Key data 6 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY_6 Specifies AES key data [63:32]

CRYPTO_AES_KEY_7

Address: Operational Base + offset (0x00d4)

AES Key data 7 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY_7 Specifies the key data [31:0]

CRYPTO_AES_CNT_0

Address: Operational Base + offset (0x00d8)

AES Input Counter 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_CNT_0 Specifies AES Input Counter [127:96].

CRYPTO_AES_CNT_1

Address: Operational Base + offset (0x00dc)

AES Input Counter 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_CNT_1 Specifies AES Input Counter [95:64].

CRYPTO_AES_CNT_2

Address: Operational Base + offset (0x00e0)

AES Input Counter 2 Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_CNT_2 Specifies AES Input Counter[63:32]

CRYPTO_AES_CNT_3

Address: Operational Base + offset (0x00e4)

AES Input Counter 3 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_CNT_3 Specifies AES Input Counter [31:0]

CRYPTO_TDES_CTRL

Address: Operational Base + offset (0x0100)

TDES Control Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	TDES_BitSwap_Key 0 = Disables Key byte swap 1 = Enables Key byte swap
7	RW	0x0	TDES_BitSwap_IV 0 = Disables Initial value byte swap 1 = Enables Initial value byte swap
6	RW	0x0	TDES_BitSwap_DO 0 = Disables Output data byte swap 1 = Enables Output data byte swap
5	RW	0x0	TDES_BitSwap_DI 0 = Disables Input data byte swap 1 = Enables Input data byte swap
4	RW	0x0	TDES_ChainMode Specifies TDES chain mode selection 0 : ECB mode 1 : CBC mode
3	RW	0x0	TDES_EEE Specifies the TDES key mode selection 1'b0 : EDE 1'b1 : EEE
2	RW	0x0	TDES_Select Specify DES or TDES cipher 1'b0 : DES 1'b1 : TDES
1	RW	0x0	TDES_FifoMode Specify TDES Fifo Mode 1'b0: Slave mode 1'b1: Fifo mode
0	RW	0x0	TDES_Enc Specifies the Encryption/ Decryption mode selection signal 0 : Encryption 1 : Decryption

CRYPTO_TDES_STS

Address: Operational Base + offset (0x0104)

Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	TDES_DONE When DES/TDES finishes, it will be HIGH, And it will not be LOW until it restart . 1: done 0: not done

CRYPTO_TDES_DIN_0

Address: Operational Base + offset (0x0108)

TDES Input Data 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_DIN_0 Specifies TDES Input data [63:32].

CRYPTO_TDES_DIN_1

Address: Operational Base + offset (0x010c)

TDES Input Data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_DIN_1 Specifies TDES Input data [31:0].

CRYPTO_TDES_DOUT_0

Address: Operational Base + offset (0x0110)

TDES Output Data 0 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TDES_DOUT_0 Specifies TDES Output data [63:32].

CRYPTO_TDES_DOUT_1

Address: Operational Base + offset (0x0114)

TDES Output Data 1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TDES_DOUT_1 Specifies TDES Output data [31:0].

CRYPTO_TDES_IV_0

Address: Operational Base + offset (0x0118)

TDES IV data 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_IV_0 Specifies TDES Initialization vector [63:32]

CRYPTO_TDES_IV_1

Address: Operational Base + offset (0x011c)

TDES IV data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_IV_1 Specifies TDES Initialization vector [31:0]

CRYPTO_TDES_KEY1_0

Address: Operational Base + offset (0x0120)

TDES Key1 data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_KEY1_0 Specifies TDES key1 data [63:32]

CRYPTO_TDES_KEY1_1

Address: Operational Base + offset (0x0124)

TDES Key1 data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_KEY1_1 Specifies TDES key1 data [31:0]

CRYPTO_TDES_KEY2_0

Address: Operational Base + offset (0x0128)

TDES Key2 data 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_KEY2_0 Specifies TDES key2 data [63:32]

CRYPTO_TDES_KEY2_1

Address: Operational Base + offset (0x012c)

TDES Key2 data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_KEY_1 Specifies TDES key data [31:0]

CRYPTO_TDES_KEY3_0

Address: Operational Base + offset (0x0130)

TDES Key3 data 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDES_KEY3_0 Specifies TDES key3 data [63:32]

CRYPTO_TDES_KEY3_1

Address: Operational Base + offset (0x0134)

TDES Key3 data 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AES_KEY3_1 Specifies TDES key3 data [31:0]

CRYPTO_HASH_CTRL

Address: Operational Base + offset (0x0180)

Hash Control Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	HASH_SWAP_DO Specifies the Byte swap of data output (hash result) 0 = Does not swap (default) 1 = Swap
2	RW	0x0	HASH_SWAP_DI Specifies the Byte swap of data input. 0 = Does not swap (default) 1 = Swap
1:0	RW	0x0	Engine_Selection 2'b00: SHA1_HASH 2'b01: MD5_HASH 2'b10: SHA256_HASH 2'b11: PRNG

CRYPTO_HASH_STS

Address: Operational Base + offset (0x0184)

Hash Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	HASH_DONE Hash Done Signal When HASH finishes, it will be HIGH, And it will not be LOW until it restart 1'b1 : done 1'b0 : not done

CRYPTO_HASH_MSG_LEN

Address: Operational Base + offset (0x0188)

Hash Message Len

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Msg_size Hash total byte.

CRYPTO_HASH_DOUT_0

Address: Operational Base + offset (0x018c)

Hash Result Register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HASH_RESULT_0 Specifies the HASH Result [159:128]

CRYPTO_HASH_DOUT_1

Address: Operational Base + offset (0x0190)

Hash Result Register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HASH_RESULT_1 Specifies the HASH Result [127:96]

CRYPTO_HASH_DOUT_2

Address: Operational Base + offset (0x0194)

Hash Result Register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HASH_RESULT_2 Specifies the HASH Result [95:64]

CRYPTO_HASH_DOUT_3

Address: Operational Base + offset (0x0198)

Hash Result Register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HASH_RESULT_3 Specifies the HASH Result [63:32]

CRYPTO_HASH_DOUT_4

Address: Operational Base + offset (0x019c)

Hash Result Register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HASH_RESULT_4 Specifies the HASH Result [31:0]

CRYPTO_HASH_DOUT_5

Address: Operational Base + offset (0x01a0)

Hash Result Register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HASH_RESULT_5 Specifies the HASH Result [31:0]

CRYPTO_HASH_DOUT_6

Address: Operational Base + offset (0x01a4)

Hash Result Register 6

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HASH_RESULT_6 Specifies the HASH Result [31:0]

CRYPTO_HASH_DOUT_7

Address: Operational Base + offset (0x01a8)

Hash Result Register 7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HASH_RESULT_7 Specifies the HASH Result [31:0]

CRYPTO_HASH_SEED_0

Address: Operational Base + offset (0x01ac)

PRNG Seed/HMAC Key Register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HASH_SEED_0 Specifies PRNG Seed/HMAC Key buffer [159:128]

CRYPTO_HASH_SEED_1

Address: Operational Base + offset (0x01b0)

PRNG Seed/HMAC Key Register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HASH_SEED_1 Specifies PRNG Seed/HMAC Key buffer [127:96]

CRYPTO_HASH_SEED_2

Address: Operational Base + offset (0x01b4)

PRNG Seed/HMAC Key Register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HASH_SEED_2 Specifies PRNG Seed/HMAC Key buffer [95:64]

CRYPTO_HASH_SEED_3

Address: Operational Base + offset (0x01b8)

PRNG Seed/HMAC Key Register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HASH_SEED_3 Specifies PRNG Seed/HMAC Key buffer [63:32]

CRYPTO_HASH_SEED_4

Address: Operational Base + offset (0x01bc)

PRNG Seed/HMAC Key Register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HASH_SEED_4 Specifies PRNG Seed/HMAC Key buffer [31:0]

CRYPTO_TRNG_CTRL

Address: Operational Base + offset (0x0200)

TRNG Control

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	osc_enable osc_ring enable It controll the running of osc_ring. And it is indepent of clock and flush signal. This mean that it can run even when clock is gating or flush is asserted as long as osc_enable is asserted. Before it is used to get TRNG result , please run osc_ring first to get enough entropy. 1'b1: Enable ; 1'b0: Disable ;
15:0	RW	0x0000	period sample period TRNG use clock_crypto to sample ring osc output, this parameter is specify how many cycles to generate 1 bit random data.

CRYPTO_TRNG_DOUT_0

Address: Operational Base + offset (0x0204)

TRNG Output Data 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TRNG_DOUT_0

CRYPTO_TRNG_DOUT_1

Address: Operational Base + offset (0x0208)

TRNG Output Data 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TRNG_DOUT_1

CRYPTO_TRNG_DOUT_2

Address: Operational Base + offset (0x020c)

TRNG Output Data 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TRNG_DOUT_2

CRYPTO_TRNG_DOUT_3

Address: Operational Base + offset (0x0210)

TRNG Output Data 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TRNG_DOUT_3

CRYPTO_TRNG_DOUT_4

Address: Operational Base + offset (0x0214)

TRNG Output Data 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TRNG_DOUT_4

CRYPTO_TRNG_DOUT_5

Address: Operational Base + offset (0x0218)

TRNG Output Data 5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TRNG_DOUT_5

CRYPTO_TRNG_DOUT_6

Address: Operational Base + offset (0x021c)

TRNG Output Data 6

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TRNG_DOUT_6

CRYPTO_TRNG_DOUT_7

Address: Operational Base + offset (0x0220)

TRNG Output Data 7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TRNG_DOUT_7

CRYPTO_PKA_CTRL

Address: Operational Base + offset (0x0280)
 PKA Control Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	block_size PKA Size It specifies the bits of N in PKA calculation. 2'b00: 512 bit 2'b01: 1024 bit 2'b10: 2048 bit

CRYPTO_PKA_M

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	m PKA input or output data. PKA result = (M ^ E) mod N. When it finishes, the result data is in M position. Start from PKA_M base address, and may contain 512/1024/2048 bits data.

CRYPTO_PKA_C

Address: Operational Base + offset (0x0500)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	c PKA pre-calculate data, C = 2 ^ (2n+2) mod N

CRYPTO_PKA_N

Address: Operational Base + offset (0x0600)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	n PKA modular

CRYPTO_PKA_E

Address: Operational Base + offset (0x0700)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	e PKA exponent.

11.4 Application Note

11.4.1 Reset a port

CRU_SOFTRST3_CON. crypto_srstn_req is used to do a soft reset to crypto . Please refer to “Chapter CRU” for more details.

11.4.2 Overall Performance

Use CLKSEL24_CON.crypto_div_con to select crypto frequency: $F_{\text{crypto}} = F_{\text{aclk}} / (\text{div} + 1)$.

Make sure F_{crypto} do not exceed 150M.

The performance of crypto FIFO mode is list below.

algorithm	cycle	block size	frequency	throughput rate
DES	17	64 bit	100M	376 M bps
TDES	51	64 bit	100M	125 M bps
AES	11/13/15	128 bit	100M	1160/984/853Mbps
SHA-1	81	512 bit	100M	632 Mbps
MD5	65	512 bit	100M	787 Mbps

11.4.3 Usage

1. Symmetric algorithm

DES/3DES, AES are symmetric algorithms. There are two ways of using these algorithms: Slave mode and FIFO mode.

In Slave mode, you can calculate 1 block size of data by starting the engine. Take AES-128 for example, you should

- Program Input 128 bit Data to AES_DIN_0~AES_DIN_3
- Program Input 128 bit Key to AES_KEY_0~AES_KEY_3
- Program control mode to AES_CTRL to run in different mode
- Program CTRL.AES_START to run
- wait AES_STS.DONE High
- Read AES_DOUT_0 ~ AES_DOUT_3 to get result.

In FIFO mode,

- Program the source address to BRDMAS, the destination address to BTDMAS, program the length in word unit to BRDMAL;
- Program Input 128 bit Key to AES_KEY_0~AES_KEY_3;
- Program control mode to AES_CTRL to run in different mode;
- Program INTENA to enable interrupt;
- Program CTRL.BLOCK_START to start;
- wait interrupt asserted;
- Program INTSTS to clear interrupt status;
- Read the destination address which BTDMA points to.

FIFO mode get much higher throughput rate.

2. HASH

HASH is used to get digest of data. Only support FIFO mode.

There are three source: (1) hr_fifo; (2) br_fifo; (3) bt_fifo.

Take hr_fifo for example

- (1) Program CTRL.HASH_FLUSH 1'b1 to clear, wait several cycle (≥ 10 cycles), and Program CTRL.HASH_FLUSH 1'b0
- (2) Program data source address to HRDMAS, program 1 time data length in word unit to HRDMAL, program total length in byte unit to HASH_MSG_LEN
- (3) Program HASH_CTRL to choose algorithm, for example SHA-256
- (4) Program INTENA to enable interrupt;
- (5) Program CTRL.HASH_START 1'b1 to start;
- (6) Wait interrupt asserted; Only if HRDMAL length meets can this interrupt be asserted
- (7) If you have another section of data to hash, then go to (2), HASH_MSG_LEN need not to be programmed;
else go to (8)
- (8) wait HASH_STS.done asserted. Only if Hash_MSG_LEN meet can this bit status register asserted.
- (9) Read HASH_DOUT_0 – HASH_DOUT_7 to get result.

3. Asymmetric Algorithm

Support 512/1024/2048 bit RSA calculation. It provide the big number calculation. Result = $M^E \bmod N$

Program CTRL.PKA_FLUSH 1'b1 to flush RSA module;

Wait CTRL.PKA_FLUSH to be LOW. It is self-cleared;

Program input_data(M) to PKA_M; Program pre_caculated C to PKA_C; Program Key(N) to PKA_N; Program Key(E) to PKA_E. $C = 2^{(2n+2)} \bmod N$. n is the required bit of N. For example 2048 bit N, n = 2048;

Program PKA_CTRL to select RSA size: 512/1024/2048

Program INTENA to enable interrupt;

Program CTRL.PKA_START to start;

Wait interrupt asserted.

Read PKA_M to get results.

Chapter 12 NandC(Nand Flash Controller)

12.1 Overview

Nand Flash Controller (NandC) is used to control data transmission from host to flash device or from flash device to host. NandC is connected to AHB BUS through an AHB Master and an AHB Slave. The data transmission between host and external memory can be done through AHB Master interface or AHB Slave interface.

NandC supports the following features:

- NandC AHB bus clock (hclk) is asynchronous to NandC working clock (nclk)
- Software Interface Type
 - Support directly mode
 - Support LLP(Linked List Pointer) mode
- Flash Interface Type
 - Support Asynchronous Flash Interface with 8bits datawidth ("Asyn8x" for short)
 - Support Asynchronous Flash Interface with 16bits data width ("Asyn16x" for short)
 - Support ONFI Synchronous Flash Interface ("ONFI Syn" for short)
 - Support Toggle Flash Interface ("Toggle" for short)
 - Support 8 flash devices at most
- Flash Type
 - Support Managed NAND Flash(LBA) and Raw NAND Flash(NO-LBA)
 - Support SLC/MLC/TLC Flash
- Flash Interface Timing
 - Asyn8x: configurable timing, one byte per two Nandc working clocks at the fastest speed
 - Asyn16x: configurable timing, two bytes per two Nandc working clocks at the fastest speed
 - ONFI Syn: configurable timing, two bytes per two Nandc working clocks at the fastest speed
 - Toggle: configurable timing, two byte per two Nandc working clocks at the fastest speed
- Randomizer Ability
 - Support three randomizer mode with different polynomial
 - Support two randomizer width, 8bit and 16bit parallel
- BCH/ECC Ability
 - 16bit/1KB BCH/ECC: support 16bitBCH/ECC, which can detect and correct up to 16 error bits in every 1K bytes data
 - 24bit/1KB BCH/ECC: support 24bitBCH/ECC, which can detect and correct up to 24 error bits in every 1K bytes data
 - 40bit/1KB BCH/ECC: support 40bitBCH/ECC, which can detect and correct up to 40 error bits in every 1K bytes data
 - 60bit/1KB BCH/ECC: support 60bitBCH/ECC, which can detect and correct up to 60 error bits in every 1K bytes data
 - 8bit/512B BCH/ECC: support 8bitBCH/ECC, which can detect and correct up to 8 error bits in every 512 bytes data
 - 12bit/512B BCH/ECC: support 12bitBCH/ECC, which can detect and correct up to 12 error bits in every 512 bytes data
 - 20bit/512B BCH/ECC: support 20bitBCH/ECC, which can detect and correct up to 20 error bits in every 512 bytes data

- 30bit/512B BCH/ECC: support 30bitBCH/ECC, which can detect and correct up to 30 error bits in every 512 bytes data
- 16bit/512B BCH/ECC: support 16bitBCH/ECC, which can detect and correct up to 16 error bits in every 512 bytes data
- 24bit/512B BCH/ECC: support 24bitBCH/ECC, which can detect and correct up to 24 error bits in every 512 bytes data
- 40bit/512B BCH/ECC: support 40bitBCH/ECC, which can detect and correct up to 40 error bits in every 512 bytes data
- 60bit/512B BCH/ECC: support 60bitBCH/ECC, which can detect and correct up to 60 error bits in every 512 bytes data
- Transmission Ability
 - Support 32K bytes data transmission at a time at most
 - Support two transfer working modes: Bypass or DMA
 - Support two transfer codeword size for Managed NAND Flash: 1024 bytes/codeword or 512 bytes/codeword
- Internal Memory
 - 2 built-in srams, and the size is 1k bytes respectively
 - Can be accessed by other masters
 - Can be operated in pingpong mode by other masters

12.2 Block Diagram

NandC comprises with:

- MIF: AHB Master Interface
- SIF : AHB Slave Interface
- SRIF : Sram Interface
- TRANSC : Transfer Controller
- LLPC : LLP Controller
- BCHENC : BCH Encoder
- BCHDEC : BCH Decoder
- RANDMZ : Randomizer
- FIF_GEN : Flash Interface Generation
- DLC : Delay Line Controller

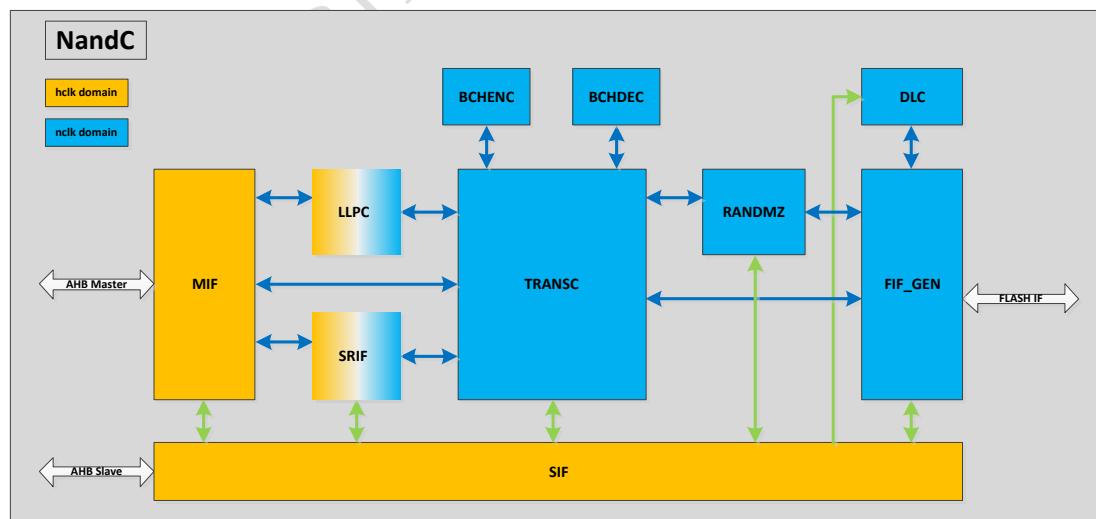


Fig. 12-1 NandC Block Diagram

12.3 Function Description

12.3.1 AHB Interface

There is an AHB master interface in NandC, which is selectable and configurable. It is responsible for transferring data from external memory to internal memory when flash program, or inverse when flash read; and transferring LLP data from external memory to internal register file when LLP is active.

There is an AHB slave interface in NandC. It is responsible for accessing registers and internal memories. The addresses of these registers and memories are listed in "Internal Address Mapping" section.

12.3.2 Flash Type/Flash Interface

Flash device with different types of interfaces is supported. These interfaces include: asynchronous 8bits flash interface, asynchronous 16bits flash interface, ONFI synchronous flash interface, toggle flash interface, and so on. You can select one of them by software (configure FMCTL) to suit for these devices. Also you can configure their timing parameters by software (configure FMWAIT_ASYN/FMWAIT_SYN) to have your desired rate.

12.3.3 Linked List Pointer Mode (LLP)

To save the software resource and improve the performance, a LLP is add, which is selectable. When LLP is selected, the flash operation instructions stored in external memory with specific format should be loaded for flash working. The detailed format and working flow are referred to "LLP Application" section.

12.3.4 BCH Encoder/BCH Decoder

The BCH Encoder is responsible for encoding data to be written into flash device. The max encoded length is 1133bytes, in which the data length is 1024bytes, system information is 4bytes, BCH code is 105bytes.

The BCH Decoder is responsible for decoding data read from flash device. The max decoded length is 1133bytes, in which the data length is 1024bytes, spare length is 109bytes.

12.3.5 Randomizer

To improve device lifetime, a randomizer is added in NandC. It includes two parts: Scrambler and Descrambler, which is responsible for scrambling data to be written into flash after bch encoding, and descrambling data read from flash before bch decoding.

12.3.6 Delay Line Controller

For ONFI Synchronous Flash or Toggle Flash, the data read from flash follows with a strobe signal: DQS, where a skew between them exists. To remove the skew and improve the timing between data and DQS, a Delay Line Controller is needed. It is responsible for detecting the phase of the signal similar to DQS, determining the element number to be shifted, and then shifting the DQS with the determined number.

12.4 Register Description

12.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

Table 12-1 NandC Address Mapping

Base Address[12:8]	Device	Address Length	Offset Address Range
5'b00_00x(x=0, 1)	FLR	512 BYTE	0x0000 ~ 0x01ff
5'b00_01x(x=0, 1)	SPR	512 BYTE	0x0200 ~ 0x03ff
5'b00_10x(x=0, 1)	FLR1	512 BYTE	0x0400 ~ 0x05ff
5'b00_11x(x=0,1)	FLR2	512 BYTE	0x0600 ~ 0x07ff
5'b01_000	Flash0	256 BYTE	0x0800 ~ 0x08ff
5'b01_001	Flash1	256 BYTE	0x0900 ~ 0x09ff
5'b01_010	Flash2	256 BYTE	0xa00 ~ 0xaaff
5'b01_011	Flash3	256 BYTE	0xb00 ~ 0xbff
5'b01_100	Flash4	256 BYTE	0xc00 ~ 0xcff
5'b01_101	Flash5	256 BYTE	0xd00 ~ 0xdff
5'b01_110	Flash6	256 BYTE	0xe00 ~ 0xeff
5'b01_111	Flash7	256 BYTE	0xf00 ~ 0xffff
5'b10_0xx(x=0, 1)	Sram0	1K BYTE	0x1000 ~ 0x13ff
5'b10_1xx(x=0, 1)	Sram1	1K BYTE	0x1400 ~ 0x17ff

12.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
NANDC_FMCTL	0x0000	W	0x000000200	Flash Interface Control Register
NANDC_FMWAIT_AS_YN	0x0004	W	0x3f03f7ff	Flash Timing Control Register For Asynchronous Timing
NANDC_FLCTL	0x0008	W	0x00100000	Internal Transfer Control Register
NANDC_BCHCTL	0x000c	W	0x000000008	BCH Control Register
NANDC_MTRANS_CFG	0x0010	W	0x000001d0	Bus Transfer Configuration Register
NANDC_MTRANS_SA_DDR0	0x0014	W	0x000000000	Start Address Register For Page Data Transmission
NANDC_MTRANS_SA_DDR1	0x0018	W	0x000000000	Start Address Register For Spare Data Transmission
NANDC_MTRANS_STAT	0x001c	W	0x000000000	Bus Transfer Status Register
NANDC_BCHST0	0x0020	W	0x040000000	BCH Status Register For Codeword 0~1
NANDC_BCHST1	0x0024	W	0x000000000	BCH Status Register For Codeword 2~3
NANDC_BCHST2	0x0028	W	0x000000000	BCH Status Register For Codeword 4~5
NANDC_BCHST3	0x002c	W	0x000000000	BCH Status Register For Codeword 6~7

Name	Offset	Size	Reset Value	Description
NANDC_BCHST4	0x0030	W	0x00000000	BCH Status Register For Codeword 8~9
NANDC_BCHST5	0x0034	W	0x00000000	BCH Status Register For Codeword 10~11
NANDC_BCHST6	0x0038	W	0x00000000	BCH Status Register For Codeword 12~13
NANDC_BCHST7	0x003c	W	0x00000000	BCH Status Register For Codeword 14~15
NANDC_BCHLOC0	0x0040	W	0x00000000	BCH Error Bit Location Number Register For Codeword 0~5
NANDC_BCHLOC1	0x0044	W	0x00000000	BCH Error Bit Location Number Register For Codeword 6~11
NANDC_BCHLOC2	0x0048	W	0x00000000	BCH Error Bit Location Number Register For Codeword 12~17
NANDC_BCHLOC3	0x004c	W	0x00000000	BCH Error Bit Location Number Register For Codeword 24~29
NANDC_BCHLOC4	0x0050	W	0x00000000	BCH Error Bit Location Number Register For Codeword 24~29
NANDC_BCHLOC5	0x0054	W	0x00000000	BCH Error Bit Location Number Register For Codeword 30~31
NANDC_BCHLOC6	0x0058	W	0x00000000	Highest Bit For BCH Error Bit Location Number Register
NANDC_BCHDE0_0	0x0070	W	0x00000000	BCH decode result of 0th error bit for codeword 0
NANDC_BCHDE0_1	0x0074	W	0x00000000	BCH decode result of 1th error bit for codeword 0
NANDC_BCHDE0_2	0x0078	W	0x00000000	BCH decode result of 2th error bit for codeword 0
NANDC_BCHDE0_3	0x007c	W	0x00000000	BCH decode result of 3th error bit for codeword 0
NANDC_BCHDE0_4	0x0080	W	0x00000000	BCH decode result of 4th error bit for codeword 0
NANDC_BCHDE0_5	0x0084	W	0x00000000	BCH decode result of 5th error bit for codeword 0
NANDC_BCHDE0_6	0x0088	W	0x00000000	BCH decode result of 6th error bit for codeword 0
NANDC_BCHDE0_7	0x008c	W	0x00000000	BCH decode result of 7th error bit for codeword 0
NANDC_BCHDE0_8	0x0090	W	0x00000000	BCH decode result of 8th error bit for codeword 0
NANDC_BCHDE0_9	0x0094	W	0x00000000	BCH decode result of 9th error bit for codeword 0
NANDC_BCHDE0_10	0x0098	W	0x00000000	BCH decode result of 10th error bit for codeword 0

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE0_11	0x009c	W	0x00000000	BCH decode result of 11th error bit for codeword 0
NANDC_BCHDE0_12	0x00a0	W	0x00000000	BCH decode result of 12th error bit for codeword 0
NANDC_BCHDE0_13	0x00a4	W	0x00000000	BCH decode result of 13th error bit for codeword 0
NANDC_BCHDE0_14	0x00a8	W	0x00000000	BCH decode result of 14th error bit for codeword 0
NANDC_BCHDE0_15	0x00ac	W	0x00000000	BCH decode result of 15th error bit for codeword 0
NANDC_BCHDE0_16	0x00b0	W	0x00000000	BCH decode result of 16th error bit for codeword 0
NANDC_BCHDE0_17	0x00b4	W	0x00000000	BCH decode result of 17th error bit for codeword 0
NANDC_BCHDE0_18	0x00b8	W	0x00000000	BCH decode result of 18th error bit for codeword 0
NANDC_BCHDE0_19	0x00bc	W	0x00000000	BCH decode result of 19th error bit for codeword 0
NANDC_BCHDE0_20	0x00c0	W	0x00000000	BCH decode result of 20th error bit for codeword 0
NANDC_BCHDE0_21	0x00c4	W	0x00000000	BCH decode result of 21th error bit for codeword 0
NANDC_BCHDE0_22	0x00c8	W	0x00000000	BCH decode result of 22th error bit for codeword 0
NANDC_BCHDE0_23	0x00cc	W	0x00000000	BCH decode result of 23th error bit for codeword 0
NANDC_BCHDE1_0	0x00d0	W	0x00000000	BCH decode result of 0th error bit for codeword 1
NANDC_BCHDE1_1	0x00d4	W	0x00000000	BCH decode result of 1th error bit for codeword 1
NANDC_BCHDE1_2	0x00d8	W	0x00000000	BCH decode result of 2th error bit for codeword 1
NANDC_BCHDE1_3	0x00dc	W	0x00000000	BCH decode result of 3th error bit for codeword 1
NANDC_BCHDE1_4	0x00e0	W	0x00000000	BCH decode result of 4th error bit for codeword 1
NANDC_BCHDE1_5	0x00e4	W	0x00000000	BCH decode result of 5th error bit for codeword 1
NANDC_BCHDE1_6	0x00e8	W	0x00000000	BCH decode result of 6th error bit for codeword 1
NANDC_BCHDE1_7	0x00ec	W	0x00000000	BCH decode result of 7th error bit for codeword 1
NANDC_BCHDE1_8	0x00f0	W	0x00000000	BCH decode result of 8th error bit for codeword 1
NANDC_BCHDE1_9	0x00f4	W	0x00000000	BCH decode result of 9th error bit for codeword 1
NANDC_BCHDE1_10	0x00f8	W	0x00000000	BCH decode result of 10th error bit for codeword 1
NANDC_BCHDE1_11	0x00fc	W	0x00000000	BCH decode result of 11th error bit for codeword 1

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE1_12	0x0100	W	0x00000000	BCH decode result of 12th error bit for codeword 1
NANDC_BCHDE1_13	0x0104	W	0x00000000	BCH decode result of 13th error bit for codeword 1
NANDC_BCHDE1_14	0x0108	W	0x00000000	BCH decode result of 14th error bit for codeword 1
NANDC_BCHDE1_15	0x010c	W	0x00000000	BCH decode result of 15th error bit for codeword 1
NANDC_BCHDE1_16	0x0110	W	0x00000000	BCH decode result of 16th error bit for codeword 1
NANDC_BCHDE1_17	0x0114	W	0x00000000	BCH decode result of 17th error bit for codeword 1
NANDC_BCHDE1_18	0x0118	W	0x00000000	BCH decode result of 1th error bit for codeword 1
NANDC_BCHDE1_19	0x011c	W	0x00000000	BCH decode result of 19th error bit for codeword 1
NANDC_BCHDE1_20	0x0120	W	0x00000000	BCH decode result of 20th error bit for codeword 1
NANDC_BCHDE1_21	0x0124	W	0x00000000	BCH decode result of 21th error bit for codeword 1
NANDC_BCHDE1_22	0x0128	W	0x00000000	BCH decode result of 22th error bit for codeword 1
NANDC_BCHDE1_23	0x012c	W	0x00000000	BCH decode result of 23th error bit for codeword 1
NANDC_DLL_CTL_REG0	0x0130	W	0x007f7f05	DLL Control Register 0
NANDC_DLL_CTL_REG1	0x0134	W	0x00000022	DLL Control Register 1
NANDC_DLL_OBS_REG0	0x0138	W	0x00000200	DLL Status Register
NANDC_RANDMZ_CFG	0x0150	W	0x00000000	Randomizer Configure Register
NANDC_FMWAIT_SYN	0x0158	W	0x00000000	Flash Timing Control Register For Synchronous Timing
NANDC_MTRANS_ST_AT2	0x015c	W	0x00000000	Bus Transfer Status Register2
NANDC_NANDC_VER	0x0160	W	0x56363232	Nandc Version Register
NANDC_LLPC_CTL	0x0164	W	0x00000000	LLP Control Register
NANDC_LLPC_STAT	0x0168	W	0x00000001	LLP Status Register
NANDC_INTEN	0x016c	W	0x00000000	NandC Interrupt Enable Register
NANDC_INTCLR	0x0170	W	0x00000000	NandC Interrupt Clear Register
NANDC_INTST	0x0174	W	0x00000000	NandC Interrupt Status Register
NANDC_SPARE0_0	0x0200	W	0xffffffff	System Information for codeword 0
NANDC_SPARE0_1	0x0204	W	0x00000000	Spare Data and BCH Encode Information for codeword 0

Name	Offset	Size	Reset Value	Description
NANDC_SPARE0_2	0x0208	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_3	0x020c	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_4	0x0210	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_5	0x0214	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_6	0x0218	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_7	0x021c	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_8	0x0220	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_9	0x0224	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_10	0x0228	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_11	0x022c	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE1_0	0x0230	W	0xffffffff	System Information for codeword 1
NANDC_SPARE1_1	0x0234	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_2	0x0238	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_3	0x023c	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_4	0x0240	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_5	0x0244	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_6	0x0248	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_7	0x024c	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_8	0x0250	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_9	0x0254	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_10	0x0258	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_11	0x025c	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE0_12	0x0260	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_13	0x0264	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_14	0x0268	W	0x00000000	Spare Data and BCH Encode Information for codeword 0

Name	Offset	Size	Reset Value	Description
NANDC_SPARE0_15	0x026c	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_16	0x0270	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_17	0x0274	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_18	0x0278	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_19	0x027c	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_20	0x0280	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_21	0x0284	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_22	0x0288	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_23	0x028c	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_24	0x0290	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_25	0x0294	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_26	0x0298	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE0_27	0x029c	W	0x00000000	Spare Data and BCH Encode Information for codeword 0
NANDC_SPARE1_12	0x02a0	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_13	0x02a4	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_14	0x02a8	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_15	0x02ac	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_16	0x02b0	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_17	0x02b4	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_18	0x02b8	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_19	0x02bc	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_20	0x02c0	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_21	0x02c4	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_22	0x02c8	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_23	0x02cc	W	0x00000000	Spare Data and BCH Encode Information for codeword 1

Name	Offset	Size	Reset Value	Description
NANDC_SPARE1_24	0x02d0	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_25	0x02d4	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_26	0x02d8	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_SPARE1_27	0x02dc	W	0x00000000	Spare Data and BCH Encode Information for codeword 1
NANDC_BCHDE0_24	0x0400	W	0x00000000	BCH decode result of 24th error bit for codeword 0
NANDC_BCHDE0_25	0x0404	W	0x00000000	BCH decode result of 25th error bit for codeword 0
NANDC_BCHDE0_26	0x0408	W	0x00000000	BCH decode result of 26th error bit for codeword 0
NANDC_BCHDE0_27	0x040c	W	0x00000000	BCH decode result of 27th error bit for codeword 0
NANDC_BCHDE0_28	0x0410	W	0x00000000	BCH decode result of 28th error bit for codeword 0
NANDC_BCHDE0_29	0x0414	W	0x00000000	BCH decode result of 29th error bit for codeword 0
NANDC_BCHDE0_30	0x0418	W	0x00000000	BCH decode result of 30th error bit for codeword 0
NANDC_BCHDE0_31	0x041c	W	0x00000000	BCH decode result of 31th error bit for codeword 0
NANDC_BCHDE0_32	0x0420	W	0x00000000	BCH decode result of 32th error bit for codeword 0
NANDC_BCHDE0_33	0x0424	W	0x00000000	BCH decode result of 33th error bit for codeword 0
NANDC_BCHDE0_34	0x0428	W	0x00000000	BCH decode result of 34th error bit for codeword 0
NANDC_BCHDE0_35	0x042c	W	0x00000000	BCH decode result of 35th error bit for codeword 0
NANDC_BCHDE0_36	0x0430	W	0x00000000	BCH decode result of 36th error bit for codeword 0
NANDC_BCHDE0_37	0x0434	W	0x00000000	BCH decode result of 37th error bit for codeword 0
NANDC_BCHDE0_38	0x0438	W	0x00000000	BCH decode result of 38th error bit for codeword 0
NANDC_BCHDE0_39	0x043c	W	0x00000000	BCH decode result of 39th error bit for codeword 0
NANDC_BCHDE0_40	0x0440	W	0x00000000	BCH decode result of 40th error bit for codeword 0
NANDC_BCHDE0_41	0x0444	W	0x00000000	BCH decode result of 41th error bit for codeword 0
NANDC_BCHDE0_42	0x0448	W	0x00000000	BCH decode result of 42th error bit for codeword 0
NANDC_BCHDE0_43	0x044c	W	0x00000000	BCH decode result of 43th error bit for codeword 0
NANDC_BCHDE0_44	0x0450	W	0x00000000	BCH decode result of 44th error bit for codeword 0

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE0_45	0x0454	W	0x00000000	BCH decode result of 45th error bit for codeword 0
NANDC_BCHDE0_46	0x0458	W	0x00000000	BCH decode result of 46th error bit for codeword 0
NANDC_BCHDE0_47	0x045c	W	0x00000000	BCH decode result of 47th error bit for codeword 0
NANDC_BCHDE0_48	0x0460	W	0x00000000	BCH decode result of 48th error bit for codeword 0
NANDC_BCHDE0_49	0x0464	W	0x00000000	BCH decode result of 49th error bit for codeword 0
NANDC_BCHDE0_50	0x0468	W	0x00000000	BCH decode result of 50th error bit for codeword 0
NANDC_BCHDE0_51	0x046c	W	0x00000000	BCH decode result of 51th error bit for codeword 0
NANDC_BCHDE0_52	0x0470	W	0x00000000	BCH decode result of 52th error bit for codeword 0
NANDC_BCHDE0_53	0x0474	W	0x00000000	BCH decode result of 53th error bit for codeword 0
NANDC_BCHDE0_54	0x0478	W	0x00000000	BCH decode result of 54th error bit for codeword 0
NANDC_BCHDE0_55	0x047c	W	0x00000000	BCH decode result of 55th error bit for codeword 0
NANDC_BCHDE0_56	0x0480	W	0x00000000	BCH decode result of 56th error bit for codeword 0
NANDC_BCHDE0_57	0x0484	W	0x00000000	BCH decode result of 57th error bit for codeword 0
NANDC_BCHDE0_58	0x0488	W	0x00000000	BCH decode result of 58th error bit for codeword 0
NANDC_BCHDE0_59	0x048c	W	0x00000000	BCH decode result of 59th error bit for codeword 0
NANDC_BCHDE1_24	0x0490	W	0x00000000	BCH decode result of 24th error bit for codeword 1
NANDC_BCHDE1_25	0x0494	W	0x00000000	BCH decode result of 25th error bit for codeword 1
NANDC_BCHDE1_26	0x0498	W	0x00000000	BCH decode result of 26th error bit for codeword 1
NANDC_BCHDE1_27	0x049c	W	0x00000000	BCH decode result of 27th error bit for codeword 1
NANDC_BCHDE1_28	0x04a0	W	0x00000000	BCH decode result of 28th error bit for codeword 1
NANDC_BCHDE1_29	0x04a4	W	0x00000000	BCH decode result of 29th error bit for codeword 1
NANDC_BCHDE1_30	0x04a8	W	0x00000000	BCH decode result of 30th error bit for codeword 1
NANDC_BCHDE1_31	0x04ac	W	0x00000000	BCH decode result of 31th error bit for codeword 1
NANDC_BCHDE1_32	0x04b0	W	0x00000000	BCH decode result of 32th error bit for codeword 1
NANDC_BCHDE1_33	0x04b4	W	0x00000000	BCH decode result of 33th error bit for codeword 1

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE1_34	0x04b8	W	0x00000000	BCH decode result of 34th error bit for codeword 1
NANDC_BCHDE1_35	0x04bc	W	0x00000000	BCH decode result of 35th error bit for codeword 1
NANDC_BCHDE1_36	0x04c0	W	0x00000000	BCH decode result of 36th error bit for codeword 1
NANDC_BCHDE1_37	0x04c4	W	0x00000000	BCH decode result of 37th error bit for codeword 1
NANDC_BCHDE1_38	0x04c8	W	0x00000000	BCH decode result of 38th error bit for codeword 1
NANDC_BCHDE1_39	0x04cc	W	0x00000000	BCH decode result of 39th error bit for codeword 1
NANDC_BCHDE1_40	0x04d0	W	0x00000000	BCH decode result of 40th error bit for codeword 1
NANDC_BCHDE1_41	0x04d4	W	0x00000000	BCH decode result of 41th error bit for codeword 1
NANDC_BCHDE1_42	0x04d8	W	0x00000000	BCH decode result of 42th error bit for codeword 1
NANDC_BCHDE1_43	0x04dc	W	0x00000000	BCH decode result of 43th error bit for codeword 1
NANDC_BCHDE1_44	0x04e0	W	0x00000000	BCH decode result of 44th error bit for codeword 1
NANDC_BCHDE1_45	0x04e4	W	0x00000000	BCH decode result of 45th error bit for codeword 1
NANDC_BCHDE1_46	0x04e8	W	0x00000000	BCH decode result of 46th error bit for codeword 1
NANDC_BCHDE1_47	0x04ec	W	0x00000000	BCH decode result of 47th error bit for codeword 1
NANDC_BCHDE1_48	0x04f0	W	0x00000000	BCH decode result of 48th error bit for codeword 1
NANDC_BCHDE1_49	0x04f4	W	0x00000000	BCH decode result of 49th error bit for codeword 1
NANDC_BCHDE1_50	0x04f8	W	0x00000000	BCH decode result of 50th error bit for codeword 1
NANDC_BCHDE1_51	0x04fc	W	0x00000000	BCH decode result of 51th error bit for codeword 1
NANDC_BCHDE1_52	0x0500	W	0x00000000	BCH decode result of 52th error bit for codeword 1
NANDC_BCHDE1_53	0x0504	W	0x00000000	BCH decode result of 53th error bit for codeword 1
NANDC_BCHDE1_54	0x0508	W	0x00000000	BCH decode result of 54th error bit for codeword 1
NANDC_BCHDE1_55	0x050c	W	0x00000000	BCH decode result of 55th error bit for codeword 1
NANDC_BCHDE1_56	0x0510	W	0x00000000	BCH decode result of 56th error bit for codeword 1
NANDC_BCHDE1_57	0x0514	W	0x00000000	BCH decode result of 57th error bit for codeword 1
NANDC_BCHDE1_58	0x0518	W	0x00000000	BCH decode result of 58th error bit for codeword 1

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE1_59	0x051c	W	0x00000000	BCH decode result of 59th error bit for codeword 1
NANDC_BCHST8	0x0520	W	0x00000000	BCH Status Register For Codeword 16~17
NANDC_BCHST9	0x0524	W	0x00000000	BCH Status Register For Codeword 18~19
NANDC_BCHST10	0x0528	W	0x00000000	BCH Status Register For Codeword 20~21
NANDC_BCHST11	0x052c	W	0x00000000	BCH Status Register For Codeword 22~23
NANDC_BCHST12	0x0530	W	0x00000000	BCH Status Register For Codeword 24~25
NANDC_BCHST13	0x0534	W	0x00000000	BCH Status Register For Codeword 26~27
NANDC_BCHST14	0x0538	W	0x00000000	BCH Status Register For Codeword 28~29
NANDC_BCHST15	0x053c	W	0x00000000	BCH Status Register For Codeword 30~31
NANDC_RANDMZ_SE_ED13_0	0x0600	W	0x00000000	Seed 0 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_1	0x0604	W	0x00000000	Seed 1 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_2	0x0608	W	0x00000000	Seed 2 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_3	0x060c	W	0x00000000	Seed 3 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_4	0x0610	W	0x00000000	Seed 4 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_5	0x0614	W	0x00000000	Seed 5 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_6	0x0618	W	0x00000000	Seed 6 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_7	0x061c	W	0x00000000	Seed 7 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_8	0x0620	W	0x00000000	Seed 8 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_9	0x0624	W	0x00000000	Seed 9 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_10	0x0628	W	0x00000000	Seed 10 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_11	0x062c	W	0x00000000	Seed 11 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_12	0x0630	W	0x00000000	Seed 12 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_13	0x0634	W	0x00000000	Seed 13 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_14	0x0638	W	0x00000000	Seed 14 for Toshiba 13 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED13_15	0x063c	W	0x00000000	Seed 15 for Toshiba 13 Power Polynomial Randomizer

Name	Offset	Size	Reset Value	Description
NANDC_RANDMZ_SE_ED17_0	0x0640	W	0x00000000	Seed 0 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_1	0x0644	W	0x00000000	Seed 1 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_2	0x0648	W	0x00000000	Seed 2 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_3	0x064c	W	0x00000000	Seed 3 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_4	0x0650	W	0x00000000	Seed 4 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_5	0x0654	W	0x00000000	Seed 5 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_6	0x0658	W	0x00000000	Seed 6 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_7	0x065c	W	0x00000000	Seed 7 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_8	0x0660	W	0x00000000	Seed 8 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_9	0x0664	W	0x00000000	Seed 9 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_10	0x0668	W	0x00000000	Seed 10 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_11	0x066c	W	0x00000000	Seed 11 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_12	0x0670	W	0x00000000	Seed 12 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_13	0x0674	W	0x00000000	Seed 13 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_14	0x0678	W	0x00000000	Seed 14 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED17_15	0x067c	W	0x00000000	Seed 15 for Toshiba 17 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_0	0x0680	W	0x00000000	Seed 0 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_1	0x0684	W	0x00000000	Seed 1 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_2	0x0688	W	0x00000000	Seed 2 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_3	0x068c	W	0x00000000	Seed 3 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_4	0x0690	W	0x00000000	Seed 4 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_5	0x0694	W	0x00000000	Seed 5 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_6	0x0698	W	0x00000000	Seed 6 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_7	0x069c	W	0x00000000	Seed 7 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_8	0x06a0	W	0x00000000	Seed 8 for Toshiba 19 Power Polynomial Randomizer

Name	Offset	Size	Reset Value	Description
NANDC_RANDMZ_SE_ED19_9	0x06a4	W	0x00000000	Seed 9 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_10	0x06a8	W	0x00000000	Seed 10 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_11	0x06ac	W	0x00000000	Seed 11 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_12	0x06b0	W	0x00000000	Seed 12 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_13	0x06b4	W	0x00000000	Seed 13 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_14	0x06b8	W	0x00000000	Seed 14 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED19_15	0x06bc	W	0x00000000	Seed 15 for Toshiba 19 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_0	0x06c0	W	0x00000000	Seed 0 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_1	0x06c4	W	0x00000000	Seed 1 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_2	0x06c8	W	0x00000000	Seed 2 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_3	0x06cc	W	0x00000000	Seed 3 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_4	0x06d0	W	0x00000000	Seed 4 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_5	0x06d4	W	0x00000000	Seed 5 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_6	0x06d8	W	0x00000000	Seed 6 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_7	0x06dc	W	0x00000000	Seed 7 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_8	0x06e0	W	0x00000000	Seed 8 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_9	0x06e4	W	0x00000000	Seed 9 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_10	0x06e8	W	0x00000000	Seed 10 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_11	0x06ec	W	0x00000000	Seed 11 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_12	0x06f0	W	0x00000000	Seed 12 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_13	0x06f4	W	0x00000000	Seed 13 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_14	0x06f8	W	0x00000000	Seed 14 for Toshiba 23 Power Polynomial Randomizer
NANDC_RANDMZ_SE_ED23_15	0x06fc	W	0x00000000	Seed 15 for Toshiba 23 Power Polynomial Randomizer

Notes: Size : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** - WORD (32 bits) access

12.4.3 Detail Register Description

NANDC_FMCTL

Address: Operational Base + offset (0x0000)

Flash Interface Control Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	syn_mode Toggle enable signal, 1 active. 0: ONFI synchronous flash. 1: Toggle synchronous flash.
14	RW	0x0	syn_clken Synchronous flash clock enable signal, 1 active. Only available in Synchronous Mode. 0: flash clock is disabled. 1: flash clock is enabled.
13	RW	0x0	tm Timing mode indication. 0: Asynchronous Mode. 1: Synchronous Mode (Toggle or ONFI Synchronous).
12	RW	0x0	dwidth Flash data bus width indication. 0: 8bits, active in both Asynchronous Mode flash and Synchronous Mode flash. 1: 16bits, active only in Asynchronous Mode flash.
11:10	RO	0x0	reserved
9	RO	0x1	frdy Flash ready/busy indicate signal. 0: flash is busy. 1: flash is ready. This bit is the sample of the pin of R/Bn.
8	RW	0x0	wp Flash write protect. 0: flash program/erase disabled. 1: flash program/erase enabled. This bit is output to the pin of WPn.
7	RW	0x0	fcs7 Flash memory chip 7 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
6	RW	0x0	fcs6 Flash memory chip 6 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
5	RW	0x0	fcs5 Flash memory chip 5 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.

Bit	Attr	Reset Value	Description
4	RW	0x0	fcs4 Flash memory chip 4 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
3	RW	0x0	fcs3 Flash memory chip 3 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
2	RW	0x0	fcs2 Flash memory chip 2 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
1	RW	0x0	fcs1 Flash memory chip 1 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
0	RW	0x0	fcs0 Flash memory chip 0 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.

NANDC_FMWAIT_ASYN

Address: Operational Base + offset (0x0004)

Flash Timing Control Register For Asynchronous Timing

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	fmw_dly_en fmw_dly enable signal,1 active.
29:24	RW	0x3f	fmw_dly The number of delay cycle between two codeword transmission.
23:18	RO	0x0	reserved
17:12	RW	0x3f	csrwr When in Asynchronous mode or Toggle address/command mode, this field specifies the number of processor clock cycles from the falling edge of CSn to the falling edge of RDn or WRn. The min value of csrwr is 0.
11	RO	0x0	reserved
10:5	RW	0x3f	rwpw When in Asynchronous mode or Toggle address/command mode, this field specifies the width of RDn or WRn in processor clock cycles, $0x0 \leq rwpw \leq 0x3f$.
4:0	RW	0x1f	rwcs When in Asynchronous mode or Toggle address/command mode, this field specifies the number of processor clock cycles from the rising edge of RDn or WRn to the rising edge of CSn, $0x0 \leq rwcs \leq 0x1f$.

NANDC_FLCTL

Address: Operational Base + offset (0x0008)

Internal Transfer Control Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RW	0x0	low_power Nandc low power control 0: normal mode 1: low power mode
27:22	RW	0x00	page_num Transmission codeword number in internal DMA mode when bus-mode is master-mode 1~32: 1~32 codeword. default: not support. Notes: a. Only active in internal DMA mode b. Only active when bus-mode is master-mode
21	RW	0x0	page_size Transmission codeword size in internal DMA mode 0: 1024bytes/codeword 1: 512bytes/codeword
20	RO	0x1	tr_rdy Internal DMA transmission ready indication. 0: internal DMA transmission is busy 1: internal DMA transmission is ready When reading flash, tr_rdy should not be set to 1 until all data transmission and correct finished. When programing flash, tr_rdy should not be set to 1 until all data transmission finished. Notes: Only active in internal DMA mode.
19	RO	0x0	reserved
18:12	RW	0x00	spare_size Spare byte number when lba_en=1. 0<= spare_size<=109. When spare_size>=109, it is treated as 0. Notes: The spare_size must be even number when flash is ONFI Synchronous Flash or Aynchronous Flash with 16bits data width.

Bit	Attr	Reset Value	Description
11	RW	0x0	<p>Iba_en LBA mode indication, 1 active. 0: NO-LBA mode, NandC should transfer both page data and spare data in every codeword, and the page size is 1024 bytes or 512 bytes determined by BCHCTL[16](bchpage), spare size is 32/46/74 bytes or 109 bytes determined by BCHCTL[4] and BCHCTL[18]. 1: LBA mode, NandC should transfer both page data and spare data in every codeword, and the page size is 1024 bytes or 512 bytes determined by FLCTL[21](page_size), spare size is determined by FLCTL[17:12](spare_size).</p> <p>Notes:</p> <ul style="list-style-type: none"> a. When Iba_en is active, BCH CODEC should be disabled, spare_size and page_size are configurable. b. When Iba_en is active, cor_able is inactive.
10	RW	0x0	<p>cor_able Auto correct enable indication, 1 active. 0: auto correct disable 1: auto correct enable</p> <p>Notes:</p> <ul style="list-style-type: none"> a. Only active in internal DMA mode. b. Iba_en is prior to cor_able. When Iba_en=1, cor_able is ignored.
9:8	RO	0x0	reserved
7	RW	0x0	<p>flash_st_mod Mode for NandC to start internal data transmission in internal DMA mode. 0: busy mode: hardware should not start internal data transmission until flash is ready even flash_st is asserted. 1: ready mode: hardware should start internal data transmission directly when flash_st is asserted.</p> <p>Notes: Only active in internal DMA mode.</p>
6:5	RW	0x0	<p>tr_count Transmission codeword number in internal DMA mode when bus-mode is slave-mode 00: 0 codeword need transferred 01: 1 codeword need transferred 10: 2 codeword need transferred 11: not supported</p> <p>Notes:</p> <ul style="list-style-type: none"> a. Only active in internal DMA mode. b. Only active when bus-mode is slave-mode.

Bit	Attr	Reset Value	Description
4	RW	0x0	<p>st_addr Start buffer address. 0: start transfer from sram0 1: start transfer from sram1 Notes: Only active in internal DMA mode.</p>
3	RW	0x0	<p>bypass NandC internal DMA bypass indication. 0: bypass the internal DMA, data are transferred to/from flash by direct path. 1: internal DMA active, data are transferred to/from flash by internal DMA.</p>
2	RW	0x0	<p>flash_st Start signal for NandC to transfer data between flash and internal buffer in internal DMA mode. When asserted, it will auto cleared. 0: not start transmission 1: start transmission Notes: Only active in internal DMA mode</p>
1	RW	0x0	<p>flash_rdn Indicate data flow direction. 0: NandC read data from flash. 1: NandC write data to flash</p>
0	RW	0x0	<p>flash_RST NandC software reset indication. When asserted, it will auto cleared. 0: not software reset 1: software reset Notes: flash_RST is prior to flash_st</p>

NANDC_BCHCTL

Address: Operational Base + offset (0x000c)

BCH Control Register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:19	RW	0x00	bchthres BCH error number threshold
18	RW	0x0	<p>bchmode1 High bit of BCH mode selection for 40bitBCH or 60bitBCH. BchMode=bchmode1, bchmode0: 00: 16bitBCH 01: 24bitBCH 10: 40bitBCH 11: 60bitBCH</p>
17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16	RW	0x0	<p>bchpage The data size indication when BCH is active. 0: 1024 bytes, all the 1024 bytes data in codeword are valid data to be transferred. 1: 512 bytes, higher 512bytes are valid, and lower 512bytes are invalid and stuffed with 0xff.</p> <p>Notes:</p> <ul style="list-style-type: none"> a. Only active when data transferred in internal DMA mode. b. Only active for asynchronous flash.
15:8	RW	0x00	<p>addr BCH active range selection. BCH should be active when access in range address.</p>
7:5	RW	0x0	<p>region BCH active region selection indication. 000: Flash memory 0 region (flash 0) 001: Flash memory 1 region (flash 1) 010: Flash memory 2 region (flash 2) 011: Flash memory 3 region (flash 3) 100: Flash memory 4 region (flash 4) 101: Flash memory 5 region (flash 5) 110: Flash memory 6 region (flash 6) 111: Flash memory 7 region (flash 7)</p>
4	RW	0x0	<p>bchmode0 BCH mode selection indication. BCH mode is determined by both bchmode0 and bchmode1,detailed information is showed in BCHCTL[18].</p>
3	RW	0x1	<p>bchepd BCH encoder/decoder power down indication. 0: BCH encoder/decoder working. 1: BCH encoder/decoder not working.</p>
2	RW	0x0	<p>mode_addrare BCH address care mode selection indication. 0: address care. 1: address not care.</p> <p>Notes: This bit is just active for data transmission in bypass mode, but not for command and address transmission.</p>
1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>bchrst BCH software reset indication, When asserted, it will auto cleared. 0: not software reset 1: software reset Notes: a. BCH Decoder should be software reset before decode begin. b. bch software reset should be used with nandc software reset at the same time.</p>

NANDC_MTRANS_CFG

Address: Operational Base + offset (0x0010)

Bus Transfer Configuration Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	W1C	0x0	<p>ahb_rst ahb master interface software reset, auto cleared</p>
14	RW	0x0	<p>fl_pwd Flash power down indication, 1 active. 0: Flash power on, data transferred through master interface is data that to be written into or read from flash. 1: Flash power down, data transferred through master interface is not data that to be written into or read from flash. NandC is just used as DMA for external memory and internal memory.</p>
13:9	RW	0x00	<p>incr_num AHB Master incr num indication. incr_num=1~16. When burst=001, software should configure incr_num. Notes: Only active for master-mode.</p>
8:6	RW	0x7	<p>burst AHB Master burst type indication: 000 : Single transfer 011 : 4-beat burst 101 : 8-beat Burst 111 : 16-beat burst default : not supported Notes: Only active for master-mode.</p>

Bit	Attr	Reset Value	Description
5:3	RW	0x2	<p>hsize AHB Master data size indication: 000 : 8 bits 001 : 16 bits 010 : 32 bits default : not supported Notes: Only active for master-mode.</p>
2	RW	0x0	<p>bus_mode Bus interface selection. 0: Slave interface, flash data is transferred through slave interface 1: Master interface, flash data is transferred through master interface</p>
1	RW	0x0	<p>ahb_wr Data transfer direction through master interface. 0: read direction(internal memory ->external memory) 1: write direction (internal memory->external memory) Notes: a. Only active for master-mode. b. When read flash(flash_rdn=0), ahb_wr=1; when program flash(flash_rdn=1), ahb_wr=0.</p>
0	W1C	0x0	<p>ahb_wr_st Start indication for loading data from external memory to internal memory or storing data from internal memory to external memory through master. When asserted, it will auto cleared. Notes: a. Only active for master-mode and fl_pwd=1. b. When fl_pwd=0, flash is active, NandC start to transfer data through master interface if flash_st=1 c. When fl_pwd=1, flash is not active, NandC start to transfer data through master interface if ahb_wr_st=1</p>

NANDC_MTRANS_SADDR0

Address: Operational Base + offset (0x0014)

Start Address Register For Page Data Transmission

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>saddr0 Start address for page data transmission. Notes: a. Only active for master-mode. b. Should be aligned with hsize in MTRANS_CFG[5:3].</p>

NANDC_MTRANS_SADDR1

Address: Operational Base + offset (0x0018)

Start Address Register For Spare Data Transmission

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	saddr1 Start address for spare data. Notes: a. Only active for master-mode. b. Should be aligned with hsize in MTRANS_CFG[5:3].

NANDC_MTRANS_STAT

Address: Operational Base + offset (0x001c)

Bus Transfer Status Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:16	RO	0x00	mtrans_cnt finished counter for codeword transmission through Master interface Notes: Only active for master-mode.
15:0	RO	0x0000	bus_err Bus error indication for codeword0~15. [0] : bus error for codeword 0 [15] : bus error for codeword 15 Notes: Only active for master-mode.

NANDC_BCHST0

Address: Operational Base + offset (0x0020)

BCH Status Register For Codeword 0~1

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	err_hnum1_h1 Highest bit of err_hnum1
29	RO	0x0	err_tnum1_h1 Highest bit of err_tnum1
28	RO	0x0	err_hnum0_h1 Highest bit of err_hnum0
27	RO	0x0	err_tnum0_h1 Highest bit of err_tnum0
26	RO	0x1	bchrdy Ready indication for bch encoder/decoder, 1 active. 0: bch encoder/decoder is busy 1: bch encoder/decoder is ready
25:21	RO	0x00	err_hnum1_l5 Lower 5 bits of number of error bits found in first 512bytes of 1st backup codeword

Bit	Attr	Reset Value	Description
20:16	RO	0x00	err_tnum1_l5 Lower 5 bits of number of error bits found in 1st backup codeword
15	RO	0x0	fail1 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
14	RO	0x0	done1 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
13	RO	0x0	errf1 Indication for error found in 1st backup codeword. 0: no error 1: error found
12:8	RO	0x00	err_hnum0_l5 Lower 5 bits of number of error bits found in first 512bytes of current backup codeword
7:3	RO	0x00	err_tnum0_l5 Lower 5 bits of number of error bits found in current backup codeword
2	RO	0x0	fail0 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	done0 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf0 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST1

Address: Operational Base + offset (0x0024)

BCH Status Register For Codeword 2~3

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	err_hnum3_h1 Highest bit of err_hnum3
29	RO	0x0	err_tnum3_h1 Highest bit of err_tnum3

Bit	Attr	Reset Value	Description
28	RO	0x0	err_hnum2_h1 Highest bit of err_hnum2
27	RO	0x0	err_tnum2_h1 Highest bit of err_tnum2
26	RO	0x0	reserved
25:21	RO	0x00	err_hnum3_l5 Lower 5 bits of number of error bits found in first 512bytes of 3th backup codeword
20:16	RO	0x00	err_tnum3_l5 Lower 5 bits of number of error bits found in 3th backup codeword
15	RO	0x0	fail3 Indication for the 3th backup codeword decoded failed or not. 0: decode successfully 1: decode fail
14	RO	0x0	done3 Indication for finishing decoding the 3th backup codeword 0: not finished 1: finished
13	RO	0x0	errf3 Indication for error found in 3th backup codeword. 0: no error 1: error found
12:8	RO	0x00	err_hnum2_l5 Lower 5 bits of number of error bits found in first 512bytes of 2th backup codeword
7:3	RO	0x00	err_tnum2_l5 Lower 5 bits of number of error bits found in 2th backup codeword
2	RO	0x0	fail2 Indication for 2th backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	done2 Indication for finishing decoding the 2th backup codeword. 0: not finished 1: finished
0	RO	0x0	errf2 Indication for error found in 2th backup codeword. 0: no error 1: error found

NANDC_BCHST2

Address: Operational Base + offset (0x0028)

BCH Status Register For Codeword 4~5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd4_cwd5 BCHST information for 4th and 5th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST3

Address: Operational Base + offset (0x002c)

BCH Status Register For Codeword 6~7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd6_cwd7 BCHST information for 6th and 7th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST4

Address: Operational Base + offset (0x0030)

BCH Status Register For Codeword 8~9

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd8_cwd9 BCHST information for 8th and 9th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST5

Address: Operational Base + offset (0x0034)

BCH Status Register For Codeword 10~11

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd10_cwd11 BCHST information for 10th and 11th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST6

Address: Operational Base + offset (0x0038)

BCH Status Register For Codeword 12~13

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd12_cwd13 BCHST information for 12th and 13th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST7

Address: Operational Base + offset (0x003c)

BCH Status Register For Codeword 14~15

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd14_cwd15 BCHST information for 14th and 15th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHLOC0

Address: Operational Base + offset (0x0040)

BCH Error Bit Location Number Register For Codeword 0~5

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RO	0x00	err_loc5_I5 Lower 5 bits of number of 8bit error location in 5th backup codeword
24:20	RO	0x00	err_loc4_I5 Lower 5 bits of number of 8bit error location in 4th backup codeword
19:15	RO	0x00	err_loc3_I5 Lower 5 bits of number of 8bit error location in 3rd backup codeword
14:10	RO	0x00	err_loc2_I5 Lower 5 bits of number of 8bit error location in 2nd backup codeword
9:5	RO	0x00	err_loc1_I5 Lower 5 bits of number of 8bit error location in 1st backup codeword
4:0	RO	0x00	err_loc0_I5 Lower 5 bits of number of 8bit error location in current backup codeword

NANDC_BCHLOC1

Address: Operational Base + offset (0x0044)

BCH Error Bit Location Number Register For Codeword 6~11

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29:25	RO	0x00	err_loc11_I5 Lower 5 bits of number of 8bit error location in 11th backup codeword
24:20	RO	0x00	err_loc10_I5 Lower 5 bits of number of 8bit error location in 10th backup codeword
19:15	RO	0x00	err_loc9_I5 Lower 5 bits of number of 8bit error location in 9th backup codeword
14:10	RO	0x00	err_loc8_I5 Lower 5 bits of number of 8bit error location in 8th backup codeword
9:5	RO	0x00	err_loc7_I5 Lower 5 bits of number of 8bit error location in 7th backup codeword
4:0	RO	0x00	err_loc6_I5 Lower 5 bits of number of 8bit error location in 6th backup codeword

NANDC_BCHLOC2

Address: Operational Base + offset (0x0048)

BCH Error Bit Location Number Register For Codeword 12~17

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RO	0x00	err_loc17_I5 Lower 5 bits of number of 8bit error location in 17th backup codeword
24:20	RO	0x00	err_loc16_I5 Lower 5 bits of number of 8bit error location in 16th backup codeword
19:15	RO	0x00	err_loc15_I5 Lower 5 bits of number of 8bit error location in 15th backup codeword
14:10	RO	0x00	err_loc14_I5 Lower 5 bits of number of 8bit error location in 14th backup codeword
9:5	RO	0x00	err_loc13_I5 Lower 5 bits of number of 8bit error location in 13th backup codeword
4:0	RO	0x00	err_loc12_I5 Lower 5 bits of number of 8bit error location in 12th backup codeword

NANDC_BCHLOC3

Address: Operational Base + offset (0x004c)

BCH Error Bit Location Number Register For Codeword 24~29

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29:25	RO	0x00	err_loc23_I5 Lower 5 bits of number of 8bit error location in 23th backup codeword
24:20	RO	0x00	err_loc22_I5 Lower 5 bits of number of 8bit error location in 22th backup codeword
19:15	RO	0x00	err_loc21_I5 Lower 5 bits of number of 8bit error location in 21th backup codeword
14:10	RO	0x00	err_loc20_I5 Lower 5 bits of number of 8bit error location in 20th backup codeword
9:5	RO	0x00	err_loc19_I5 Lower 5 bits of number of 8bit error location in 19th backup codeword
4:0	RO	0x00	err_loc18_I5 Lower 5 bits of number of 8bit error location in 18th backup codeword

NANDC_BCHLOC4

Address: Operational Base + offset (0x0050)

BCH Error Bit Location Number Register For Codeword 24~29

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RO	0x00	err_loc29_I5 Lower 5 bits of number of 8bit error location in 23th backup codeword
24:20	RO	0x00	err_loc28_I5 Lower 5 bits of number of 8bit error location in 22th backup codeword
19:15	RO	0x00	err_loc27_I5 Lower 5 bits of number of 8bit error location in 21th backup codeword
14:10	RO	0x00	err_loc26_I5 Lower 5 bits of number of 8bit error location in 20th backup codeword
9:5	RO	0x00	err_loc25_I5 Lower 5 bits of number of 8bit error location in 19th backup codeword
4:0	RO	0x00	err_loc24_I5 Lower 5 bits of number of 8bit error location in 18th backup codeword

NANDC_BCHLOC5

Address: Operational Base + offset (0x0054)

BCH Error Bit Location Number Register For Codeword 30~31

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:5	RO	0x00	err_loc31_l5 Lower 5 bits of number of 8bit error location in 31th backup codeword
4:0	RO	0x00	err_loc30_l5 Lower 5 bits of number of 8bit error location in 30th backup codeword

NANDC_BCHLOC6

Address: Operational Base + offset (0x0058)

Highest Bit For BCH Error Bit Location Number Register

Bit	Attr	Reset Value	Description
31	RO	0x0	err_loc31_h1 High bit for numbers of 8bit error location in 31th codeword
30	RO	0x0	err_loc30_h1 High bit for numbers of 8bit error location in 30th codeword
29	RO	0x0	err_loc29_h1 High bit for numbers of 8bit error location in 29th codeword
28	RO	0x0	err_loc28_h1 High bit for numbers of 8bit error location in 28th codeword
27	RO	0x0	err_loc27_h1 High bit for numbers of 8bit error location in 27th codeword
26	RO	0x0	err_loc26_h1 High bit for numbers of 8bit error location in 26th codeword
25	RO	0x0	err_loc25_h1 High bit for numbers of 8bit error location in 25th codeword
24	RO	0x0	err_loc24_h1 High bit for numbers of 8bit error location in 24th codeword
23	RO	0x0	err_loc23_h1 High bit for numbers of 8bit error location in 23th codeword
22	RO	0x0	err_loc22_h1 High bit for numbers of 8bit error location in 22th codeword
21	RO	0x0	err_loc21_h1 High bit for numbers of 8bit error location in 21th codeword
20	RO	0x0	err_loc20_h1 High bit for numbers of 8bit error location in 20th codeword
19	RO	0x0	err_loc19_h1 High bit for numbers of 8bit error location in 19th codeword

Bit	Attr	Reset Value	Description
18	RO	0x0	err_loc18_h1 High bit for numbers of 8bit error location in 18th codeword
17	RO	0x0	err_loc17_h1 High bit for numbers of 8bit error location in 17th codeword
16	RO	0x0	err_loc16_h1 High bit for numbers of 8bit error location in 16th codeword
15	RO	0x0	err_loc15_h1 High bit for numbers of 8bit error location in 15th codeword
14	RO	0x0	err_loc14_h1 High bit for numbers of 8bit error location in 14th codeword
13	RO	0x0	err_loc13_h1 High bit for numbers of 8bit error location in 13th codeword
12	RO	0x0	err_loc12_h1 High bit for numbers of 8bit error location in 12th codeword
11	RO	0x0	err_loc11_h1 High bit for numbers of 8bit error location in 11th codeword
10	RO	0x0	err_loc10_h1 High bit for numbers of 8bit error location in 10th codeword
9	RO	0x0	err_loc9_h1 High bit for numbers of 8bit error location in 9th codeword
8	RO	0x0	err_loc8_h1 High bit for numbers of 8bit error location in 8th codeword
7	RO	0x0	err_loc7_h1 High bit for numbers of 8bit error location in 7th codeword
6	RO	0x0	err_loc6_h1 High bit for numbers of 8bit error location in 6th codeword
5	RO	0x0	err_loc5_h1 High bit for numbers of 8bit error location in 5th codeword
4	RO	0x0	err_loc4_h1 High bit for numbers of 8bit error location in 4th codeword
3	RO	0x0	err_loc3_h1 High bit for numbers of 8bit error location in 3th codeword
2	RO	0x0	err_loc2_h1 High bit for numbers of 8bit error location in 2th codeword

Bit	Attr	Reset Value	Description
1	RO	0x0	err_loc1_h1 High bit for numbers of 8bit error location in 1th codeword
0	RO	0x0	err_loc0_h1 High bit for numbers of 8bit error location in 0th codeword

NANDC_BCHDE0_0

Address: Operational Base + offset (0x0070)

BCH decode result of 0th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:8	RO	0x000	offset The offset byte address of the error bit. The value is 11bit, which is the byte offset address in the codeword. The address can be divided into different part for different use, showed as follows. 0 ~1023: page data 1024~1027: system information 1028~1055: bch information for 16bitBCH 1028~1069: bch information for 24bitBCH 1028~1097: bch information for 40bitBCH 1028~1132: bch information for 60bitBCH
7:0	RO	0x00	err_val The error value of corresponding error byte

NANDC_BCHDE0_1

Address: Operational Base + offset (0x0074)

BCH decode result of 1th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_1 decode result of 1th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_2

Address: Operational Base + offset (0x0078)

BCH decode result of 2th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_2 decode result of 2th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_3

Address: Operational Base + offset (0x007c)

BCH decode result of 3th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_3 decode result of 3th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_4

Address: Operational Base + offset (0x0080)

BCH decode result of 4th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_4 decode result of 4th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_5

Address: Operational Base + offset (0x0084)

BCH decode result of 5th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_5 decode result of 5th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_6

Address: Operational Base + offset (0x0088)

BCH decode result of 6th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_6 decode result of 6th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_7

Address: Operational Base + offset (0x008c)

BCH decode result of 7th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_7 decode result of 7th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_8

Address: Operational Base + offset (0x0090)

BCH decode result of 8th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_8 decode result of 8th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_9

Address: Operational Base + offset (0x0094)

BCH decode result of 9th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_9 decode result of 9th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_10

Address: Operational Base + offset (0x0098)

BCH decode result of 10th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_10 decode result of 10th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_11

Address: Operational Base + offset (0x009c)

BCH decode result of 11th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_11 decode result of 11th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_12

Address: Operational Base + offset (0x00a0)

BCH decode result of 12th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_12 decode result of 12th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_13

Address: Operational Base + offset (0x00a4)

BCH decode result of 13th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_13 decode result of 13th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_14

Address: Operational Base + offset (0x00a8)

BCH decode result of 14th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_14 decode result of 14th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_15

Address: Operational Base + offset (0x00ac)

BCH decode result of 15th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_15 decode result of 15th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_16

Address: Operational Base + offset (0x00b0)

BCH decode result of 16th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_16 decode result of 16th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_17

Address: Operational Base + offset (0x00b4)

BCH decode result of 17th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_17 decode result of 17th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_18

Address: Operational Base + offset (0x00b8)

BCH decode result of 18th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_18 decode result of 18th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_19

Address: Operational Base + offset (0x00bc)

BCH decode result of 19th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_19 decode result of 1th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_20

Address: Operational Base + offset (0x00c0)

BCH decode result of 20th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_20 decode result of 20th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_21

Address: Operational Base + offset (0x00c4)

BCH decode result of 21th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_21 decode result of 21th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_22

Address: Operational Base + offset (0x00c8)

BCH decode result of 22th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_22 decode result of 22th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_23

Address: Operational Base + offset (0x00cc)

BCH decode result of 23th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_23 decode result of 23th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE1_0

Address: Operational Base + offset (0x00d0)

BCH decode result of 0th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:8	RO	0x000	offset The offset byte address of the error bit. The value is 11bit, which is the byte offset address in the codeword. The address can be divided into different part for different use, showed as follows. 0 ~1023: page data 1024~1027: system information 1028~1055: bch information for 16bitBCH 1028~1069: bch information for 24bitBCH 1028~1097: bch information for 40bitBCH 1028~1132: bch information for 60bitBCH
7:0	RO	0x00	err_val The error value of corresponding error byte

NANDC_BCHDE1_1

Address: Operational Base + offset (0x00d4)

BCH decode result of 1th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_1 decode result of 1th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_2

Address: Operational Base + offset (0x00d8)

BCH decode result of 2th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_2 decode result of 2th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_3

Address: Operational Base + offset (0x00dc)

BCH decode result of 3th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_3 decode result of 3th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_4

Address: Operational Base + offset (0x00e0)

BCH decode result of 4th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_4 decode result of 4th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_5

Address: Operational Base + offset (0x00e4)

BCH decode result of 5th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_5 decode result of 5th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_6

Address: Operational Base + offset (0x00e8)

BCH decode result of 6th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_6 decode result of 6th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_7

Address: Operational Base + offset (0x00ec)

BCH decode result of 7th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_7 decode result of 7th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_8

Address: Operational Base + offset (0x00f0)

BCH decode result of 8th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_8 decode result of 8th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_9

Address: Operational Base + offset (0x00f4)

BCH decode result of 9th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_9 decode result of 9th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_10

Address: Operational Base + offset (0x00f8)

BCH decode result of 10th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_10 decode result of 10th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_11

Address: Operational Base + offset (0x00fc)

BCH decode result of 11th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_11 decode result of 11th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_12

Address: Operational Base + offset (0x0100)

BCH decode result of 12th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_12 decode result of 12th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_13

Address: Operational Base + offset (0x0104)

BCH decode result of 13th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_13 decode result of 13th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_14

Address: Operational Base + offset (0x0108)

BCH decode result of 14th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_14 decode result of 14th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_15

Address: Operational Base + offset (0x010c)

BCH decode result of 15th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_15 decode result of 15th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_16

Address: Operational Base + offset (0x0110)

BCH decode result of 16th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_16 decode result of 16th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_17

Address: Operational Base + offset (0x0114)

BCH decode result of 17th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_17 decode result of 17th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_18

Address: Operational Base + offset (0x0118)

BCH decode result of 18th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_18 decode result of 18th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_19

Address: Operational Base + offset (0x011c)

BCH decode result of 19th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_19 decode result of 19th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_20

Address: Operational Base + offset (0x0120)

BCH decode result of 20th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_20 decode result of 20th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_21

Address: Operational Base + offset (0x0124)

BCH decode result of 21th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_21 decode result of 21th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_22

Address: Operational Base + offset (0x0128)

BCH decode result of 22th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_22 decode result of 22th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_23

Address: Operational Base + offset (0x012c)

BCH decode result of 23th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_23 decode result of 23th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_DLL_CTL_REG0

Address: Operational Base + offset (0x0130)

DLL Control Register 0

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x7f	dll_dqs_dly_bypass Holds the read DQS delay setting when the DLL is operating in bypass mode.
15:8	RW	0x7f	dll_dqs_dly Holds the read DQS delay setting when the DLL is operating in normal mode. Typically, this value is 1/4 of a clock cycle. Each increment of this field represents 1/128th of a clock cycle.

Bit	Attr	Reset Value	Description
7:0	RW	0x05	dll_start_point DLL Start Point Control. This value is loaded into the DLL at initialization and is the value at which the DLL will begin searching for a lock. Each increment of this field represents 1/128th of a clock cycle.

NANDC_DLL_CTL_REG1

Address: Operational Base + offset (0x0134)

DLL Control Register 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:4	RW	0x02	dll_incr DLL Increment Value. This sets the increment used by the DLL when searching for a lock. It is recommended keeping this field small (around 0x4) to keep the steps gradual
3:2	RW	0x0	dll_qtren Quarter flag of DLL, active in no-bypass mode. 01:1/4 fclk, dqs_dly=128. 10:1/8 fclk, dqs_dly=64. Default: dqs_dly=dll_dqs_dly(DLL_CTL_REG0[15:8]). When dll_qtr='b01 or 'b10, software not need to configure dll_dqs_dly , and hardware should delay the input signal for 1/4 or 1/8 fclk cycle time; When dll_qtr=0, software need to configure dll_dqs_dly.
1	RW	0x1	dll_bypass DLL Bypass Control, 1active 0: dll not bypass, dll_dqs_dley= dqs_dly 1: dll bypass, dll_dqs_dley= dll_dqs_dley_bypass
0	RW	0x0	dll_start Start signal for DLL, 1 active. Notes: It will keep high until dll disabled.

NANDC_DLL_OBS_REG0

Address: Operational Base + offset (0x0138)

DLL Status Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:9	RO	0x01	dll_dqs_delay_value Report the delay value for the read DQS signal

Bit	Attr	Reset Value	Description
8:1	RO	0x00	dll_lock_value Reports the DLL encoder value from the master DLL to the slave DLL's. The slaves use this value to set up their delays for the clk_wr and read DQS signals.
0	RO	0x0	dll_lock DLL Lock indication: 0: DLL has not locked 1: DLL is locked.

NANDC_RANDMZ_CFG

Address: Operational Base + offset (0x0150)

Randomizer Configure Register

Bit	Attr	Reset Value	Description
31	RW	0x0	randmz_en Randomizer enable indication, 1 active. 0: Randomizer active 1: Randomizer not active Notes: a. Not active when data transmission in bypass mode. b. Just active for data, but not for address and command. c. Not active when BchPage=1.
30:29	RW	0x0	randmz_mode Randomizer mode: 00- Samsung randomizer Polynomial= $1+x+x^{15}$ 10- Samsung randomizer Polynomial= $1+x^{14}+x^{15}$ 01-TOSHIBA randomizer
28:24	RW	0x00	page_offset basic seed rotation bits for every 16page
23:20	RW	0x0	cwd_offset basic seed start point for every page
19:0	RW	0x00000	randmz_seed when Samsung randomizer: The seed for randomizer(initial value); when Toshiba randomizer: Seed Agitation Register.

NANDC_FMWAIT_SYN

Address: Operational Base + offset (0x0158)

Flash Timing Control Register For Synchronous Timing

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:9	RW	0x00	pst Write/Read Postamble time for ONFI synchronous mode or Toggle data mode. This field specifies the number of processor clock cycle for Postamble time.
8:3	RW	0x00	pre Write/Read Preamble time for ONFI synchronous mode or Toggle data mode. This field specifies the number of processor clock cycle for preamble time.
2:0	RW	0x0	fclk Half hclk cycle number for flash clock for ONFI synchronous mode or Toggle data mode

NANDC_MTRANS_STAT2

Address: Operational Base + offset (0x015c)

Bus Transfer Status Register2

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	bus_err2 Bus error indication for codeword16~31. [0] : bus error for codeword 16 [15] : bus error for codeword 31 Notes: Only active for master-mode.

NANDC_NANDC_VER

Address: Operational Base + offset (0x0160)

Nandc Version Register

Bit	Attr	Reset Value	Description
31:0	RO	0x56363232	version Version indication for NANDC

NANDC_LLPC_CTL

Address: Operational Base + offset (0x0164)

LLP Control Register

Bit	Attr	Reset Value	Description
31:6	RW	0x00000000	llp_loc Starting address for LLI0, 64byte align
5	RW	0x0	llp_frdy Working time for FOP_WAIT_FRDY for all FOP in first LLP group: 0: FOP_WAIT_FRDY begin working when started 1: FOP_WAIT_FRDY not begin working until 16 cycles later after started
4:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	llp_rst Reset signal for LLP. When asserted, it will auto cleared.
1	RW	0x0	llp_mode 0-current LLI only has FOP 1-current LLI has both CFG and FOP
0	RW	0x0	llp_en Enable signal for LLP 0-LLP disable 1-LLP enable

NANDC_LL_P_STAT

Address: Operational Base + offset (0x0168)

LLP Status Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	llp_stat latest LLI_LOC finished, 64byte align
5:2	RO	0x0	reserved
1	RO	0x0	llp_err error status for llp load or execute 0-llp is correct 1-llp is error
0	RO	0x1	llp_rdy ready status for all llp load 0-llp load is busy 1-llp load is ready

NANDC_INTEN

Address: Operational Base + offset (0x016c)

NandC Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	llp_int_en Enable for LLP finished interrupt. 0: interrupt disable 1: interrupt enable When llp_en_en is active, an interrupt is generated if LLP operation is finished
3	RW	0x0	bchfail_int_en Enable for bch fail interrupt. 0-interrupt disable 1-interrupt enable When bchfail_int_en is active, an interrupt is generated if bch decode failed

Bit	Attr	Reset Value	Description
2	RW	0x0	bcherr_int_en Enable for bch error interrupt. 0-interrupt disable 1-interrupt enable When bcherr_int_en is active, an interrupt is generated if bch decode error bit is larger than bchthres(BCHCTL[26:19])
1	RW	0x0	frdy_int_en Enable for flash_rdy interrupt 0-interrupt disable 1-interrupt enable When frdy_int_en is active, an interrupt is generated if flash R/B# changes from 0 to 1
0	RW	0x0	dma_int_en Enable for internal DMA transfer finished interrupt 0-interrupt disable 1-interrupt enable When dma_int_en is active, an interrupt is generated if page_num(FLCTL[27:22]) of flash data transfer in DMA mode is finished

NANDC_INTCLR

Address: Operational Base + offset (0x0170)

NandC Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	llp_int_clr Clear for LLP finished interrupt. When asserted, this bit will be auto cleared. 0: interrupt not cleared 1: interrupt cleared
3	RW	0x0	bchfail_int_clr Clear for bch decode fail interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared
2	RW	0x0	bcherr_int_clr Clear for bch error interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared
1	RW	0x0	frdy_int_clr Clear for flash_rdy interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared

Bit	Attr	Reset Value	Description
0	RW	0x0	dma_int_clr Clear for internal DMA transfer finished interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared

NANDC_INTST

Address: Operational Base + offset (0x0174)

NandC Interrupt Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	llp_int_stat Status for LLP finished interrupt, high active
3	RO	0x0	bchfail_int_stat Status for bch decode fail interrupt, high active
2	RO	0x0	bcherr_int_stat Status for bch error interrupt, high active
1	RO	0x0	frdy_int_stat Status for flash_rdy interrupt, high active
0	RO	0x0	dma_int_stat Status for internal DMA transfer finished interrupt, high active

NANDC_SPARE0_0

Address: Operational Base + offset (0x0200)

System Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0xff	system_3 the 4th system byte of codeword 0
23:16	RW	0xff	system_2 the 3rd system byte of codeword 0
15:8	RW	0xff	system_1 the 2nd system byte of codeword 0
7:0	RW	0xff	system_0 the 1st system byte of codeword 0

NANDC_SPARE0_1

Address: Operational Base + offset (0x0204)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_2

Address: Operational Base + offset (0x0208)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_3

Address: Operational Base + offset (0x020c)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_4

Address: Operational Base + offset (0x0210)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_5

Address: Operational Base + offset (0x0214)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_6

Address: Operational Base + offset (0x0218)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_7

Address: Operational Base + offset (0x021c)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_8

Address: Operational Base + offset (0x0220)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_9

Address: Operational Base + offset (0x0224)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_10

Address: Operational Base + offset (0x0228)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_11

Address: Operational Base + offset (0x022c)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_0

Address: Operational Base + offset (0x0230)

System Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0xff	system_3 the 4th system byte of codeword 1
23:16	RW	0xff	system_2 the 3rd system byte of codeword 1
15:8	RW	0xff	system_1 the 2nd system byte of codeword 1
7:0	RW	0xff	system_0 the 1st system byte of codeword 1

NANDC_SPARE1_1

Address: Operational Base + offset (0x0234)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_2

Address: Operational Base + offset (0x0238)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_3

Address: Operational Base + offset (0x023c)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_4

Address: Operational Base + offset (0x0240)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_5

Address: Operational Base + offset (0x0244)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_6

Address: Operational Base + offset (0x0248)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_7

Address: Operational Base + offset (0x024c)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_8

Address: Operational Base + offset (0x0250)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_9

Address: Operational Base + offset (0x0254)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_10

Address: Operational Base + offset (0x0258)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_11

Address: Operational Base + offset (0x025c)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_12

Address: Operational Base + offset (0x0260)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_13

Address: Operational Base + offset (0x0264)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_14

Address: Operational Base + offset (0x0268)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_15

Address: Operational Base + offset (0x026c)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_16

Address: Operational Base + offset (0x0270)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_17

Address: Operational Base + offset (0x0274)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_18

Address: Operational Base + offset (0x0278)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_19

Address: Operational Base + offset (0x027c)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_20

Address: Operational Base + offset (0x0280)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_21

Address: Operational Base + offset (0x0284)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_22

Address: Operational Base + offset (0x0288)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_23

Address: Operational Base + offset (0x028c)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_24

Address: Operational Base + offset (0x0290)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_25

Address: Operational Base + offset (0x0294)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_26

Address: Operational Base + offset (0x0298)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE0_27

Address: Operational Base + offset (0x029c)

Spare Data and BCH Encode Information for codeword 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH0_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH0_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH0_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH0_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_12

Address: Operational Base + offset (0x02a0)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_13

Address: Operational Base + offset (0x02a4)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_14

Address: Operational Base + offset (0x02a8)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_15

Address: Operational Base + offset (0x02ac)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_16

Address: Operational Base + offset (0x02b0)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_17

Address: Operational Base + offset (0x02b4)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_18

Address: Operational Base + offset (0x02b8)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_19

Address: Operational Base + offset (0x02bc)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_20

Address: Operational Base + offset (0x02c0)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_21

Address: Operational Base + offset (0x02c4)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_22

Address: Operational Base + offset (0x02c8)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_23

Address: Operational Base + offset (0x02cc)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_24

Address: Operational Base + offset (0x02d0)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_25

Address: Operational Base + offset (0x02d4)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25

Bit	Attr	Reset Value	Description
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_26

Address: Operational Base + offset (0x02d8)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_SPARE1_27

Address: Operational Base + offset (0x02dc)

Spare Data and BCH Encode Information for codeword 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	BCH1_4x_3 (4x+3)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
23:16	RW	0x00	BCH1_4x_2 (4x+2)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~9 40bitBCH: x=0~16 60bitBCH: x=0~25
15:8	RW	0x00	BCH1_4x_1 (4x+1)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~25
7:0	RW	0x00	BCH1_4x_0 (4x+0)th byte of bch spare bits 16bitBCH: x=0~6 24bitBCH: x=0~10 40bitBCH: x=0~17 60bitBCH: x=0~26

NANDC_BCHDE0_24

Address: Operational Base + offset (0x0400)

BCH decode result of 24th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_24 decode result of 24th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_25

Address: Operational Base + offset (0x0404)

BCH decode result of 25th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_25 decode result of 25th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_26

Address: Operational Base + offset (0x0408)

BCH decode result of 26th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_26 decode result of 26th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_27

Address: Operational Base + offset (0x040c)

BCH decode result of 27th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_27 decode result of 27th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_28

Address: Operational Base + offset (0x0410)

BCH decode result of 28th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_28 decode result of 28th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_29

Address: Operational Base + offset (0x0414)

BCH decode result of 29th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_29 decode result of 29th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_30

Address: Operational Base + offset (0x0418)

BCH decode result of 30th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_30 decode result of 30th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_31

Address: Operational Base + offset (0x041c)

BCH decode result of 31th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_31 decode result of 31th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_32

Address: Operational Base + offset (0x0420)

BCH decode result of 32th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_3 decode result of 3th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_33

Address: Operational Base + offset (0x0424)

BCH decode result of 33th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_33 decode result of 33th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_34

Address: Operational Base + offset (0x0428)

BCH decode result of 34th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_34 decode result of 34th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_35

Address: Operational Base + offset (0x042c)

BCH decode result of 35th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_35 decode result of 35th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_36

Address: Operational Base + offset (0x0430)

BCH decode result of 36th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_36 decode result of 36th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_37

Address: Operational Base + offset (0x0434)

BCH decode result of 37th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_37 decode result of 37th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_38

Address: Operational Base + offset (0x0438)

BCH decode result of 38th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_38 decode result of 38th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_39

Address: Operational Base + offset (0x043c)

BCH decode result of 39th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_39 decode result of 39th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_40

Address: Operational Base + offset (0x0440)

BCH decode result of 40th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_40 decode result of 40th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_41

Address: Operational Base + offset (0x0444)

BCH decode result of 41th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_41 decode result of 41th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_42

Address: Operational Base + offset (0x0448)

BCH decode result of 42th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_42 decode result of 4th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_43

Address: Operational Base + offset (0x044c)

BCH decode result of 43th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_43 decode result of 43th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_44

Address: Operational Base + offset (0x0450)

BCH decode result of 44th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_44 decode result of 44th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_45

Address: Operational Base + offset (0x0454)

BCH decode result of 45th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_45 decode result of 45th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_46

Address: Operational Base + offset (0x0458)

BCH decode result of 46th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_46 decode result of 46th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_47

Address: Operational Base + offset (0x045c)

BCH decode result of 47th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_47 decode result of 47th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_48

Address: Operational Base + offset (0x0460)

BCH decode result of 48th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_48 decode result of 48th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_49

Address: Operational Base + offset (0x0464)

BCH decode result of 49th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_49 decode result of 49th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_50

Address: Operational Base + offset (0x0468)

BCH decode result of 50th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_50 decode result of 50th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_51

Address: Operational Base + offset (0x046c)

BCH decode result of 51th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_51 decode result of 51th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_52

Address: Operational Base + offset (0x0470)

BCH decode result of 52th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_52 decode result of 52th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_53

Address: Operational Base + offset (0x0474)

BCH decode result of 53th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_53 decode result of 53th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_54

Address: Operational Base + offset (0x0478)

BCH decode result of 54th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_54 decode result of 54th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_55

Address: Operational Base + offset (0x047c)

BCH decode result of 55th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_55 decode result of 55th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_56

Address: Operational Base + offset (0x0480)

BCH decode result of 56th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_56 decode result of 56th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_57

Address: Operational Base + offset (0x0484)

BCH decode result of 57th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_57 decode result of 57th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_58

Address: Operational Base + offset (0x0488)

BCH decode result of 58th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_58 decode result of 58th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_59

Address: Operational Base + offset (0x048c)

BCH decode result of 59th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_59 decode result of 59th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE1_24

Address: Operational Base + offset (0x0490)

BCH decode result of 24th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_24 decode result of 24th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_25

Address: Operational Base + offset (0x0494)

BCH decode result of 25th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_25 decode result of 25th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_26

Address: Operational Base + offset (0x0498)

BCH decode result of 26th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_26 decode result of 26th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_27

Address: Operational Base + offset (0x049c)

BCH decode result of 27th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_27 decode result of 27th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_28

Address: Operational Base + offset (0x04a0)

BCH decode result of 28th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_28 decode result of 28th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_29

Address: Operational Base + offset (0x04a4)

BCH decode result of 29th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_29 decode result of 2th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_30

Address: Operational Base + offset (0x04a8)

BCH decode result of 30th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_30 decode result of 30th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_31

Address: Operational Base + offset (0x04ac)

BCH decode result of 31th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_31 decode result of 31th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_32

Address: Operational Base + offset (0x04b0)

BCH decode result of 32th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_32 decode result of 32th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_33

Address: Operational Base + offset (0x04b4)

BCH decode result of 33th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_33 decode result of 33th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_34

Address: Operational Base + offset (0x04b8)

BCH decode result of 34th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_34 decode result of 34th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_35

Address: Operational Base + offset (0x04bc)

BCH decode result of 35th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_35 decode result of 35th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_36

Address: Operational Base + offset (0x04c0)

BCH decode result of 36th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_36 decode result of 36th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_37

Address: Operational Base + offset (0x04c4)

BCH decode result of 37th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_37 decode result of 37th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_38

Address: Operational Base + offset (0x04c8)

BCH decode result of 38th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_38 decode result of 38th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_39

Address: Operational Base + offset (0x04cc)

BCH decode result of 39th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_39 decode result of 39th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_40

Address: Operational Base + offset (0x04d0)

BCH decode result of 40th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_40 decode result of 40th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_41

Address: Operational Base + offset (0x04d4)

BCH decode result of 41th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_41 decode result of 41th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_42

Address: Operational Base + offset (0x04d8)

BCH decode result of 42th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_42 decode result of 42th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_43

Address: Operational Base + offset (0x04dc)

BCH decode result of 43th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_43 decode result of 43th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_44

Address: Operational Base + offset (0x04e0)

BCH decode result of 44th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_44 decode result of 44th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_45

Address: Operational Base + offset (0x04e4)

BCH decode result of 45th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_45 decode result of 45th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_46

Address: Operational Base + offset (0x04e8)

BCH decode result of 46th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_46 decode result of 46th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_47

Address: Operational Base + offset (0x04ec)

BCH decode result of 47th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_47 decode result of 47th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_48

Address: Operational Base + offset (0x04f0)

BCH decode result of 48th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_48 decode result of 48th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_49

Address: Operational Base + offset (0x04f4)

BCH decode result of 49th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_49 decode result of 49th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_50

Address: Operational Base + offset (0x04f8)

BCH decode result of 50th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_50 decode result of 50th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_51

Address: Operational Base + offset (0x04fc)

BCH decode result of 51th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_51 decode result of 51th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_52

Address: Operational Base + offset (0x0500)

BCH decode result of 52th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_52 decode result of 52th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_53

Address: Operational Base + offset (0x0504)

BCH decode result of 53th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_53 decode result of 53th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_54

Address: Operational Base + offset (0x0508)

BCH decode result of 54th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_54 decode result of 54th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_55

Address: Operational Base + offset (0x050c)

BCH decode result of 55th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_55 decode result of 55th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_56

Address: Operational Base + offset (0x0510)

BCH decode result of 56th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_56 decode result of 56th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_57

Address: Operational Base + offset (0x0514)

BCH decode result of 57th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde1_57 decode result of 57th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_58

Address: Operational Base + offset (0x0518)

BCH decode result of 58th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_58 decode result of 58th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHDE1_59

Address: Operational Base + offset (0x051c)

BCH decode result of 59th error bit for codeword 1

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde1_59 decode result of 59th error bit for codeword 1. Bit assignment is similar to BCHDE1_0 register. For more description, please refer to BCHDE1_0 register.

NANDC_BCHST8

Address: Operational Base + offset (0x0520)

BCH Status Register For Codeword 16~17

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd16_cwd17 BCHST information for 16th and 17th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST9

Address: Operational Base + offset (0x0524)

BCH Status Register For Codeword 18~19

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd18_cwd19 BCHST information for 18th and 19th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST10

Address: Operational Base + offset (0x0528)

BCH Status Register For Codeword 20~21

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd20_cwd21 BCHST information for 20th and 21th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST11

Address: Operational Base + offset (0x052c)

BCH Status Register For Codeword 22~23

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd22_cwd23 BCHST information for 22th and 23th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST12

Address: Operational Base + offset (0x0530)

BCH Status Register For Codeword 24~25

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd24_cwd25 BCHST information for 24th and 25th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST13

Address: Operational Base + offset (0x0534)

BCH Status Register For Codeword 26~27

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd26_cwd27 BCHST information for 26th and 27th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST14

Address: Operational Base + offset (0x0538)

BCH Status Register For Codeword 28~29

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd28_cwd29 BCHST information for 28th and 29th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST15

Address: Operational Base + offset (0x053c)

BCH Status Register For Codeword 30~31

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd30_cwd31 BCHST information for 30th and 31th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_RANDMZ_SEED13_0

Address: Operational Base + offset (0x0600)

Seed 0 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_0 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_1

Address: Operational Base + offset (0x0604)

Seed 1 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_1 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_2

Address: Operational Base + offset (0x0608)

Seed 2 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_2 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_3

Address: Operational Base + offset (0x060c)

Seed 3 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_3 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_4

Address: Operational Base + offset (0x0610)

Seed 4 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_4 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_5

Address: Operational Base + offset (0x0614)

Seed 5 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_5 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_6

Address: Operational Base + offset (0x0618)

Seed 6 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_6 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_7

Address: Operational Base + offset (0x061c)

Seed 7 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_7 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_8

Address: Operational Base + offset (0x0620)

Seed 8 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_8 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_9

Address: Operational Base + offset (0x0624)

Seed 9 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_9 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_10

Address: Operational Base + offset (0x0628)

Seed 10 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_10 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_11

Address: Operational Base + offset (0x062c)

Seed 11 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_11 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_12

Address: Operational Base + offset (0x0630)

Seed 12 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_12 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_13

Address: Operational Base + offset (0x0634)

Seed 13 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:0	RW	0x0000	randmz_seed13_13 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_14

Address: Operational Base + offset (0x0638)

Seed 14 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_14 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED13_15

Address: Operational Base + offset (0x063c)

Seed 15 for Toshiba 13 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	randmz_seed13_15 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_0

Address: Operational Base + offset (0x0640)

Seed 0 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_0 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_1

Address: Operational Base + offset (0x0644)

Seed 1 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_1 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_2

Address: Operational Base + offset (0x0648)

Seed 2 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_2 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_3

Address: Operational Base + offset (0x064c)

Seed 3 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_3 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_4

Address: Operational Base + offset (0x0650)

Seed 4 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_4 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_5

Address: Operational Base + offset (0x0654)

Seed 5 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_5 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_6

Address: Operational Base + offset (0x0658)

Seed 6 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_6 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_7

Address: Operational Base + offset (0x065c)

Seed 7 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_7 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_8

Address: Operational Base + offset (0x0660)

Seed 8 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_8 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_9

Address: Operational Base + offset (0x0664)

Seed 9 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_9 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_10

Address: Operational Base + offset (0x0668)

Seed 10 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_10 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_11

Address: Operational Base + offset (0x066c)

Seed 11 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_11 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_12

Address: Operational Base + offset (0x0670)

Seed 12 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_12 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_13

Address: Operational Base + offset (0x0674)

Seed 13 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_13 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_14

Address: Operational Base + offset (0x0678)

Seed 14 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_14 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED17_15

Address: Operational Base + offset (0x067c)

Seed 15 for Toshiba 17 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	randmz_seed17_15 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_0

Address: Operational Base + offset (0x0680)

Seed 0 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_0 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_1

Address: Operational Base + offset (0x0684)

Seed 1 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_1 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_2

Address: Operational Base + offset (0x0688)

Seed 2 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_2 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_3

Address: Operational Base + offset (0x068c)

Seed 3 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_3 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_4

Address: Operational Base + offset (0x0690)

Seed 4 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_4 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_5

Address: Operational Base + offset (0x0694)

Seed 5 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_5 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_6

Address: Operational Base + offset (0x0698)

Seed 6 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_6 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_7

Address: Operational Base + offset (0x069c)

Seed 7 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_7 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_8

Address: Operational Base + offset (0x06a0)

Seed 8 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_8 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_9

Address: Operational Base + offset (0x06a4)

Seed 9 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_9 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_10

Address: Operational Base + offset (0x06a8)

Seed 10 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_10 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_11

Address: Operational Base + offset (0x06ac)

Seed 11 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_11 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_12

Address: Operational Base + offset (0x06b0)

Seed 12 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_12 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_13

Address: Operational Base + offset (0x06b4)

Seed 13 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_13 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_14

Address: Operational Base + offset (0x06b8)

Seed 14 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_14 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED19_15

Address: Operational Base + offset (0x06bc)

Seed 15 for Toshiba 19 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	randmz_seed19_15 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_0

Address: Operational Base + offset (0x06c0)

Seed 0 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_0 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_1

Address: Operational Base + offset (0x06c4)

Seed 1 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_1 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_2

Address: Operational Base + offset (0x06c8)

Seed 2 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_2 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_3

Address: Operational Base + offset (0x06cc)

Seed 3 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_3 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_4

Address: Operational Base + offset (0x06d0)

Seed 4 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_4 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_5

Address: Operational Base + offset (0x06d4)

Seed 5 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_5 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_6

Address: Operational Base + offset (0x06d8)

Seed 6 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved

Bit	Attr	Reset Value	Description
22:0	RW	0x000000	randmz_seed23_6 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_7

Address: Operational Base + offset (0x06dc)

Seed 7 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_7 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_8

Address: Operational Base + offset (0x06e0)

Seed 8 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_8 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_9

Address: Operational Base + offset (0x06e4)

Seed 9 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_9 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_10

Address: Operational Base + offset (0x06e8)

Seed 10 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_10 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_11

Address: Operational Base + offset (0x06ec)

Seed 11 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_11 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_12

Address: Operational Base + offset (0x06f0)

Seed 12 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_12 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_13

Address: Operational Base + offset (0x06f4)

Seed 13 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_13 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_14

Address: Operational Base + offset (0x06f8)

Seed 14 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_14 The seed for randomizer(initial value);

NANDC_RANDMZ_SEED23_15

Address: Operational Base + offset (0x06fc)

Seed 15 for Toshiba 23 Power Polynomial Randomizer

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:0	RW	0x000000	randmz_seed23_15 The seed for randomizer(initial value);

12.5 Interface Description

Table 12-2 NandC Interface Description

Module Pin	IO	Pad Name	IOMUX Setting
flash_wp	O	IO_NANDwp_EMMCpwren_GPIO2a5	GPIO2A_IOMUX[11:10] = 2'b01
flash_ale	O	IO_NANDale_SPI1clk_GPIO2a0	GPIO2A_IOMUX[1:0] = 2'b01
flash_cle	O	IO_NANDcle_GPIO2a1	GPIO2A_IOMUX[3:2]

			= 2'b01
flash_wrn	O	IO_NANDwrn_SFCcsn0_GPIO2a2	GPIO2A_IOMUX[5:4] = 2'b01
flash_rdn	O	IO_NANDrdn_SFCcsn1_GPIO2a3	GPIO2A_IOMUX[7:6] = 2'b01
flash_dqs	I/O	IO_NANDdqs_EMMCclkout_GPIO2a7	GPIO2A_IOMUX[15:14]= 2'b01
flash_rdy	I	IO_NANDrdy_EMMCcmd1_SFCclk_GPIO2a4	GPIO2A_IOMUX[9:8] = 2'b01
flash_csn0	O	IO_NANDcs0_GPIO2a6	GPIO2A_IOMUX[12]= 1'b1
flash_csn1	O	IO_NANDcs1_GPIO0c7	GPIO0C_IOMUX[14]= 1'b1
flash_csn2	O	IO_NANDcs2_EMMCcmd_GPIO1c6	GPIO1C_IOMUX[13:12]= 2'b01
flash_csn3	O	IO_NANDcs3_EMMCrstnout_GPIO1c7	GPIO1C_IOMUX[15:14]= 2'b01
flash_data0	I/O	IO_NANDd0_EMMCd0_SFCd0_GPIO1d0	GPIO1D_IOMUX[1:0] = 2'b01
flash_data1	I/O	IO_NANDd1_EMMCd1_SFCd1_GPIO1d1	GPIO1D_IOMUX[3:2] = 2'b01
flash_data2	I/O	IO_NANDd2_EMMCd2_SFCd2_GPIO1d2	GPIO1D_IOMUX[5:4] = 2'b01
flash_data3	I/O	IO_NANDd3_EMMCd3_SFCd3_GPIO1d3	GPIO1D_IOMUX[7:6] = 2'b01
flash_data4	I/O	IO_NANDd4_EMMCd4_SPI1rxdi1_GPIO1d4	GPIO1D_IOMUX[9:8] = 2'b01
flash_data5	I/O	IO_NANDd5_EMMCd5_SPI1txdi1_GPIO1d5	GPIO1D_IOMUX[11:10]= 2'b01
flash_data6	I/O	IO_NANDd6_EMMCd6_SPI1csn0_GPIO1d6	GPIO1D_IOMUX[13:12]= 2'b01
flash_data7	I/O	IO_NANDd7_EMMCd7_SPI1csn1_GPIO1d7	GPIO1D_IOMUX[15:14]= 2'b01

Notes: I=input, O=output, I/O=input/output, bidirectional

Furthermore, different IOs are selected and connected to different flash interface, which is shown as follows.

Table 12-3 NandC Interface Connection

Module Pin	Direction	Flash Interface			
		Asyn8x	Asyn16x	ONFI	Toggle
flash_csn <i>i</i> (i=0~7)	O	✓	✓	✓	✓
flash_wp	O	✓	✓	✓	✓
flash_ale	O	✓	✓	✓	✓
flash_cle	O	✓	✓	✓	✓
flash_wrn	O	✓	✓	✓	✓
flash_rdn	O	✓	✓	✓	✓
flash_data[7:0]	I/O	✓	✓	✓	✓
flash_data[15:8]	I/O	-	✓	-	-
flash_dqs	I/O	-	-	✓	✓
flash_rdy	I	✓	✓	✓	✓

12.6 Application Notes

12.6.1 Clock Description

There are two clock domains in the NandC. One is hclk, and the other is nclk.

- AHB slave bus uses the hclk to configure the NandC registers.
- AHB master bus uses the hclk to transmit the data between the external memory and internal sram.
- NandC use nclk to transmit the data between the nand flash and internal sram.
- All the flash interface timing configuration is relative to the nclk.

12.6.2 BCHST/BCHLOC/BCHDE/SPARE Application

1. BCHST

There are 16 BCHST-registers in NandC to store 32 codeword's BCH decode status(bchst) information. Every register stores 2 codeword's bchst information except BCHST0, which not only includes bchst information, but also includes one bit for *bchrdy*.

Let *bchst_cwd0*~*bchst_cwd31* be the bchst information for 32 codewords. In BCHST-registers, the latest codeword's bchst is stored into *bchst_cwd0*, and the former is shifted into *bchst_cwd1*. That is, *bchst_cwd0*→*bchst_cwd1*→.....→*bchst_cwd31*. Therefore, for example, if 32 codewords are decoded, then *bchst_cwd0* is the bch decode status for codeword31, and *bchst_cwd31* is the bch decode status for codeword0.

```

bchst_cwd0 =
{BCHST0[28],BCHST0[12:8],BCHST0[27],BCHST0[7:3],BCHST0[2:0]}

bchst_cwd1 =
{BCHST0[30],BCHST0[25:21],BCHST0[29],BCHST0[20:16],BCHST0[15:13]}

bchst_cwd2 =
{BCHST1[28],BCHST1[12:8],BCHST1[27],BCHST1[7:3],BCHST1[2:0]}

bchst_cwd3 =
{BCHST1[30],BCHST1[25:21],BCHST1[29],BCHST1[20:16],BCHST1[15:13]}

bchst_cwd4 =
{BCHST2[28],BCHST2[12:8] ,BCHST2[27],BCHST2[7:3],BCHST2[2:0]}

bchst_cwd5 =
{BCHST2[30],BCHST2[25:21],BCHST2[29],BCHST2[20:16],BCHST2[15:13]}

bchst_cwd6 =
{BCHST3[28],BCHST3[12:8] ,BCHST3[27],BCHST3[7:3],BCHST3[2:0]}

bchst_cwd7 =
{BCHST3[30],BCHST3[25:21],BCHST3[29],BCHST3[20:16],BCHST3[15:13]}

bchst_cwd8 =
{BCHST4[28],BCHST4[12:8] ,BCHST4[27],BCHST4[7:3],BCHST4[2:0]}

bchst_cwd9 =
{BCHST4[30],BCHST4[25:21],BCHST4[29],BCHST4[20:16],BCHST4[15:13]}

bchst_cwd10 =
{BCHST5[28],BCHST5[12:8] ,BCHST5[27],BCHST5[7:3],BCHST5[2:0]}

bchst_cwd11 =
{BCHST5[30],BCHST5[25:21],BCHST5[29],BCHST5[20:16],BCHST5[15:13]}

bchst_cwd12 =
{BCHST6[28],BCHST6[12:8] ,BCHST6[27],BCHST6[7:3],BCHST6[2:0]}

bchst_cwd13 =

```

```

{BCHST6[30],BCHST6[25:21],BCHST6[29],BCHST6[20:16],BCHST6[15:13]}
bchst_cwd14 =
{BCHST7[28],BCHST7[12:8] ,BCHST7[27],BCHST7[7:3],BCHST7[2:0]}
bchst_cwd15 =
{BCHST7[30],BCHST7[25:21],BCHST7[29],BCHST7[20:16],BCHST7[15:13]}
bchst_cwd16 =
{BCHST8[28],BCHST8[12:8],BCHST8[27],BCHST8[7:3],BCHST8[2:0]}
bchst_cwd17 =
{BCHST8[30],BCHST8[25:21],BCHST8[29],BCHST8[20:16],BCHST8[15:13]}
bchst_cwd18 =
{BCHST9[28],BCHST9[12:8],BCHST9[27],BCHST9[7:3],BCHST9[2:0]}
bchst_cwd19 =
{BCHST9[30],BCHST9[25:21],BCHST9[29],BCHST9[20:16],BCHST9[15:13]}
bchst_cwd20 =
{BCHST10[28],BCHST10[12:8] ,BCHST10[27],BCHST10[7:3],BCHST10[2:0]}
bchst_cwd21 =
{BCHST10[30],BCHST10[25:21],BCHST10[29],BCHST10[20:16],BCHST10[15
:13]}
bchst_cwd22 =
{BCHST11[28],BCHST11[12:8] ,BCHST11[27],BCHST11[7:3],BCHST11[2:0]}
bchst_cwd23 =
{BCHST11[30],BCHST11[25:21],BCHST11[29],BCHST11[20:16],BCHST11[15
:13]}
bchst_cwd24 =
{BCHST12[28],BCHST12[12:8] ,BCHST12[27],BCHST12[7:3],BCHST12[2:0]}
bchst_cwd25 =
{BCHST12[30],BCHST12[25:21],BCHST12[29],BCHST12[20:16],BCHST12[15
:13]}
bchst_cwd26 =
{BCHST13[28],BCHST13[12:8] ,BCHST13[27],BCHST13[7:3],BCHST13[2:0]}
bchst_cwd27 =
{BCHST13[30],BCHST13[25:21],BCHST13[29],BCHST13[20:16],BCHST13[15
:13]}
bchst_cwd28 =
{BCHST14[28],BCHST14[12:8] ,BCHST14[27],BCHST14[7:3],BCHST14[2:0]}
bchst_cwd29 =
{BCHST14[30],BCHST14[25:21],BCHST14[29],BCHST14[20:16],BCHST14[15
:13]}
bchst_cwd30 =
{BCHST15[28],BCHST15[12:8] ,BCHST15[27],BCHST15[7:3],BCHST15[2:0]}
bchst_cwd31 =
{BCHST15[30],BCHST15[25:21],BCHST15[29],BCHST15[20:16],BCHST15[15
:13]}

```

2. BCHLOC

There are 7 BCHLOC-registers in NandC to store 32 codeword's bch decode location(bchloc) information.

Let bchloc_cwd0~bchloc_cwd31 be the bchloc information for the 32 codeword. In BCHLOC registers, the latest codeword's bchloc is stored into bchloc_cwd0, and the former is shifted into bchloc_cwd1. That is,

bchloc_cwd0→bchloc_cwd1→.....→bchloc_cwd31. Therefore, for example, if 32 codeword are decoded, then bchloc_cwd0 is the bch decode status for codeword31, and bchloc_cwd31 is the bch decode status for codeword0.

```
bchloc_cwd0 = {BCHLOC6[0], BCHLOC0[4:0]}\n\nbchloc_cwd1 = {BCHLOC6[1], BCHLOC0[9:5]}\n\nbchloc_cwd2 = {BCHLOC6[2], BCHLOC0[14:10]}\n\nbchloc_cwd3 = {BCHLOC6[3], BCHLOC0[19:15]}\n\nbchloc_cwd4 = {BCHLOC6[4], BCHLOC0[24:20]}\n\nbchloc_cwd5 = {BCHLOC6[5], BCHLOC0[29:25]}\n\nbchloc_cwd6 = {BCHLOC6[6], BCHLOC1[4:0]}\n\nbchloc_cwd7 = {BCHLOC6[7], BCHLOC1[9:5]}\n\nbchloc_cwd8 = {BCHLOC6[8], BCHLOC1[14:10]}\n\nbchloc_cwd9 = {BCHLOC6[9], BCHLOC1[19:15]}\n\nbchloc_cwd10 = {BCHLOC6[10], BCHLOC1[24:20]}\n\nbchloc_cwd11 = {BCHLOC6[11], BCHLOC1[29:25]}\n\nbchloc_cwd12 = {BCHLOC6[12], BCHLOC2[4:0]}\n\nbchloc_cwd13 = {BCHLOC6[13], BCHLOC2[9:5]}\n\nbchloc_cwd14 = {BCHLOC6[14], BCHLOC2[14:10]}\n\nbchloc_cwd15 = {BCHLOC6[15], BCHLOC2[19:15]}\n\nbchloc_cwd16 = {BCHLOC6[16], BCHLOC2[24:20]}\n\nbchloc_cwd17 = {BCHLOC6[17], BCHLOC2[29:25]}\n\nbchloc_cwd18 = {BCHLOC6[18], BCHLOC3[4:0]}\n\nbchloc_cwd19 = {BCHLOC6[19], BCHLOC3[9:5]}\n\nbchloc_cwd20 = {BCHLOC6[20], BCHLOC3[14:10]}\n\nbchloc_cwd21 = {BCHLOC6[21], BCHLOC3[19:15]}\n\nbchloc_cwd22 = {BCHLOC6[22], BCHLOC3[24:20]}\n\nbchloc_cwd23 = {BCHLOC6[23], BCHLOC3[29:25]}\n\nbchloc_cwd24 = {BCHLOC6[24], BCHLOC4[4:0]}\n\nbchloc_cwd25 = {BCHLOC6[25], BCHLOC4[9:5]}\n\nbchloc_cwd26 = {BCHLOC6[26], BCHLOC4[14:10]}\n\nbchloc_cwd27 = {BCHLOC6[27], BCHLOC4[19:15]}\n\nbchloc_cwd28 = {BCHLOC6[28], BCHLOC4[24:20]}\n\nbchloc_cwd29 = {BCHLOC6[29], BCHLOC4[29:25]}\n\nbchloc_cwd30 = {BCHLOC6[30], BCHLOC5[4:0]}\n\nbchloc_cwd31 = {BCHLOC6[31], BCHLOC5[9:5]}
```

3. BCHDE

BCHDE includes two register-groups, BCHDE0 and BCHDE1. Each group has 60 registers: BCHDE0_0~BCHDE0_59 and BCHDE1_0~BCHDE1_59.

BCHDE0_n(n=0~59) is the decode information of the nth error bit for codeword in sram0, and BCHDE1_n(n=0~59) is the decode information of the nth error bit for codeword in sram1.

The needed number of BCHDE registers is determined by bchmode. That is:

- a. When 16bitBCH selected, BCHDEM_0 ~ BCHDEM_15 are available
- b. When 24bitBCH selected, BCHDEM_0 ~ BCHDEM_23 are available
- c. When 40bitBCH selected, BCHDEM_0 ~ BCHDEM_39 are available
- d. When 60bitBCH selected, BCHDEM_0 ~ BCHDEM_59 are available

4. SPARE

SPARE includes two register-groups, SPARE0 and SPARE1. Each group has 28 registers: SPARE0_0~SPARE0_27 and SPARE1_0~SPARE1_27.

When in bch encoding, SPARE0_0 stores system information for codeword in sram0, SPARE0_n(n=1~27) stores encode information for codeword in sram0; SPARE1_0 stores system information for codeword in sram1, SPARE1_n(n=1~27) stores encode information for codeword in sram1.

When in bch decoding, SPARE0_n(n=0~27) stores the spare data read from flash for codeword in sram0; SPARE1_n(n=0~27) stores the spare data read from flash for codeword in sram1.

The needed number of BCHDE registers is determined by bchmode. That is:

- a. When 16bitBCH selected, spare data=28bytes, SPAREm_0~SPAREm_7 are available
- b. When 24bitBCH selected, spare data=42bytes, SPAREm_0~SPAREm_10 and SPAREm_11[15:0] are available
- c. When 40bitBCH selected, spare data=70bytes, SPAREm_0~SPAREm_17 and SPAREm_18[15:0] are available
- d. When 60bitBCH selected, spare data=105bytes, SPAREm_0~SPAREm_26 and SPAREm_27[7:0] are available

12.6.3 Bus Mode Application

MTRANS_CFG[2] determines whether the data load/store between internal memory and external memory is through slave interface or master interface.

1. Slave Mode

When MTRANS_CFG[2]=0, slave is selected. i. e. , flash data load/store between internal memory and external memory is through slave interface by cpu or external DMA.

In this mode, software should store page data into internal memory and spare data into SPARE registers before starting flash program operation; and should load page data from internal memory and spare data from SPARE registers after finishing flash read operation.

In this mode, MTRANS_CFG, MTRANS_SADDR0 and MTRANS_SADDR1 are unused. The transfer codeword number is determined by FLCTL[6:5], and the maximum number is 2.

The judgment condition for finishing data transfer is FLCTL[20]. When FLCTL[20] is high, it means that data transfer is finished.

2. Master Mode

When MTRANS_CFG[2]=1, master is selected. i. e. , flash data load/store between internal memory and external memory is through master interface.

In this mode, software should initialize page data and spare data into external memory, and set their addresses in MTRANS_SADDR0 and MTRANS_SADDR1 respectively before starting flash program operation. Similarly, software should configure MTRANS_SADDR0 and MTRANS_SADDR1 respectively before starting flash read operation and could read data from addresses in MTRANS_SADDR0 and MTRANS_SADDR1 after NandC transfer finish.

In this mode, MTRANS_CFG, MTRANS_SADDR0 and MTRANS_SADDR1 are used. The transfer codeword number is determined by FLCTL[26:22], and the maximum number is 16.

The judgment condition for finishing data transfer is FLCTL[20]. When FLCTL[20] is high, it means that data transmission is finished.

When MTRANS_CFG[2]=1, page data and spare data are stored in the continuous space of external memory respectively.

For page data, source address is named Saddr0, specified in MTRANS_SADDR0. The space can be divided into many continuous units, and the unit size(named PUnit) is 1024 bytes or 512 bytes determined by FLCTL[21] and FLCTL[11]:

- when FLCTL[11]=0, PUnit is always equal to 1024 bytes
- when FLCTL[11]=1 and FLCTL[21]=0, PUnit is equal to 1024 bytes
- when FLCTL[11]=1 and FLCTL[21]=1, PUnit is equal to 512 bytes

For spare data, source address is named Saddr1, specified in MTRANS_SADDR1. The space can be divided into many continuous units, and the unit size(named SUnit) is 64 bytes or 128 bytes determined by BCHCTL[18], FLCTL[11] and FLCTL[21]:

- When FLCTL[11]=0 and BCHCTL[18]=0, SUnit is equal to 64 bytes
- When FLCTL[11]=0 and BCHCTL[18]=1, SUnit is equal to 128 bytes
- When FLCTL[11]=1 and FLCTL[21]=0, SUnit is equal to 128 bytes
- When FLCTL[11]=1 and FLCTL[21]=1, SUnit is equal to 64 bytes

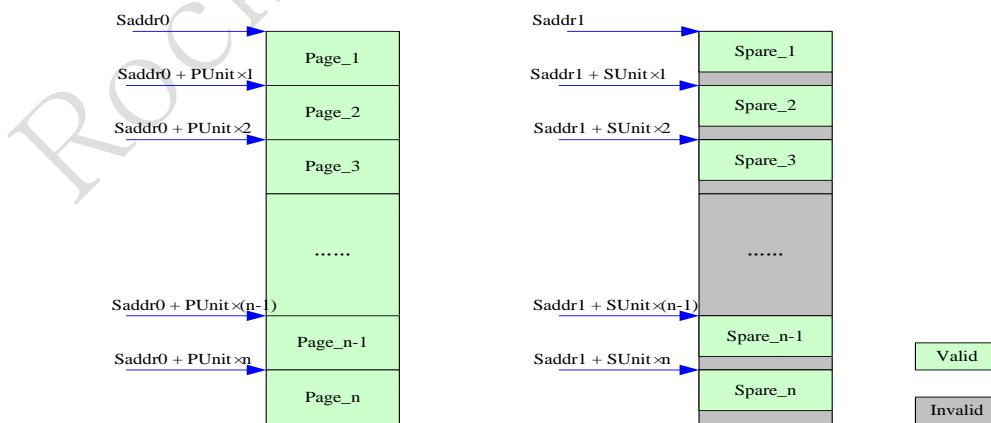


Fig. 12-2 NandC Address Assignment

The detailed format for page data and spare data in every unit is shown in following figures.

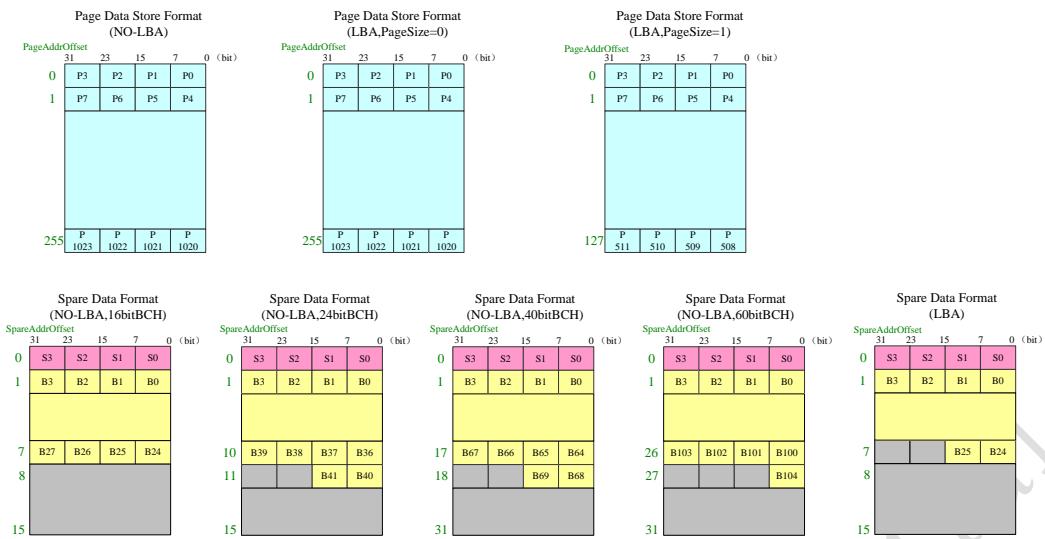


Fig. 12-3 NandC DataFormat

12.6.4 BchPage Application

BCHCTL[16] determines whether codeword size for page data is 1024 bytes or 512 bytes when FLCTL[11] is 0.

1. 1024bytes

When BCHCTL[16]=0, BchPage=0, hardware needs to write 1024 bytes page data and spare data into flash or read 1024 bytes page data and spare data from flash. All the 1024 bytes page data and spare data are encoded when writing or decoded when reading.

2. 512bytes

When BCHCTL[16]=1, BchPage=1, hardware needs to write 512 bytes page data and spare data into flash or read 512 bytes page data and spare data from flash.

In this mode, the page data unit size for BCH encoder and BCH decoder still is 1024byte. So to support BCH encoder and decoder, software should configure page data as follows: 1th~512th bytes are invalid data which must be stuffed with 0xff, 513th~1024th bytes are valid page data.

However, Randomizer function is not supported under this condition.

12.6.5 PageSize/SpareSize Application

FLCTL[21] determines whether the codeword size is 1024 bytes or 512 bytes when FLCTL[11] is 1.

1. Big Page

When FLCTL[11]=0(LbaEn=0), the flash to be operated is Raw NAND Flash. Every codeword size is 1024 bytes and FLCTL[21] should always be set to 0, and the PageStep in external memory is 1024 bytes if bus mode is master mode.

At this mode, the spare size and SpareStep in external memory are determined by BCH Mode as follows:

BCH Mode=16bitBCH: spare size=(28+4)bytes , SpareStep=64bytes

BCH Mode=24bitBCH: spare size=(42+4)bytes , SpareStep=64bytes

BCH Mode=40bitBCH: spare size=(70+4)bytes , SpareStep=128bytes

BCH Mode=60bitBCH: spare size=(105+4)bytes, SpareStep=128bytes

2. Small Page

When FLCTL[11]=1, LbaEn=1, the flash to be operated is Managed NAND Flash. Every codeword size could be 1024 bytes or 512 bytes according to FLCTL[21]. If FLCTL[21]=0, codeword size is 1024 bytes, PageStep in external memory is 1024 bytes, and SpareStep is 128bytes. If FLCTL[21]=1, codeword size is 512 bytes, PageStep in external memory is 512 bytes, and SpareStep is 64 bytes.

At this mode, the spare size is configured in FLCTL[18:12], and the max available number is 109.

In the summary, the total data size in every codeword for flash or for software including page data and spare data, is determined by BCHCTL[16], FLCTL[11], FLCTL[21], BCHCTL[4], BCHCTL[18]. Their relationship is shown as follows.

Table 12-4 NandC Page/Spare size for flash

page/spare size for software	page size /codeword	spare size /codeword
FLCTL[11]=0	16bitECC	1024 byte (4+28)byte
	24bitECC	1024 byte (4+42)byte
	40bitECC	1024 byte (4+70)byte
	60bitECC	1024 byte (4+105)byte
	FLCTL[21]=0	1024 byte FLCTL[18:12]
FLCTL[11]=1	FLCTL[21]=1	512 byte FLCTL[18:12]

Notes: that "page/spare size for flash" means that hardware should transfer these numbers of bytes in every codeword to or from flash.

12.6.6 Randomizer Application

RANDMZ_CFG[31] determines whether randomizer is enable or not. When RANDMZ_CFG[31] equals to 1, randomizer is active. Data should be scrambled before written into flash, and descrambled after read from flash.

RANDMZ_CFG[30:29] determines the randomizer polynomial.

When RANDMZ_CFG[30:29]=2'b00, Samsung randomizer, Polynomial = $1+x+x^15$. RANDMZ_CFG[14:0] is the seed for randomizer.

When RANDMZ_CFG[30:29]=2'b01, TOSHIBA randomizer. RANDMZ_CFG[19:0] is the seed agitation register for randomizer. RANDMZ_CFG[23:20] is the basic seed start point for every page. RANDMZ_CFG[28:24] is the basic seed rotation bits for every 16 page.

When RANDMZ_CFG[30:29]=2'b10, Samsung randomizer, Polynomial = $1+x^{14}+x^{15}$. RANDMZ_CFG[14:0] is the seed for randomizer.

The data unit for randomizer is one codeword(data+spare).

However, Randomizer is just available for data transfer by internal DMA mode, but not by for bypass mode. Furthermore, it should not be enable if BCHCTL[16]=0 (BchPage=512bytes).

12.6.7 DLL Application

When Toggle Flash or ONFI Synchronous Flash interface is active, DLL should be used to adjust DQS input with DQ when reading flash.

There are 2 registers for DLL configuration(DLL_CFG_REG0 and DLL_CFG_REG1), and 1 register for DLL status(DLL_OBS_REG0).

The usage guide is as follows:

If bypass mode is used, you should set *dll_bypass* in DLL_CFG_REG1[1] to 1, and set *dll_dqs_dly_bypass* in DLL_CFG_REG0[23:16] to determine the dll element number needed. And then set *dll_start* in DLL_CFG_REG1[0] to 1 to start the DLL.

If auto adjusting is used, you should set *dll_bypass* in DLL_CFG_REG1[1] to 0, and set the *dll_start_point* in DLL_CFG_REG0[7:0] and *dll_incr* in DLL_CFG_REG1[11:4]. You also should set the adjusting mode *dll_qtren* in DLL_CFG_REG1[3:2] to compute the dll element number needed. If *dll_qtren*=2'b00, the dll element number is determined by *dll_dqs_dly* in DLL_CFG_REG0[15:8]; otherwise, it is 1/4 or 1/8 of the total number of dll elements used for *dll_qtren*=2'b01 or *dll_qtren*=2'b10 separately. The last step is to set *dll_start* in DLL_CFG_REG1[0] to 1 to start the DLL.

If you want to monitor the dll working status, you could read DLL_OBS_REG0. If DLL_OBS_REG0[0]=0, it means that DLL is not locked, and still in detecting status. Otherwise, it means that DLL is locked, and *dll_lock_value* in DLL_OBS_REG0[8:1] is the total number of dll elements used, *dll_dqs_delay_value* in DLL_OBS_REG0[16:9] is the total number of DQS delay used.

12.6.8 NandC Interrupt Application

NandC has 1 interrupt output signal and 4 interrupt sources: dma finish interrupt source, flash ready interrupt source, bch error interrupt source, bchfail interrupt source. When one or more of these interrupt source are enabled, NandC interrupt is asserted if one or more interrupt source is high. Software can determine the interrupt source by reading INTST and clear interrupt by writing corresponding bit in INTCLR.

12.6.9 LLP Application

LLP is used in NandC to store and execute instruction groups configured in external memory by software. When LLPCTL[0]=1, LLP is active, NandC will load instruction groups stored in {LLPCTL[31:6], 6'h0} and execute them. Next instruction groups should not be loaded until current instruction execution finished.

1. LLP Structure

The structure of LLP is shown as follows:

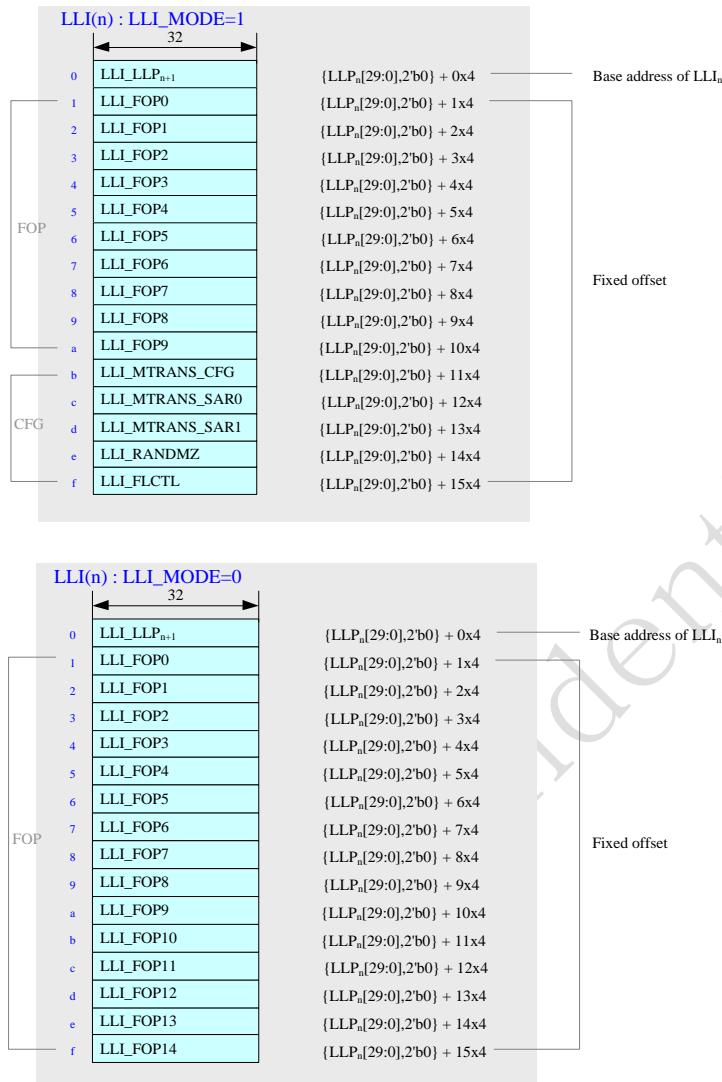


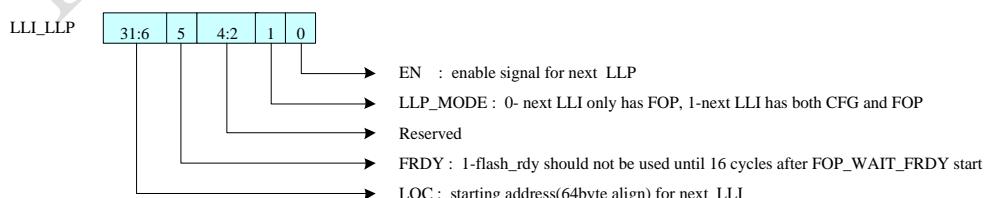
Fig. 12-4 NandC LLP Data Format

LLI_MODE is determined by LLPCTL[1]. If current operation is flash program or flash read, then LLI_MODE=1 is need; otherwise, LLI_MODE=0 is workable.

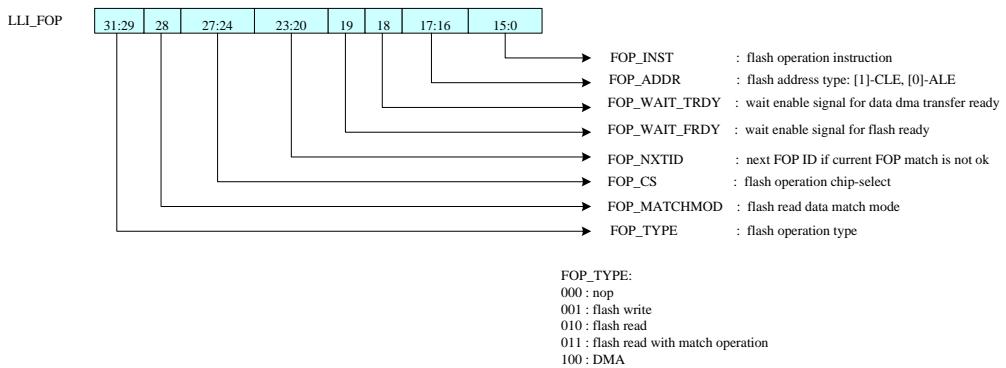
In addition, you could do more than one flash operation in one LLP group, but you should not separate one flash operation into two LLI groups.

2. LLI Format

a. LLI_LLPO_{n+1} stores the address for next LLI group data



b. LLI_FOP0~LLI_FOP14 store the flash operation instruction



When FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. It is matched when “RDATA|PATTERN=PATTERN” with FOP_MATCHMOD=0, or when “RDATA&PATTERN=PATTERN” with FOP_MATCHMOD=1.

- c. LLI_MTRANS_CFG/LLI_MTRANS_SADDR0/LLI_MTRANS_SADDR1/LLI_RANDMZ/ LLI_FLCTL store the configuration for MTRANS_CFG/MTRANS_SADDR0/MTRANS_SADDR1/RANDMZ/FLCTL.

3. LLP Working Mode

There are two working modes for LLP:

- Normal mode: LLPCTL[0] is kept to 1 until all LLP loading and executing finished. Software can monitor the progress by LLPSTAT[31:6], LLPSTAT[0].
- Pause mode: LLPCTL[0] is changed from 1 to 0 during LLP loading or LLP executing. NandC should not stop working until current LLP executing finished. Software can monitor the progress by LLPSTAT[31:6], LLPSTAT[0].

Chapter 13 Mobile Storage Host Controller

13.1 Overview

The Mobile Storage Host Controller is designed to support Secure Digital memory (SD-max version 3.01) with 1 bits or 4 bits data width, Multimedia Card(MMC-max version 4.51) with 1 bits or 4 bits or 8 bits data width.

The Host Controller is instantiated for SDMMC, SDIO EMMC in RKAUDI. The interface difference between these instances is shown in "Interface Description".

The Host Controller supports following features:

- Bus Interface Features:
 - Support AMBA AHB interface for master and slave
 - In addition to AMBA slave interface, Support optional external DMA controllers for data transfers
 - Support combined single FIFO for both transmit and receive operations
 - Support FIFO size of 256x32
 - Support FIFO over-run and under-run prevention by stopping card clock
- Card Interface Features:
 - Support Secure Digital memory protocol commands
 - Support Secure Digital I/O protocol commands
 - Support Multimedia Card protocol commands
 - Support Command Completion Signal and interrupts to host
 - Support CRC generation and error detection
 - Support programmable baud rate
 - Support power management and power switch
 - Support card detection
 - Support write protection
 - Support hardware reset
 - Support SDIO interrupts in 1-bit and 4-bit modes
 - Support 4-bit mode in SDIO3.0
 - Support SDIO suspend and resume operation
 - Support SDIO read wait
 - Support block size of 1 to 65,535 bytes
 - Support 1-bit, 4-bit and 8-bit SDR modes
 - Support 4-bit DDR,8-bit DDR, as defined by SD3.0 and MMC4.41
 - Support boot in 1-bit, 4-bit and 8-bit SDR modes
 - Support Packed Commands, CMD21, CMD49
- Clock Interface Features:
 - Support 0/90/180/270-degree phase shift operation for sample clock (cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock(cclk_in) respectively
 - Support phase tuning using delay line for sample clock(cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock (cclk_in) respectively. The max number of delay element number is 256.

13.2 Block Diagram

The Host Controller consists of the following main functional blocks.

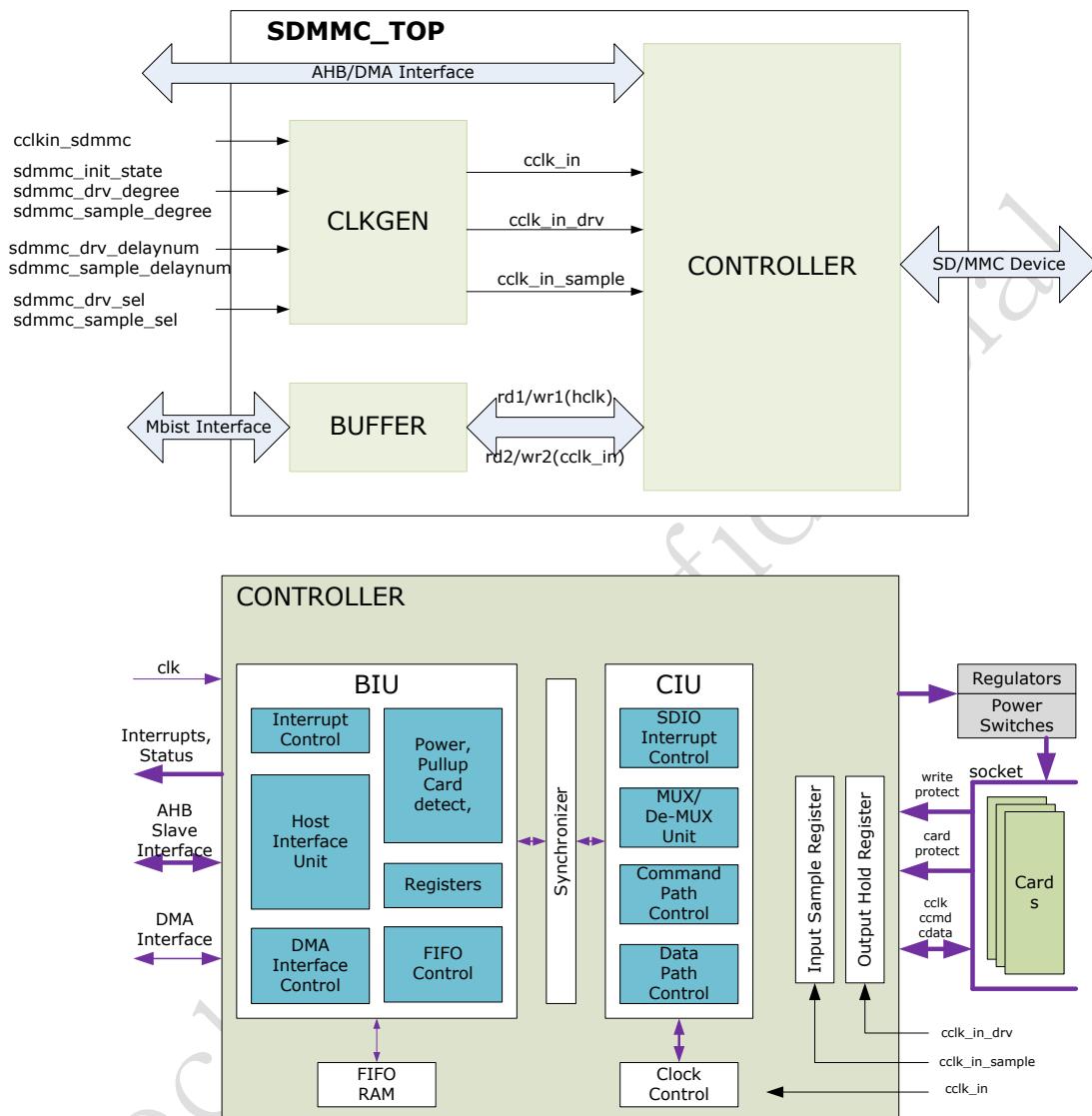


Fig. 13-1 Host Controller Block Diagram

- Clock Generate Unit(CLKGEN): generates card interface clock `cclk_in`/`cclk_sample`/`cclk_drv` based on `cclkin` and configuration information.
- Asynchronous dual-port memory(BUFFER): Uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, and the second port is connected to the card clock.
- Bus Interface Unit (BIU): Provides AMBA AHB interfaces for register and data read/writes.
- Card Interface Unit (CIU): Takes care of the SD/MMC protocols and provides clock management.

13.3 Function Description

13.3.1 Bus Interface Unit

The Bus Interface Unit provides the following functions:

- Host interface
- Interrupt control
- Register access
- External FIFO access
- Power control and card detection

1. Host Interface Unit

The Host Interface Unit is an AHB slave interface, which provides the interface between the SD/MMC card and the host bus. You can configure the host interface as either an AHB.

2. Register Unit

The register unit is part of the bus interface unit; it provides read and write access to the registers.

All registers reside in the Bus Interface Unit clock domain. When a command is sent to a card by setting the start_bit, which is bit[31] of the CMD register, all relevant registers needed for the CIU operation are transferred to the CIU block. During this time, the registers that are transferred from the BIU to the CIU should not be written. The software should wait for the hardware to clear the start bit before writing to these registers again. The register unit has a hardware locking feature to prevent illegal writes to registers. The lock is necessary in order to avoid metastability violations, both because the host and card clock domains are different and to prevent illegal software operations.

Once a command start is issued by setting the start_bit of the CMD register, the following registers cannot be reprogrammed until the command is accepted by the card interface unit:

- CMD – Command
- CMDARG – Command Argument
- BYTCNT – Byte Count
- BLKSIZ – Block Size
- CLKDIV – Clock Divider
- CLKENA – Clock Enable
- CLKSRC – Clock Source
- TMOUT – Timeout
- CTYPE – Card Type

The hardware resets the start_bit once the CIU accepts the command. If a host write to any of these registers is attempted during this locked time, then the write is ignored and the hardware lock error bit is set in the raw interrupt status register. Additionally, if the interrupt is enabled and not masked for a hardware lock error, then an interrupt is sent to the host.

When the Card Interface Unit is in an idle state, it typically takes the following

number of clocks for the command handshake, where clk is the BIU clock and cclk_in is the CIU clock:

$$3(\text{clk}) + 3(\text{cclk_in})$$

Once a command is accepted, you can send another command to the CIU-which has a one-deep command queue-under the following conditions:

- If the previous command was not a data transfer command, the new command is sent to the SD/MMC card once the previous command completes.
- If the previous command is a data transfer command and if wait_prvdata_complete (bit[13]) of the Command register is set for the new command, the new command is sent to the SD/MMC card only when the data transfer completes.
- If the wait_prvdata_complete is 0, then the new command is sent to the SD/MMC card as soon as the previous command is sent. Typically, you should use this only to stop or abort a previous data transfer or query the card status in the middle of a data transfer.

3. Interrupt Controller Unit

The interrupt controller unit generates an interrupt that depends on the controller raw interrupt status, the interrupt-mask register, and the global interrupt-enable register bit. Once an interrupt condition is detected, it sets the corresponding interrupt bit in the raw interrupt status register. The raw interrupt status bit stays on until the software clears the bit by writing a 1 to the interrupt bit; a 0 leaves the bit untouched.

The interrupt port, int, is an active-high, level-sensitive interrupt. The interrupt port is active only when any bit in the raw interrupt status register is active, the corresponding interrupt mask bit is 1, and the global interrupt enable bit is 1. The interrupt port is registered in order to avoid any combinational glitches.

The int_enable is reset to 0 on power-on, and the interrupt mask bits are set to 32'h0, which masks all the interrupts.

Notes:

Before enabling the interrupt, it is always recommended that you write 32'hffff_ffff to the raw interrupt status register in order to clear any pending unserviced interrupts. When clearing interrupts during normal operation, ensure that you clear only the interrupt bits that you serviced.

The SDIO Interrupts, Receive FIFO Data Request (RXDR), and Transmit FIFO Data Request (TXDR) are set by level-sensitive interrupt sources. Therefore, the interrupt source should be first cleared before you can clear the interrupt bit of the Raw Interrupt register. For example, on seeing the Receive FIFO Data Request (RXDR) interrupt, the FIFO should be emptied so that the "FIFO count greater than the RX-Watermark" condition, which triggers the interrupt, becomes inactive. The rest of the interrupts are triggered by a single clock-pulse-width source.

Table 13-1 Bits in Interrupt Status Register

Bits	Interrupt	Description
24	sdio_interrupt	Interrupt from SDIO card. In MMC-Ver3.3-only mode, these bits are always 0
16	Card no-busy	If card exit busy status, the interrupt

		happened
15	End Bit Error (read) /Write no CRC (EBE)	Error in end-bit during read operation, or no data CRC or negative CRC received during write operation. <i>Notes: For MMC CMD19, there may be no CRC status returned by the card. Hence, EBE is set for CMD19. The application should not treat this as an error.</i>
14	Auto Command Done (ACD)	Stop/abort commands automatically sent by card unit and not initiated by host; similar to Command Done (CD) interrupt.
13	Start Bit Error (SBE)	Error in data start bit when data is read from a card. In 4-bit mode, if all data bits do not have start bit, then this error is set.
12	Hardware Locked write Error (HLE)	During hardware-lock period, write attempted to one of locked registers.
11	FIFO Underrun/Overrun Error (FRUN)	Host tried to push data when FIFO was full, or host tried to read data when FIFO was empty. Typically this should not happen, except due to error in software. Card unit never pushes data into FIFO when FIFO is full, and pop data when FIFO is empty.
10	Data Starvation by Host Timeout (HTO)	To avoid data loss, card clock out (cclk_out) is stopped if FIFO is empty when writing to card, or FIFO is full when reading from card. Whenever card clock is stopped to avoid data loss, data-starvation timeout counter is started with data-timeout value. This interrupt is set if host does not fill data into FIFO during write to card, or does not read from FIFO during read from card before timeout period. Even after timeout, card clock stays in stopped state, with CIU state machines waiting. It is responsibility of host to push or pop data into FIFO upon interrupt, which automatically restarts cclk_out and card state machines. Even if host wants to send stop/abort command, it still needs to ensure it has to push or pop FIFO so that clock starts in order for stop/abort command to send on cmd signal along with data that is sent or received on data line.
9	Data Read Timeout (DRTO)	Data timeout occurred. Data Transfer Over (DTO) also set if data timeout occurs.
8	Response Timeout (RTO)	Response timeout occurred. Command Done (CD) also set if response timeout occurs. If command involves data transfer and when response times out, no data transfer is attempted by Host Controller.
7	Data CRC Error (DCRC)	Received Data CRC does not match with locally-generated CRC in CIU.
6	Response CRC Error (RCRC)	Response CRC does not match with locally-generated CRC in CIU.
5	Receive FIFO Data Request (RXDR)	Interrupt set during read operation from card when FIFO level is greater than

		Receive-Threshold level.
4	Transmit FIFO Data Request (TXDR)	Interrupt set during write operation to card when FIFO level reaches less than or equal to Transmit-Threshold level.
3	Data Transfer Over (DTO)	Data transfer completed, even if there is Start Bit Error or CRC error. This bit is also set when "read data-timeout" occurs. <i>Notes: DTO bit is set at the end of the last data block, even if the device asserts MMC busy after the last data block.</i>
2	Command Done(CD)	Command sent to card and got response from card, even if Response Error or CRC error occurs. Also set when response timeout occurs
1	Response Error (RE)	Error in received response set if one of following occurs: <ul style="list-style-type: none">● Transmission bit != 0● Command index mismatch● End-bit != 1
0	Card-Detect (CDT)	When card inserted or removed, this interrupt occurs. Software should read card-detect register (CDETECT, 0x50) to determine current card status.

4. FIFO Controller Unit

The FIFO controller interfaces the external FIFO to the host interface and the card controller unit. When FIFO overrun and under-run conditions occur, the card clock stops in order to avoid data loss.

The FIFO uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, clk, and the second port is connected to the card clock, cclk_in.

Notes: The FIFO controller does not support simultaneous read/write access from the same port. For debugging purposes, the software may try to write into the FIFO and read back the data; results are indeterminate, since the design does not support read/write access from the same port.

5. Power Control and Card Detection Unit

The register unit has registers that control the power. Power to each card can be selectively turned on or off.

The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value.

On power-on, the controller should read in the card_detect port and store the value in the memory. Upon receiving a card-detect interrupt, it should again read the card_detect port and XOR with the previous card-detect status to find out which card has interrupted. If more than one card is simultaneously removed or inserted, there is only one card-detect interrupt; the XOR value indicates which cards have been disturbed. The memory should be updated with the new card-detect value.

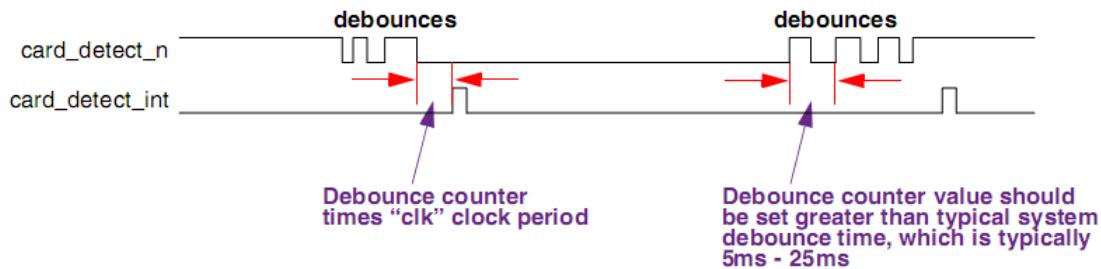


Fig. 13-2 SD/MMC Card-Detect Signal

6. DMA Interface Unit

DMA signals interface the Host Controller to an external DMA controller to reduce the software overhead during FIFO data transfers. The DMA request/acknowledge handshake is used for only data transfers. The DMA interface provides a connection to the DMA Controller.

On seeing the DMA request, the DMA controller initiates accesses through the host interface to read or write into the data FIFO. The Host Controller has FIFO transmit/receive watermark registers that you can set, depending on system latency. The DMA interface asserts the request in the following cases:

- Read from a card when the data FIFO word count exceeds the Rx-Watermark level
- Write to a card when the FIFO word count is less than or equal to the Tx-Watermark level

When the DMA interface is enabled, you can use normal host read/write to access the data FIFO.

13.3.2 Card Interface Unit

The Card Interface Unit (CIU) interfaces with the Bus Interface Unit (BIU) and the devices. The host writes command parameters to the BIU control registers, and these parameters are then passed to the CIU. Depending on control register values, the CIU generates SD/MMC command and data traffic on a selected card bus according to SD/MMC protocol. The Host Controller accordingly controls the command and data path.

The following software restrictions should be met for proper CIU operation:

- Only one data transfer command can be issued at a time.
- During an open-ended card write operation, if the card clock is stopped because the FIFO is empty, the software must first fill the data into the FIFO and start the card clock. It can then issue only a stop/abort command to the card.
- When issuing card reset commands (CMD0, CMD15 or CMD52_reset) while a card data transfer is in progress, the software must set the `stop_abort_cmd` bit in the Command register so that the Host Controller can stop the data transfer after issuing the card reset command.
- When the data end bit error is set in the RINTSTS register, the Host Controller does not guarantee SDIO interrupts. The software should ignore the SDIO interrupts and issue the stop/abort command to the card, so that the card stops sending the read data.

- If the card clock is stopped because the FIFO is full during a card read, the software should read at least two FIFO locations to start the card clock.

The CIU block consists of the following primary functional blocks:

- Command path
- Data path
- SDIO interrupt control
- Clock control
- Mux/demux unit

1. Command Path

The command path performs the following functions:

- Loads clock parameters
- Loads card command parameters
- Sends commands to card bus (ccmd_out line)
- Receives responses from card bus (ccmd_in line)
- Sends responses to BIU
- Drives the P-bit on command line

A new command is issued to the Host Controller by programming the BIU registers and setting the start_cmd bit in the Command register. The BIU asserts start_cmd, which indicates that a new command is issued to the SD/MMC device. The command path loads this new command (command, command argument, timeout) and sends acknowledge to the BIU by asserting cmd_taken.

Once the new command is loaded, the command path state machine sends a command to the device bus-including the internally generated CRC7-and receives a response, if any. The state machine then sends the received response and signals to the BIU that the command is done, and then waits for eight clocks before loading a new command.

Load Command Parameters

One of the following commands or responses is loaded in the command path:

- New command from BIU – When start_cmd is asserted, then the start_cmd bit is set in the Command register.
- Internally-generated auto-stop command – When the data path ends, the stop command request is loaded.
- IRQ response with RCA 0x000 – When the command path is waiting for an IRQ response from the MMC card and a “send irq response” request is signaled by the BIU, then the send_irq_response bit is set in the control register.

Loading a new command from the BIU in the command path depends on the following Command register bit settings:

- update_clock_registers_only – If this bit is set in the Command register, the command path updates only the clock enable, clock divider, and clock source registers. If this bit is not set, the command path loads the command,

command argument, and timeout registers; it then starts processing the new command.

- `wait_prvdata_complete` – If this bit is set, the command path loads the new command under one of the following conditions:
 - Immediately, if the data path is free (that is, there is no data transfer in progress), or if an open-ended data transfer is in progress (`byte_count` = 0).
 - After completion of the current data transfer, if a predefined data transfer is in progress.

Send Command and Receive Response

Once a new command is loaded in the command path, `update_clock_registers_only` bit is unset – the command path state machine sends out a command on the device bus; the command path state machine is illustrated in following figure.

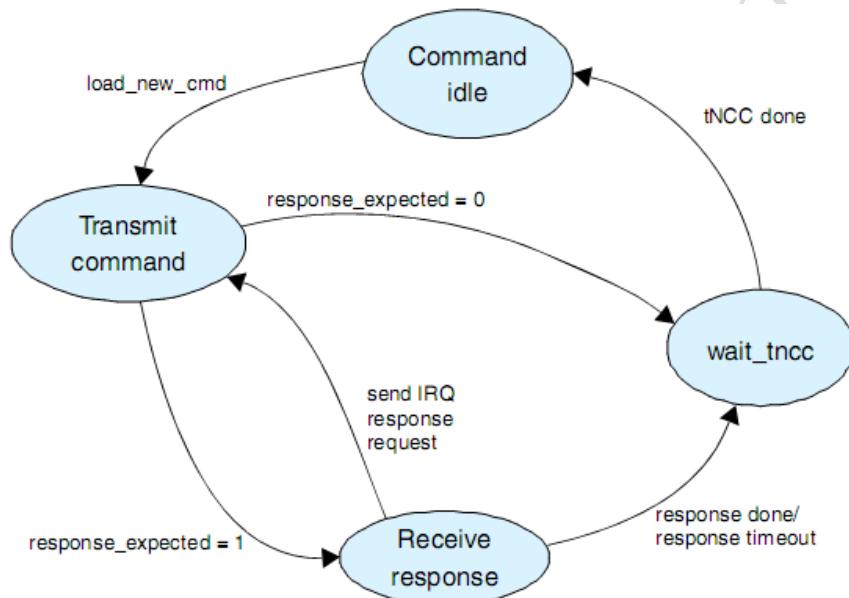


Fig. 13-3 Host Controller Command Path State Machine

The command path state machine performs the following functions, according to Command register bit values:

- `send_initialization` – Initialization sequence of 80 clocks is sent before sending the command.
- `response_expected` – Response is expected for the command. After the command is sent out, the command path state machine receives a 48-bit or 136-bit response and sends it to the BIU. If the start bit of the card response is not received within the number of clocks programmed in the timeout register, then the response timeout and command done bit is set in the Raw Interrupt Status register as a signal to the BIU. If the response-expected bit is not set, the command path sends out a command and signals a response done to the BIU; that is, the command done bit is set in the Raw Interrupt Status register.
- `response_length` – If this bit is set, a 136-bit response is received; if it is not

- set, a 48-bit response is received.
- **check_response_crc** – If this bit is set, the command path compares CRC7 received in the response with the internally-generated CRC7. If the two do not match, the response CRC error is signaled to the BIU; that is, the response CRC error bit is set in the Raw Interrupt Status register.

Send Response to BIU

If the response_expected bit is set in the Command register, the received response is sent to the BIU. The Response0 register is updated for a short response, and the Response3, Response2, Response1, and Response0 registers are updated on a long response, after which the Command Done bit is set. If the response is for an auto_stop command sent by the CIU, the response is saved in the Response1 register, after which the Auto Command Done bit is set.

Additionally, the command path checks for the following:

- Transmission bit = 0
- Command index matches command index of the sent command
- End bit = 1 in received card response

The command index is not checked for a 136-bit response or if the check_response_crc bit is unset. For a 136-bit response and reserved CRC 48-bit responses, the command index is reserved—that is, 111111.

Polling Command Completion Signal

The device generates the Command Completion Signal in order to notify the host controller of the normal command completion or command termination.

Command Completion Signal Detection and Interrupt to Host Processor

If the ccs_expected bit is set in the Command register, the Command Completion Signal (CCS) from the device is indicated by setting the Data Transfer Over (DTO) bit in the RINTSTS register. The Host Controller generates a Data Transfer Over (DTO) interrupt if this interrupt is not masked.

Command Completion Signal Timeout

If the command expects a CCS from the device—if the ccs_expected bit is set in the Command register—the command state machine waits for the CCS and remains in a wait_CCS state. If the device fails to send out the CCS, the host software should implement a timeout mechanism to free the command and data path. The host controller does not implement a hardware timer; it is the responsibility of the host software to maintain a software timer.

In the event of a CCS timeout, the host should issue a CCSD by setting the send_ccsd bit in the CTRL register. The host controller command state machine sends the CCSD to the device and exits to an idle state. After sending the CCSD, the host should also send a CMD12 to the device in order to abort the outstanding command.

Send Command Completion Signal Disable

If the send_ccsd bit is set in the CTRL register, the host sends a Command Completion Signal Disable (CCSD) pattern on the CMD line. The host can send the CCSD while waiting for the CCS or after a CCS timeout happens.

After sending the CCSD pattern, the host sets the Command Done (CD) bit in RINTSTS and also generates an interrupt to the host if the Command Done interrupt is not masked.

2. Data Path

The data path block pops the data FIFO and transmits data on cdata_out during a write data transfer, or it receives data on cdata_in and pushes it into the FIFO during a read data transfer. The data path loads new data parameters—that is, data expected, read/write data transfer, stream/block transfer, block size, byte count, card type, timeout registers—whenever a data transfer command is not in progress.

If the data_expected bit is set in the Command register, the new command is a data transfer command and the data path starts one of the following:

- Transmit data if the read/write bit = 1
- Data receive if read/write bit = 0

Data Transmit

The data transmit state machine, illustrated in following figure, starts data transmission two clocks after a response for the data write command is received; this occurs even if the command path detects a response error or response CRC error. If a response is not received from the card because of a response timeout, data is not transmitted. Depending upon the value of the transfer_mode bit in the Command register, the data transmit state machine puts data on the card data bus in a stream or in block(s).

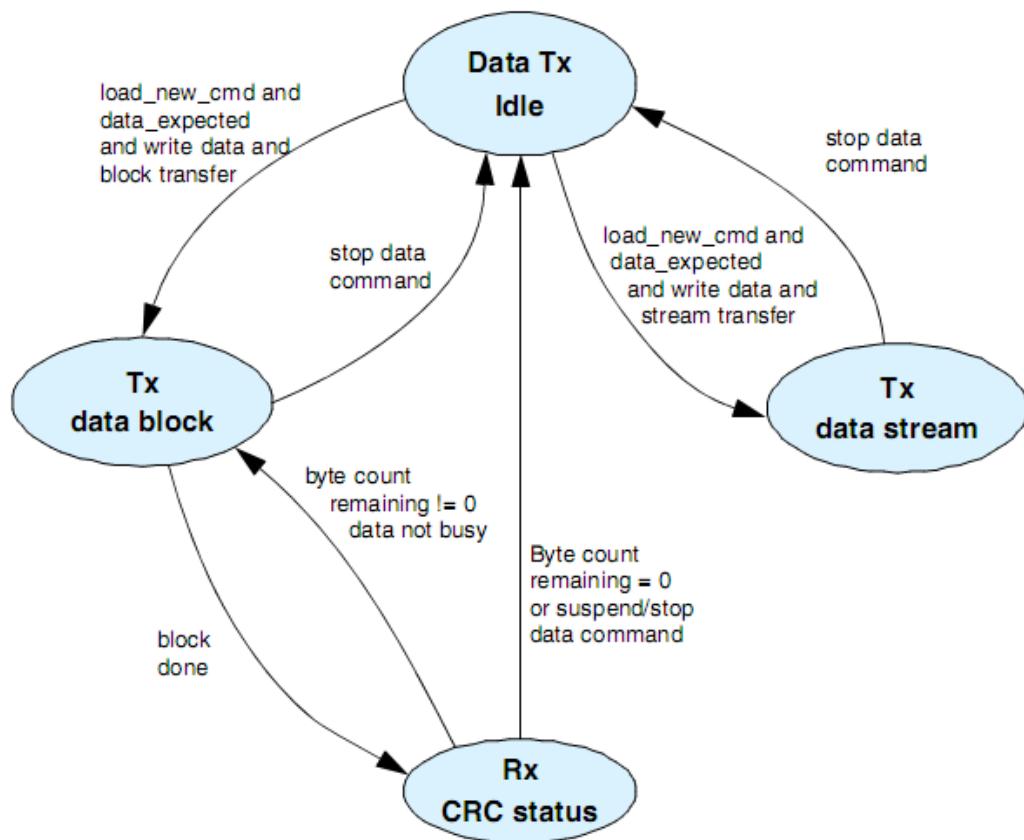


Fig. 13-4 Host Controller Data Transmit State Machine

Stream Data Transmit

If the transfer_mode bit in the Command register is set to 1, it is a stream-write data transfer. The data path pops the FIFO from the BIU and transmits in a stream to the card data bus. If the FIFO becomes empty, the card clock is stopped and restarted once data is available in the FIFO.

If the byte_count register is programmed to 0, it is an open-ended stream-write data transfer. During this data transfer, the data path continuously transmits data in a stream until the host software issues a stop command. A stream data transfer is terminated when the end bit of the stop command and end bit of the data match over two clocks.

If the byte_count register is programmed with a non-zero value and the send_auto_stop bit is set in the Command register, the stop command is internally generated and loaded in the command path when the end bit of the stop command occurs after the last byte of the stream write transfer matches.

This data transfer can also terminate if the host issues a stop command before all the data bytes are transferred to the card bus.

Single Block Data

If the transfer_mode bit in the Command register is set to 0 and the byte_count register value is equal to the value of the block_size register, a single-block write-data transfer occurs. The data transmit state machine sends data in a

single block, where the number of bytes equals the block size, including the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set for a 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted for 1, 4, or 8 data lines, respectively.

After a single data block is transmitted, the data transmit state machine receives the CRC status from the card and signals a data transfer to the BIU; this happens when the data-transfer-over bit is set in the RINTSTS register.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the RINTSTS register.

Additionally, if the start bit of the CRC status is not received by two clocks after the end of the data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the RINTSTS register.

Multiple Block Data

A multiple-block write-data transfer occurs if the transfer_mode bit in the Command register is set to 0 and the value in the byte_count register is not equal to the value of the block_size register. The data transmit state machine sends data in blocks, where the number of bytes in a block equals the block size, including the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted on 1, 4, or 8 data lines, respectively.

After one data block is transmitted, the data transmit state machine receives the CRC status from the card. If the remaining byte_count becomes 0, the data path signals to the BIU that the data transfer is done; this happens when the data-transfer-over bit is set in the RINTSTS register.

If the remaining data bytes are greater than 0, the data path state machine starts to transmit another data block.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the RINTSTS register, and continues further data transmission until all the bytes are transmitted.

Additionally, if the CRC status start bit is not received by two clocks after the end of a data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the RINTSTS register; further data transfer is terminated.

If the send_auto_stop bit is set in the Command register, the stop command is internally generated during the transfer of the last data block, where no extra bytes are transferred to the card. The end bit of the stop command may not exactly match the end bit of the CRC status in the last data block.

If the block size is less than 4, 16, or 32 for card data widths of 1 bit, 4 bits, or 8 bits, respectively, the data transmit state machine terminates the data transfer when all the data is transferred, at which time the internally generated stop command is loaded in the command path.

If the byte_count is 0 – the block size must be greater than 0 – it is an open-ended block transfer. The data transmit state machine for this type of data transfer continues the block-write data transfer until the host software issues a stop or abort command.

Data Receive

The data-receive state machine, illustrated in following figure, receives data two clock cycles after the end bit of a data read command, even if the command path detects a response error or response CRC error. If a response is not received from the card because a response timeout occurs, the BIU does not receive a signal that the data transfer is complete; this happens if the command sent by the Host Controller is an illegal operation for the card, which keeps the card from starting a read data transfer.

If data is not received before the data timeout, the data path signals a data timeout to the BIU and an end to the data transfer done. Based on the value of the transfer_mode bit in the Command register, the data-receive state machine gets data from the card data bus in a stream or block(s).

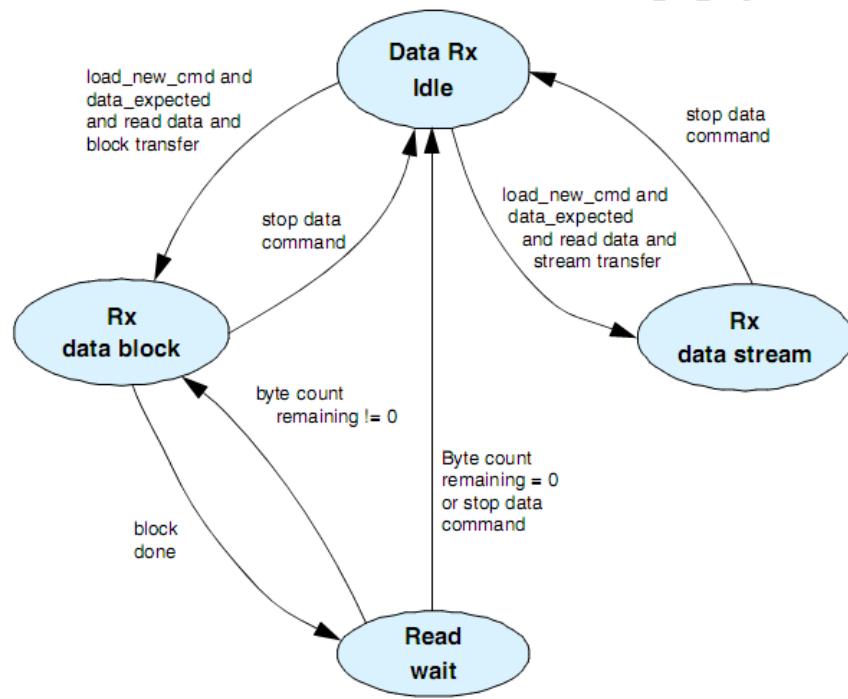


Fig. 13-5 Host Controller Data Receive State Machine

Stream Data Read

A stream-read data transfer occurs if the transfer_mode bit in the Command register equals 1, at which time the data path receives data from the card and pushes it to the FIFO. If the FIFO becomes full, the card clock stops and restarts once the FIFO is no longer full.

An open-ended stream-read data transfer occurs if the byte_count register equals 0. During this type of data transfer, the data path continuously receives data in a stream until the host software issues a stop command. A stream data

transfer terminates two clock cycles after the end bit of the stop command.

If the byte_count register contains a non-zero value and the send_auto_stop bit is set in the Command register, a stop command is internally generated and loaded into the command path, where the end bit of the stop command occurs after the last byte of the stream data transfer is received. This data transfer can terminate if the host issues a stop or abort command before all the data bytes are received from the card.

Single-Block Data Read

A single-block read-data transfer occurs if the transfer_mode bit in the Command register is set to 0 and the value of the byte_count register is equal to the value of the block_size register. When a start bit is received before the data times out, data bytes equal to the block size and CRC16 are received and checked with the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively. If there is a CRC16 mismatch, the data path signals a data CRC error to the BIU. If the received end bit is not 1, the BIU receives an end-bit error.

Multiple-Block Data Read

If the transfer_mode bit in the Command register is set to 0 and the value of the byte_count register is not equal to the value of the block_size register, it is a multiple-block read-data transfer. The data-receive state machine receives data in blocks, where the number of bytes in a block is equal to the block size, including the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively.

After a data block is received, if the remaining byte_count becomes 0, the data path signals a data transfer to the BIU.

If the remaining data bytes are greater than 0, the data path state machine causes another data block to be received. If CRC16 of a received data block does not match the internally-generated CRC16, a data CRC error to the BIU and data reception continue further data transmission until all bytes are transmitted.

Additionally, if the end of a received data block is not 1, data on the data path signals terminate the bit error to the CIU and the data-receive state machine terminates data reception, waits for data timeout, and signals to the BIU that the data transfer is complete.

If the send_auto_stop bit is set in the Command register, the stop command is internally generated when the last data block is transferred, where no extra bytes are transferred from the card; the end bit of the stop command may not exactly match the end bit of the last data block.

If the requested block size for data transfers to cards is less than 4, 16, or 32

bytes for 1-bit, 4-bit, or 8-bit data transfer modes, respectively, the data-transmit state machine terminates the data transfer when all data is transferred, at which point the internally-generated stop command is loaded in the command path. Data received from the card after that are then ignored by the data path.

If the byte_count is 0—the block size must be greater than 0—it is an open-ended block transfer. For this type of data transfer, the data-receive state machine continues the block-read data transfer until the host software issues a stop or abort command.

Auto-Stop

The Host Controller internally generates a stop command and is loaded in the command path when the send_auto_stop bit is set in the Command register.

The software should set the send_auto_stop bit according to details listed in following table.

Table 13-2 Auto-Stop Generation

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
MMC	Stream read	0	No	Open-ended stream
MMC	Stream read	>0	Yes	Auto-stop after all bytes transfer
MMC	Stream write	0	No	Open-ended stream
MMC	Stream write	>0	Yes	Auto-stop after all bytes transfer
MMC	Single-block read	>0	No	Byte count =0 is illegal
MMC	Single-block write	>0	No	Byte count =0 is illegal
MMC	Multiple-block read	0	No	Open-ended multiple block
MMC	Multiple-block read	>0	Yes ^①	Pre-defined multiple block
MMC	Multiple-block write	0	No	Open-ended multiple block
MMC	Multiple-block write	>0	Yes ^①	Pre-defined multiple block
SDMEM	Single-block read	>0	No	Byte count =0 is illegal
SDMEM	Single-block write	>0	No	Byte count =0 illegal
SDMEM	Multiple-block read	0	No	Open-ended multiple block
SDMEM	Multiple-block read	>0	Yes	Auto-stop after all bytes transfer
SDMEM	Multiple-block write	0	No	Open-ended multiple block
SDMEM	Multiple-block write	>0	Yes	Auto-stop after all bytes transfer
SDIO	Single-block read	>0	No	Byte count =0 is illegal
SDIO	Single-block write	>0	No	Byte count =0 illegal
SDIO	Multiple-block read	0	No	Open-ended multiple block
SDIO	Multiple-block read	>0	No	Pre-defined multiple block
SDIO	Multiple-block write	0	No	Open-ended multiple block
SDIO	Multiple-block write	>0	No	Pre-defined multiple block

^①:The condition under which the transfer mode is set to block transfer and byte_count is equal to block size is treated as a single-block data transfer command for both MMC and SD cards. If byte_count = n*block_size (n = 2, 3, ...), the condition is treated as a predefined multiple-block data transfer command. In the case of an MMC card, the host software can perform a predefined data transfer in two ways: 1) Issue the CMD23 command before issuing CMD18/CMD25 commands to the card – in this case, issue MD18/CMD25 commands without setting the send_auto_stop bit. 2) Issue CMD18/CMD25 commands without issuing CMD23 command to the card, with the send_auto_stop bit set. In this case, the multiple-block data transfer is terminated by an internally-generated auto-stop command after the programmed byte count.

The following list conditions for the auto-stop command.

- Stream read for MMC card with byte count greater than 0 – The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent out when the last byte of data is read from the card and no extra data byte is received. If the byte count is less than 6 (48 bits), a few extra data bytes are received from the card before the end bit of the stop command is sent.
- Stream write for MMC card with byte count greater than 0 - The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent when the last byte of data is transmitted on the card bus and no extra data byte is transmitted. If the byte count is less than 6 (48 bits), the data path transmits the data last in order to meet the above condition.
- Multiple-block read memory for SD card with byte count greater than 0 – If the block size is less than 4 (single-bit data bus), 16 (4-bit data bus), or 32 (8-bit data bus), the auto-stop command is loaded in the command path after all the bytes are read. Otherwise, the top command is loaded in the command path so that the end bit of the stop command is sent after the last data block is received.
- Multiple-block write memory for SD card with byte count greater than 0 – If the block size is less than 3 (single-bit data bus), 12 (4-bit data bus), or 24 (8-bit data bus), the auto-stop command is loaded in the command path after all data blocks are transmitted. Otherwise, the stop command is loaded in the command path so that the end bit of the stop command is sent after the end bit of the CRC status is received.
- Precaution for host software during auto-stop – Whenever an auto-stop command is issued, the host software should not issue a new command to the SD/MMC device until the auto-stop is sent by the Host Controller and the data transfer is complete. If the host issues a new command during a data transfer with the auto-stop in progress, an auto-stop command may be sent after the new command is sent and its response is received; this can delay sending the stop command, which transfers extra data bytes. For a stream write, extra data bytes are erroneous data that can corrupt the card data. If the host wants to terminate the data transfer before the data transfer is complete, it can issue a stop or abort command, in which case the Host Controller does not generate an auto-stop command.

3. Non-Data Transfer Commands that Use Data Path

Some non-data transfer commands (non-read/write commands) also use the data path. Following table lists the commands and register programming requirements for them.

Table 13-3 Non-data Transfer Commands and Requirements

	CMD27	CMD30	CMD42	ACMD13	ACMD22	ACMD51
Command register programming						
cmd_index	6'h1B	6'h1E	6'h2A	6'h0D	6'h16	6'h33
response_expect	1	1	1	1	1	1
rResponse_length	0	0	0	0	0	0
check_response_crc	1	1	1	1	1	1
data_expected	1	1	1	1	1	1
read/write	1	0	1	0	0	0
transfer_mode	0	0	0	0	0	0

send_auto_stop	0	0	0	0	0	0
wait_prevdata_comp lete	0	0	0	0	0	0
stop_abort_cmd	0	0	0	0	0	0
Command Argument register programming						
	stuff bits	32-bit writeprotect dataaddress	stuff bits	stuff bits	stuff bits	stuff bits
Block Size register programming						
	16	4	Num_by tes ^①	64	4	8
Byte Count register programming						
	16	4	Num_by tes ^①	64	4	8

①: Num_bytes = No. of bytes specified as per the lock card data structure (Refer to the SD specification and the MMC specification)

4. SDIO Interrupt Control

Interrupts for SD cards are reported to the BIU by asserting an interrupt signal for two clock cycles. SDIO cards signal an interrupt by asserting cdata_in low during the interrupt period; an interrupt period for the selected card is determined by the interrupt control state machine. An interrupt period is always valid for non-active or non-selected cards, and 1-bit data mode for the selected card. An interrupt period for a wide-bus active or selected card is valid for the following conditions:

- Card is idle
- Non-data transfer command in progress
- Third clock after end bit of data block between two data blocks
- From two clocks after end bit of last data until end bit of next data transfer command

Bear in mind that, in the following situations, the controller does not sample the SDIO interrupt of the selected card when the card data width is 4 bits. Since the SDIO interrupt is level-triggered, it is sampled in a further interrupt period and the host does not lose any SDIO interrupt from the card.

- Read/Write Resume – The CIU treats the resume command as a normal data transfer command. SDIO interrupts during the resume command are handled similarly to other data commands. According to the SDIO specification, for the normal data command the interrupt period ends after the command end bit of the data command; for the resume command, it ends after the response end bit. In the case of the resume command, the Controller stops the interrupt sampling period after the resume command end bit, instead of stopping after the response end bit of the resume command.
- Suspend during read transfer – If the read data transfer is suspended by the host, the host sets the abort_read_data bit in the controller to reset the data state machine. In the CIU, the SDIO interrupts are handled such that the interrupt sampling starts after the abort_read_data bit is set by the host. In this case the controller does not sample SDIO interrupts between the period from response of the suspend command to setting the abort_read_data bit, and starts sampling after setting the abort_read_data bit.

5. Clock Control

The clock control block provides different clock frequencies required for SD/MMC cards. The `cclk_in` signal is the source clock (`cclk_in >= card max operating frequency`) for clock divider of the clock control block. This source clock (`cclk_in`) is used to generate different card clock frequencies (`cclk_out`). The card clock can have different clock frequencies, since the card can be a low-speed card or a full-speed card. The Host Controller provides one clock signal (`cclk_out`).

The clock frequency of a card depends on the following clock control registers:

- Clock Divider register – Internal clock dividers are used to generate different clock frequencies required for card. The division factor for each clock divider can be programmed by writing to the Clock Divider register. The clock divider is an 8-bit value that provides a clock division factor from 1 to 510; a value of 0 represents a clock-divider bypass, a value of 1 represents a divide by 2, a value of 2 represents a divide by 4, and so on.
- Clock Control register – `cclk_out` can be enabled or disabled for each card under the following conditions:
 - `clk_enable` – `cclk_out` for a card is enabled if the `clk_enable` bit for a card in the Clock Control register is programmed (set to 1) or disabled (set to 0).
 - Low-power mode – Low-power mode of a card can be enabled by setting the low-power mode bit of the Clock Control register to 1. If low-power mode is enabled to save card power, the `cclk_out` is disabled when the card is idle for at least 8 card clock cycles. It is enabled when a new command is loaded and the command path goes to a non-idle state.

Additionally, `cclk_out` is disabled when an internal FIFO is full – card read (no more data can be received from card) – or when the FIFO is empty – card write (no data is available for transmission). This helps to avoid FIFO overrun and underrun conditions. It is used by the command and data path to qualify `cclk_in` for driving outputs and sampling inputs at the programmed clock frequency for the selected card, according to the Clock Divider and Clock Source register values.

Under the following conditions, the card clock is stopped or disabled, along with the active `clk_en`, for the selected card:

- Clock can be disabled by writing to Clock Enable register (`clk_en` bit = 1).
- If low-power mode is selected and card is idle, or not selected for 8 clocks.
- FIFO is full and data path cannot accept more data from the card and data transfer is incomplete –to avoid FIFO overrun.
- FIFO is empty and data path cannot transmit more data to the card and data transfer is incomplete – to avoid FIFO underrun.

6. Error Detection

- Response
 - Response timeout – Response expected with response start bit is not received within programmed number of clocks in timeout register.
 - Response CRC error – Response is expected and check response CRC requested; response CRC7 does not match with the internally-generated CRC7.

- Response error – Response transmission bit is not 0, command index does not match with the command index of the send command, or response end bit is not 1.
- Data transmit
 - No CRC status – During a write data transfer, if the CRC status start bit is not received two clocks after the end bit of the data block is sent out, the data path does the following:
 - ◆ Signals no CRC status error to the BIU
 - ◆ Terminates further data transfer
 - ◆ Signals data transfer done to the BIU
 - Negative CRC – If the CRC status received after the write data block is negative (that is, not 010), a data CRC error is signaled to the BIU and further data transfer is continued.
 - Data starvation due to empty FIFO – If the FIFO becomes empty during a write data transmission, or if the card clock is stopped and the FIFO remains empty for data timeout clocks, then a data-starvation error is signaled to the BIU and the data path continues to wait for data in the FIFO.
- Data receive
 - Data timeout – During a read-data transfer, if the data start bit is not received before the number of clocks that were programmed in the timeout register, the data path does the following:
 - ◆ Signals data-timeout error to the BIU
 - ◆ Terminates further data transfer
 - ◆ Signals data transfer done to BIU
 - Data start bit error – During a 4-bit or 8-bit read-data transfer, if the all-bit data line does not have a start bit, the data path signals a data start bit error to the BIU and waits for a data timeout, after which it signals that the data transfer is done.
 - Data CRC error – During a read-data-block transfer, if the CRC16 received does not match with the internally generated CRC16, the data path signals a data CRC error to the BIU and continues further data transfer.
 - Data end-bit error – During a read-data transfer, if the end bit of the received data is not 1, the data path signals an end-bit error to the BIU, terminates further data transfer, and signals to the BIU that the data transfer is done.
 - Data starvation due to FIFO full – During a read data transmission and when the FIFO becomes full, the card clock is stopped. If the FIFO remains full for data timeout clocks, a data starvation error is signaled to the BIU (Data Starvation by Host Timeout bit is set in RINTSTS Register) and the data path continues to wait for the FIFO to start to empty.

13.4 Register Description

13.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
SDMMC_CTRL	0x00000	W	0x01000000	Control register
SDMMC_PWREN	0x00004	W	0x00000000	Power-enable register
SDMMC_CLKDIV	0x00008	W	0x00000000	Clock-divider register

Name	Offset	Size	Reset Value	Description
SDMMC_CLKSRC	0x0000c	W	0x00000000	SD Clock Source Register
SDMMC_CLKENA	0x00010	W	0x00000000	Clock-enable register
SDMMC_TMOOUT	0x00014	W	0xfffffff40	Time-out register
SDMMC_CTYPE	0x00018	W	0x00000000	Card-type register
SDMMC_BLKSIZ	0x0001c	W	0x00000200	Block-size register
SDMMC_BYTCNT	0x00020	W	0x00000200	Byte-count register
SDMMC_INTMASK	0x00024	W	0x00000000	Interrupt-mask register
SDMMC_CMDARG	0x00028	W	0x00000000	Command-argument register
SDMMC_CMD	0x0002c	W	0x00000000	Command register
SDMMC_RESP0	0x00030	W	0x00000000	Response-0 register
SDMMC_RESP1	0x00034	W	0x00000000	Response-1 register
SDMMC_RESP2	0x00038	W	0x00000000	Response-2 register
SDMMC_RESP3	0x0003c	W	0x00000000	Response-3 register
SDMMC_MINTSTS	0x00040	W	0x00000000	Masked interrupt-status register
SDMMC_RINTSTS	0x00044	W	0x00000000	Raw interrupt-status register
SDMMC_STATUS	0x00048	W	0x00000406	Status register
SDMMC_FIFOTH	0x0004c	W	0x00000000	FIFO threshold register
SDMMC_CDETECT	0x00050	W	0x00000000	Card-detect register
SDMMC_WRTPRT	0x00054	W	0x00000000	Write-protect register
SDMMC_TCBCNT	0x0005c	W	0x00000000	Transferred CIU card byte count
SDMMC_TBBCNT	0x00060	W	0x00000000	Transferred host/DMA to/from BIU-FIFO byte count
SDMMC_DEBNCE	0x00064	W	0x00ffffff	Card detect debounce register
SDMMC_USRID	0x00068	W	0x07967797	User ID register
SDMMC_VERID	0x0006c	W	0x5342270a	Synopsys version ID register
SDMMC_HCON	0x00070	W	0x00000000	Hardware Configuration Register
SDMMC_UHS_REG	0x00074	W	0x00000000	UHS-1 register
SDMMC_RST_n	0x00078	W	0x00000001	Hardware reset register
SDMMC_PLDMND	0x00084	W	0x00000000	Poll Demand Register
SDMMC_CARDTHRCTL	0x00100	W	0x00000000	Card read threshold enable
SDMMC_BACK_END_POWER	0x00104	W	0x00000000	Back-end power
SDMMC_UHS_REG_EXT	0x00108	W	0x00000000	UHS Register
SDMMC_EMMC_DDR_REG	0x0010c	W	0x00000000	eMMC 4.5 DDR START Bit Detection Control Register
SDMMC_ENABLE_SHIFT	0x00110	W	0x00000000	Enable Phase Shift Register
SDMMC_FIFO_BASE	0x00200	W	0x00000000	FIFO Base Address

Notes: **Size:** **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** -WORD (32 bits) access

13.4.2 Detail Register Description

SDMMC_CTRL

Address: Operational Base + offset (0x000000)

Control register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	use_internal_dmac Present only for the Internal DMAC configuration; else, it is reserved. 0: The host performs data transfers through the slave interface 1: Internal DMAC used for data transfer
24	RW	0x1	enable_OD_pullup External open-drain pullup: 0: Disable 1: Enable Inverted value of this bit is output to ccmd_od_pullup_en_n port. When bit is set, command output always driven in open-drive mode; that is, DWC_mobile_storage drives either 0 or high impedance, and does not drive hard 1.
23:20	RW	0x0	Card_voltage_b Card regulator-B voltage setting; output to card_volt_b port. Optional feature; ports can be used as general-purpose outputs.
19:16	RW	0x0	Card_voltage_a Card regulator-A voltage setting; output to card_volt_a port. Optional feature; ports can be used as general-purpose outputs.
15:12	RO	0x0	reserved
11	RW	0x0	ceata_device_interrupt_status 0: Interrupts not enabled in CE-ATA device (nIEN = 1 in ATA control register) 1: Interrupts are enabled in CE-ATA device (nIEN = 0 in ATA control register) Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled (nIEN = 1). If the host enables CE-ATA device interrupt, then software should set this bit.

Bit	Attr	Reset Value	Description
10	RW	0x0	<p>send_auto_stop_ccsd 0: Clear bit if DWC_mobile_storage does not reset the bit. 1: Send internally generated STOP after sending CCSD to CE-ATA device. NOTE: Always set send_auto_stop_ccsd and send_ccsd bits together send_auto_stop_ccsd should not be set independent of send_ccsd. When set, DWC_Mobile_Storage automatically sends internally-generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) bit in RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending the CCSD, DWC_mobile_storage automatically clears send_auto_stop_ccsd bit.</p>
9	RW	0x0	<p>send_ccsd 0: Clear bit if DWC_mobile_storage does not reset the bit. 1: Send Command Completion Signal Disable (CCSD) to CE-ATA device When set, DWC_mobile_storage sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device. Once the CCSD pattern is sent to device, DWC_mobile_storage automatically clears send_ccsd bit. It also sets Command Done (CD) bit in RINTSTS register and generates interrupt to host if Command Done interrupt is not masked. NOTE: Once send_ccsd bit is set, it takes two card clock cycles to drive the CCSD on the CMD line. Due to this, during the boundary conditions it may happen that CCSD is sent to the CE-ATA device, even if the device signalled CCS</p>
8	RW	0x0	<p>abort_read_data 0: no change 1: after suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Bit automatically clears once data state machine resets to idle. Used in SDIO card suspend sequence.</p>

Bit	Attr	Reset Value	Description
7	RW	0x0	send_irq_response 0: no change 1: send auto IRQ response Bit automatically clears once response is sent. To wait for MMC card interrupts, host issues CMD40, and SDMMC Controller waits for interrupt response from MMC card(s). In meantime, if host wants SDMMC Controller to exit waiting for interrupt state, it can set this bit, at which time SDMMC Controller command state-machine sends CMD40 response on bus and returns to idle state.
6	RW	0x0	read_wait 0: clear read wait 1: assert read wait For sending read-wait to SDIO cards
5	RW	0x0	dma_enable 0: disable DMA transfer mode 1: enable DMA transfer mode Even when DMA mode is enabled, host can still push/pop data into or from FIFO; this should not happen during the normal operation. If there is simultaneous FIFO access from host/DMA, the data coherency is lost. Also, there is no arbitration inside SDMMC Controller to prioritize simultaneous host/DMA access.
4	RW	0x0	int_enable Global interrupt enable/disable bit: 0: disable interrupts 1: enable interrupts The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set.
3	RO	0x0	reserved
2	W1C	0x0	dma_reset 0: no change 1: reset internal DMA interface control logic To reset DMA interface, firmware should set bit to 1. This bit is auto-cleared after two AHB clocks.
1	W1C	0x0	fifo_reset 0: no change 1: reset to data FIFO To reset FIFO pointers To reset FIFO, firmware should set bit to 1. This bit is auto-cleared after completion of reset operation

Bit	Attr	Reset Value	Description
0	W1C	0x0	<p>controller_reset 0: no change 1: reset SDMMC controller To reset controller, firmware should set bit to 1. This bit is auto-cleared after two AHB and two cclk_in clock cycles.</p> <p>This resets: * BIU/CIU interface * CIU and state machines * abort_read_data, send_irq_response, and read_wait bits of Control register * start_cmd bit of Command register Does not affect any registers or DMA interface, or FIFO or host interrupts</p>

SDMMC_PWREN

Address: Operational Base + offset (0x00004)

Power-enable register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	<p>power_enable Power on/off switch for the card. Once power is turned on, firmware should wait for regulator/switch ramp-up time before trying to initialize card.</p> <p>0: power off 1: power on Bit values output to card_power_en port.</p>

SDMMC_CLKDIV

Address: Operational Base + offset (0x00008)

Clock-divider register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	<p>clk_divider0 Clock divider-0 value. Clock division is 2^n. For example, value of 0 means divide by $2^0 = 0$ (no division, bypass), value of 1 means divide by $2^1 = 2$, value of "ff" means divide by $2^{255} = 510$, and so on. divider>3 is not recommended.</p>

SDMMC_CLKSRC

Address: Operational Base + offset (0x0000c)

SD Clock Source Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	<p>clk_source Clock divider source for up to 16 SD cards supported. Each card has two bits assigned to it. For example, bits[1:0] assigned for card-0, which maps and internally routes clock divider[3:0] outputs to cclk_out[15:0] pins, depending on bit value.</p> <ul style="list-style-type: none"> 00 –Clock divider 0 01 –Clock divider 1 10 –Clock divider 2 11 –Clock divider 3 <p>In MMC-Ver3.3-only controller, only one clock divider supported. The cclk_out is always from clock divider 0, and this register is not implemented.</p>

SDMMC_CLKENA

Address: Operational Base + offset (0x00010)

Clock-enable register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	<p>cclk_low_power Low-power control for SD card clock and MMC card clock supported. 0: non-low-power mode 1: low-power mode; stop clock when card in IDLE (should be normally set to only MMC and SD memory cards; for SDIO cards, if interrupts must be detected, clock should not be stopped).</p>
15:1	RO	0x0	reserved
0	RW	0x0	<p>cclk_enable Clock-enable control for SD card clock and MMC card clock supported. 0: clock disabled 1: clock enabled</p>

SDMMC_TMOUT

Address: Operational Base + offset (0x00014)

Time-out register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:8	RW	0xffffffff	<p>data_timeout Value for card Data Read Timeout; same value also used for Data Starvation by Host timeout. Value is in number of card output clocks cclk_out of selected card.</p> <p>Note: The software timer should be used if the timeout value is in the order of 100 ms. In this case, read data timeout interrupt needs to be disabled.</p>
7:0	RW	0x40	<p>response_timeout Response timeout value. Value is in number of card output clocks -cclk_out.</p>

SDMMC_CTYPE

Address: Operational Base + offset (0x00018)

Card-type register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	<p>card_width_8 Indicates if card is 8-bit: 0: non 8-bit mode 1: 8-bit mode</p>
15:1	RO	0x0	reserved
0	RW	0x0	<p>card_width Indicates if card is 1-bit or 4-bit: 0: 1-bit mode 1: 4-bit mode</p>

SDMMC_BLKSIZ

Address: Operational Base + offset (0x0001c)

Block-size register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0200	block_size Block size

SDMMC_BYTCNT

Address: Operational Base + offset (0x00020)

Byte-count register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000200	byte_count Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer.

SDMMC_INTMASK

Address: Operational Base + offset (0x00024)

Interrupt-mask register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	sdio_int_mask Mask SDIO interrupts. When masked, SDIO interrupt detection for that card is disabled. A 0 masks an interrupt, and 1 enables an interrupt.
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_mask 0: data no busy interrupt not masked 1: data no busy interrupt masked
15:0	RW	0x0000	int_mask Bits used to mask unwanted interrupts. Value of 0 masks interrupt; value of 1 enables interrupt. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_CMDARG

Address: Operational Base + offset (0x00028)

Command-argument register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cmd_arg Value indicates command argument to be passed to card.

SDMMC_CMD

Address: Operational Base + offset (0x0002c)

Command register

Bit	Attr	Reset Value	Description
31	RW	0x0	start_cmd Start command. Once command is taken by CIU, bit is cleared. When bit is set, host should not attempt to write to any command registers. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD_MMC cards, Command Done bit is set in raw interrupt register.
30	RO	0x0	reserved
29	RW	0x0	use_hold_reg Use Hold Register 0: CMD and DATA sent to card bypassing HOLD Register 1: CMD and DATA sent to card through the HOLD Register Note: a. Set to 1'b1 for SDR12 and SDR25 (with non-zero phase-shifted cclk_in_drv); zero phase shift is not allowed in these modes. b. Set to 1'b0 for SDR50, SDR104, and DDR50 (with zero phase-shifted cclk_in_drv) c. Set to 1'b1 for SDR50, SDR104, and DDR50 (with non-zero phase-shifted cclk_in_drv)
28	RW	0x0	volt_switch Voltage switch bit. 0: no voltage switching 1: voltage switching enabled; must be set for CMD11 only
27	RW	0x0	boot_mode Boot Mode. 0: mandatory Boot operation 1: alternate Boot operation
26	RW	0x0	disable_boot Disable Boot. When software sets this bit along with start_cmd, CIU terminates the boot operation. Do NOT set disable_boot and enable_boot together.

Bit	Attr	Reset Value	Description
25	RW	0x0	expect_boot_ack Expect Boot Acknowledge. When Software sets this bit along with enable_boot, CIU expects a boot acknowledge start pattern of 0-1-0 from the selected card.
24	RW	0x0	enable_boot Enable Boot—this bit should be set only for mandatory boot mode. When Software sets this bit along with start_cmd, CIU starts the boot sequence for the corresponding card by asserting the CMD line low. Do NOT set disable_boot and enable_boot together.
23	RW	0x0	ccs_expected 0: Interrupts are not enabled in CE-ATA device (nIEN = 1 in ATA control register), or command does not expect CCS from device 1: Interrupts are enabled in CE-ATA device (nIEN = 0), and RW_BLK command expects command completion signal from CE-ATA device If the command expects Command Completion Signal (CCS) from the CE-ATA device, the software should set this control bit. DWC_mobile_storage sets Data Transfer Over (DTO) bit in RINTSTS register and generates interrupt to host if Data Transfer Over interrupt is not masked.
22	RW	0x0	read_ceata_device 0: Host is not performing read access (RW_REG or RW_BLK) towards CE-ATA device 1: Host is performing read access (RW_REG or RW_BLK) towards CE-ATA device Software should set this bit to indicate that CE-ATA device is being accessed for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. DWC_mobile_storage should not indicate read data timeout while waiting for data from CE-ATA device.

Bit	Attr	Reset Value	Description
21	RW	0x0	<p>update_clock_registers_only 0: normal command sequence 1: do not send commands, just update clock register value into card clock domain Following register values transferred into card clock domain: CLKDIV, CLRSRC, CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards.</p> <p>During normal command sequence, when update_clock_registers_only = 0, following control registers are transferred from BIU to CIU: CMD, CMDARG, TMOUT, CTYPE, BLKSIZ, BYTCNT. CIU uses new register values for new command sequence to card. When bit is set, there are no Command Done interrupts because no command is sent to SD_MMC_CEATA cards.</p>
20:16	RW	0x00	<p>card_number Card number in use. Represents physical slot number of card being accessed. In MMC-Ver3.3-only mode, up to 30 cards are supported; in SD-only mode, up to 16 cards are supported. Registered version of this is reflected on dw_dma_card_num and ge_dma_card_num ports, which can be used to create separate DMA requests, if needed. In addition, in SD mode this is used to mux or demux signals from selected card because each card is interfaced to DWC_mobile_storage by separate bus.</p>
15	RW	0x0	<p>send_initialization 0: do not send initialization sequence (80 clocks of 1) before sending this command 1: send initialization sequence before sending this command After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller will initialize clocks before sending command to card. This bit should not be set for either of the boot modes (alternate or mandatory).</p>

Bit	Attr	Reset Value	Description
14	RW	0x0	<p>stop_abort_cmd 0: neither stop nor abort command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1: stop or abort command intended to stop current data transfer in progress.</p> <p>When open-ended or predefined data transfer is in progress, and host issues stop or abort command to stop data transfer, bit should be set so that command/data state-machines of CIU can return correctly to idle state. This is also applicable for Boot mode transfers. To Abort boot mode, this bit should be set along with CMD[26] = disable_boot.</p>
13	RW	0x0	<p>wait_prvdata_complete 0: send command at once, even if previous data transfer has not completed 1: wait for previous data transfer completion before sending command</p> <p>The wait_prvdata_complete = 0 option typically used to query status of card during data transfer or to stop current data transfer; card_number should be same as in previous command.</p>
12	RW	0x0	<p>send_auto_stop 0: no stop command sent at end of data transfer 1: send stop command at end of data transfer</p> <p>When set, SDMMC Controller sends stop command to SD_MMC cards at end of data transfer.</p> <ul style="list-style-type: none"> * when send_auto_stop bit should be set, since some data transfers do not need explicit stop commands * open-ended transfers that software should explicitly send to stop command <p>Additionally, when "resume" is sent to resume -suspended memory access of SD-Combo card –bit should be set correctly if suspended data transfer needs send_auto_stop.</p> <p>Don't care if no data expected from card.</p>
11	RW	0x0	<p>transfer_mode 0: block data transfer command 1: stream data transfer command</p> <p>Don't care if no data expected.</p>
10	RW	0x0	<p>wr 0: read from card 1: write to card</p> <p>Don't care if no data expected from card.</p>

Bit	Attr	Reset Value	Description
9	RW	0x0	data_expected 0: no data transfer expected (read/write) 1: data transfer expected (read/write)
8	RW	0x0	check_response_crc 0: do not check response CRC 1: check response CRC Some of command responses do not return valid CRC bits. Software should disable CRC checks for those commands in order to disable CRC checking by controller
7	RW	0x0	response_length 0: short response expected from card 1: long response expected from card
6	RW	0x0	response_expect 0: no response expected from card 1: response expected from card
5:0	RW	0x00	cmd_index Command index

SDMMC_RESP0

Address: Operational Base + offset (0x00030)

Response-0 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response0 Bit[31:0] of response

SDMMC_RESP1

Address: Operational Base + offset (0x00034)

Response-1 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response Register represents bit[63:32] of long response. When CIU sends auto-stop command, then response is saved in register. Response for previous command sent by host is still preserved in Response 0 register. Additional auto-stop issued only for data transfer commands, and response type is always "short" for them.

SDMMC_RESP2

Address: Operational Base + offset (0x00038)

Response-2 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response2 Bit[95:64] of long response

SDMMC_RESP3

Address: Operational Base + offset (0x0003c)

Response-3 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response3 Bit[127:96] of long response

SDMMC_MINTSTS

Address: Operational Base + offset (0x00040)

Masked interrupt-status register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RO	0x0	sdio_interrupt Interrupt from SDIO card; SDIO interrupt for card enabled only if corresponding sdio_int_mask bit is set in Interrupt mask register (mask bit 1 enables interrupt; 0 masks interrupt). 0: no SDIO interrupt from card 1: SDIO interrupt from card
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_status Data no busy Interrupt Status
15:0	RO	0x0000	int_status Interrupt enabled only if corresponding bit in interrupt mask register is set. bit 15: End-bit error (read)/Write no CRC (EBC) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_RINTSTS

Address: Operational Base + offset (0x00044)

Raw interrupt-status register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24	RO	0x0	sdio_interrupt Interrupt from SDIO card; Writes to these bits clear them. Value of 1 clears bit and 0 leaves bit intact. 0: no SDIO interrupt from card 1: SDIO interrupt from card
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_status Data no busy interrupt status
15:0	RO	0x0000	int_status Writes to bits clear status bit. Value of 1 clears status bit, and value of 0 leaves bit intact. Bits are logged regardless of interrupt mask status. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_STATUS

Address: Operational Base + offset (0x00048)

Status register

Bit	Attr	Reset Value	Description
31	RO	0x0	dma_req DMA request signal state
30	RO	0x0	dma_ack DMA acknowledge signal state
29:17	RO	0x0000	fifo_count Number of filled locations in FIFO
16:11	RO	0x00	response_index Index of previous response, including any auto-stop sent by core
10	RO	0x1	data_state_mc_busy Data transmit or receive state-machine is busy

Bit	Attr	Reset Value	Description
9	RO	0x0	data_busy Inverted version of raw selected card_data[0] 0: card data not busy 1: card data busy default value is 1 or 0 depending on cdata_in
8	RO	0x0	data_3_status Raw selected card_data[3]; checks whether card is present 0: card not present 1: card present default value is 1 or 0 depending on cdata_in
7:4	RO	0x0	command_fsm_states Command FSM states: 0: idle 1: send init sequence 2: Tx cmd start bit 3: Tx cmd tx bit 4: Tx cmd index + arg 5: Tx cmd crc7 6: Tx cmd end bit 7: Rx resp start bit 8: Rx resp IRQ response 9: Rx resp tx bit 10: Rx resp cmd idx 11: Rx resp data 12: Rx resp crc7 13: Rx resp end bit 14: Cmd path wait NCC 15: Wait; CMD-to-response turnaround The command FSM state is represented using 19 bits. The STATUS Register[7:4] has 4 bits to represent the command FSM states. Using these 4 bits, only 16 states can be represented. Thus three states cannot be represented in the STATUS[7:4] register. The three states that are not represented in the STATUS Register[7:4] are: * Bit 16 -Wait for CCS * Bit 17 -Send CCSD * Bit 18 -Boot Mode Due to this, while command FSM is in "Wait for CCS state" or "Send CCSD" or "Boot Mode", the Status register indicates status as 0 for the bit field [7:4].
3	RO	0x0	fifo_full FIFO is full status
2	RO	0x1	fifo_empty FIFO is empty status
1	RO	0x1	fifo_tx_watermark FIFO reached Transmit watermark level; not qualified with data transfer

Bit	Attr	Reset Value	Description
0	RO	0x0	fifo_rx_watermark FIFO reached Receive watermark level; not qualified with data transfer

SDMMC_FIFOTH

Address: Operational Base + offset (0x0004c)

FIFO threshold register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:28	RW	0x0	<p>dma_multiple_transaction_size Burst size of multiple transaction; should be programmed same as DMA controller multiple-transaction-size SRC/DEST_MSIZE.</p> <p>3'b000: 1 transfers 3'b001: 4 3'b010: 8 3'b011: 16 3'b100: 32 3'b101: 64 3'b110: 128 3'b111: 256</p> <p>The unit for transfer is the H_DATA_WIDTH parameter. A single transfer (dw_dma_single assertion in case of Non DW DMA interface) would be signalled based on this value.</p> <p>Value should be sub-multiple of (RX_WMark + 1)* (F_DATA_WIDTH/H_DATA_WIDTH) and (FIFO_DEPTH - TX_WMark)* (F_DATA_WIDTH/ H_DATA_WIDTH)</p> <p>For example, if FIFO_DEPTH = 16, FDATA_WIDTH == H_DATA_WIDTH</p> <p>Allowed combinations for MSize and TX_WMark are:</p> <ul style="list-style-type: none"> MSize = 1, TX_WMARK = 1-15 MSize = 4, TX_WMark = 8 MSize = 4, TX_WMark = 4 MSize = 4, TX_WMark = 12 MSize = 8, TX_WMark = 8 MSize = 8, TX_WMark = 4 <p>Allowed combinations for MSize and RX_WMark are:</p> <ul style="list-style-type: none"> MSize = 1, RX_WMARK = 0-14 MSize = 4, RX_WMark = 3 MSize = 4, RX_WMark = 7 MSize = 4, RX_WMark = 11 MSize = 8, RX_WMark = 7 <p>Recommended:</p> <ul style="list-style-type: none"> MSize = 8, TX_WMark = 8, RX_WMark = 7

Bit	Attr	Reset Value	Description
27:16	RW	0x000	<p>rx_wmark FIFO threshold watermark level when receiving data to card. When FIFO data count reaches greater than this number, DMA/FIFO request is raised. During end of packet, request is generated regardless of threshold programming in order to complete any remaining data. In non-DMA mode, when receiver FIFO threshold (RXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, interrupt is not generated if threshold programming is larger than any remaining data. It is responsibility of host to read remaining bytes on seeing Data Transfer Done interrupt. In DMA mode, at end of packet, even if remaining bytes are less than threshold, DMA request does single transfers to flush out any remaining bytes before Data Transfer Done interrupt is set. 12 bits-1 bit less than FIFO-count of status register, which is 13 bits. Limitation: RX_WMark <= FIFO_DEPTH-2 Recommended: (FIFO_DEPTH/2) - 1; (means greater than (FIFO_DEPTH/2) - 1) NOTE: In DMA mode during CCS time-out, the DMA does not generate the request at the end of packet, even if remaining bytes are less than threshold. In this case, there will be some data left in the FIFO. It is the responsibility of the application to reset the FIFO after the CCS timeout.</p>
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	<p>tx_wmark FIFO threshold watermark level when transmitting data to card. When FIFO data count is less than or equal to this number, DMA/FIFO request is raised. If Interrupt is enabled, then interrupt occurs. During end of packet, request or interrupt is generated, regardless of threshold programming.</p> <p>In non-DMA mode, when transmit FIFO threshold (TXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, on last interrupt, host is responsible for filling FIFO with only required remaining bytes (not before FIFO is full or after CIU completes data transfers, because FIFO may not be empty).</p> <p>In DMA mode, at end of packet, if last transfer is less than burst size, DMA controller does single cycles until required bytes are transferred.</p> <p>12 bits -1 bit less than FIFO-count of status register, which is 13 bits. Limitation: TX_WMark >= 1; Recommended: FIFO_DEPTH/2; (means less than or equal to FIFO_DEPTH/2)</p>

SDMMC_CDETECT

Address: Operational Base + offset (0x00050)

Card-detect register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	card_detect_n Value on card_detect_n input ports; read-only bits. 0 represents presence of card.

SDMMC_WRTPRT

Address: Operational Base + offset (0x00054)

Write-protect register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	write_protect Value on card_write_prt input port. 1 represents write protection.

SDMMC_TCBCNT

Address: Operational Base + offset (0x0005c)

Transferred CIU card byte count

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>trans_card_byte_count Number of bytes transferred by CIU unit to card.</p> <p>In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, internal 16-bit coherency register is implemented. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, higher 16 bits of counter are stored in temporary register. When higher 16 bits are read, data from temporary register is supplied.</p> <p>Both TCBCNT and TBBCNT share same coherency register.</p> <p>When AREA_OPTIMIZED parameter is 1, register should be read only after data transfer completes; during data transfer, register returns 0.</p>

SDMMC_TBBCNT

Address: Operational Base + offset (0x00060)

Transferred host/DMA to/from BIU-FIFO byte count

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>trans_fifo_byte_count Number of bytes transferred between Host/DMA memory and BIU FIFO.</p> <p>In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, internal 16-bit coherency register is implemented. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, higher 16 bits of counter are stored in temporary register. When higher 16 bits are read, data from temporary register is supplied.</p> <p>Both TCBCNT and TBBCNT share same coherency register.</p>

SDMMC_DEBNCE

Address: Operational Base + offset (0x00064)

Card detect debounce register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0xfffffff	<p>debounce_count Number of host clocks (clk) used by debounce filter logic; typical debounce time is 5-25 ms.</p>

SDMMC_USRID

Address: Operational Base + offset (0x00068)

User ID register

Bit	Attr	Reset Value	Description
31:0	RW	0x07967797	usrid User identification register; value set by user. Default reset value can be picked by user while configuring core before synthesis. Can also be used as scratch pad register by user. The default value is determined by Configuration Value.

SDMMC_VRID

Address: Operational Base + offset (0x0006c)

Synopsys version ID register

Bit	Attr	Reset Value	Description
31:0	RO	0x5342270a	verid Version identification register; register value is hard-wired. Can be read by firmware to support different versions of core.

SDMMC_HCON

Address: Operational Base + offset (0x00070)

Hardware Configuration Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>HCON Configuration Dependent. Hardware configurations selected by user before synthesizing core. Register values can be used to develop configuration-independent software drivers.</p> <ul style="list-style-type: none"> [0]: CARD_TYPE <ul style="list-style-type: none"> 0: MMC_ONLY 1: SD_MMC [5:1]: NUM_CARDS - 1 [6]: H_BUS_TYPE <ul style="list-style-type: none"> 0: APB 1: AHB [9:7]: H_DATA_WIDTH <ul style="list-style-type: none"> 000: 16 bits 001: 32 bits 010: 64 bits others: reserved [15:10]: H_ADDR_WIDTH <ul style="list-style-type: none"> 0 to 7: reserved 8: 9 bits 9: 10 bits ... 31: 32 bits 32 to 63: reserved <p>[17:16]: DMA_INTERFACE <ul style="list-style-type: none"> 00: none 01: DW_DMA 10: GENERIC_DMA 11: NON-DW-DMA </p> <p>[20:18]: GE_DMA_DATA_WIDTH <ul style="list-style-type: none"> 000: 16 bits 001: 32 bits 010: 64 bits others: reserved </p> <p>[21]: FIFO_RAM_INSIDE <ul style="list-style-type: none"> 0: outside 1: inside </p> <p>[22]: IMPLEMENT_HOLD_REG <ul style="list-style-type: none"> 0: no hold register 1: hold register </p> <p>[23]: SET_CLK_FALSE_PATH <ul style="list-style-type: none"> 0: no false path 1: false path set </p> <p>[25:24]: NUM_CLK_DIVIDER-1</p> <p>[26]: AREA_OPTIMIZED <ul style="list-style-type: none"> 0: no area optimization 1: Area optimization </p>

SDMMC_UHS_REG

Address: Operational Base + offset (0x00074)

UHS-1 register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	ddr_reg DDR mode. Determines the voltage fed to the buffers by an external voltage regulator. 0: non-DDR mode 1: DDR mode UHS_REG [16] should be set for card.
15:1	RO	0x0	reserved
0	RW	0x0	volt_reg High Voltage mode. Determines the voltage fed to the buffers by an external voltage regulator. 0: buffers supplied with 3.3V Vdd 1: buffers supplied with 1.8V Vdd These bits function as the output of the host controller and are fed to an external voltage regulator. The voltage regulator must switch the voltage of the buffers of a particular card to either 3.3V or 1.8V, depending on the value programmed in the register. VOLT_REG[0] should be set to 1'b1 for card in order to make it operate for 1.8V.

SDMMC_RST_n

Address: Operational Base + offset (0x00078)

Hardware reset register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	card_reset Hardware reset. 0: active mode 1: reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized. CARD_RESET[0] should be set to 1'b1 to reset card.

SDMMC_PLDMND

Address: Operational Base + offset (0x00084)

Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	PD Poll Demand. If the OWN bit of a descriptor is not set, the FSM goes to the Suspend state. The host needs to write any value into this register for the IDMAC FSM to resume normal descriptor fetch operation. This is a write only register.

SDMMC_CARDTHRCTL

Address: Operational Base + offset (0x00100)

Card read threshold enable

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	CardRdThreshold Card Read Threshold size
15:2	RO	0x0	reserved
1	RW	0x0	BsyClrIntEn Busy Clear Interrupt generation: 0: Busy Clear Interrupt disabled 1: Busy Clear Interrupt enabled Note: The application can disable this feature if it does not want to wait for a Busy Clear Interrupt. For example, in a multi-card scenario, the application can switch to the other card without waiting for a busy to be completed. In such cases, the application can use the polling method to determine the status of busy. By default this feature is disabled and backward-compatible to the legacy drivers where polling is used.
0	RW	0x0	CardRdThrEn Card Read Threshold Enable. 0: Card Read Threshold disabled 1: Card Read Threshold enabled. Host Controller initiates Read Transfer only if CardRdThreshold amount of space is available in receive FIFO.

SDMMC_BACK_END_POWER

Address: Operational Base + offset (0x00104)

Back-end power

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	back_end_power Back end power 0: Off; Reset 1: Back-end Power supplied to card application

SDMMC_UHS_REG_EXT

Address: Operational Base + offset (0x00108)

UHS Register

Bit	Attr	Reset Value	Description
31:30	RW	0x0	EXT_CLK_MUX_CTRL Input clock control for cclk_in. The MUX controlled by these bits exists outside DWC_mobile_storage IP

Bit	Attr	Reset Value	Description
29:23	RW	0x00	CLK_DRV_PHASE_CTRL Control for amount of phase shift on cclk_in_drv clock. Can choose three MSBs to control delay lines and four LSBs to control phase shift; alternatively, use only LSBs
22:16	RW	0x00	CLK_SMPL_PHASE_CTRL Control for amount of phase shift on cclk_in_sample clock. Can choose three MSBs to control delay lines and four LSBs to control phase shift; alternatively, use only LSBs
15:0	RW	0x0000	MMC_VOLT_REG Support for 1.2V. MMC_VOLT_REG bits; must be read in combination with UHS_VOLT_REG to decode output selected voltage. The biu_volt_reg_1_2[NUM_CARD_BUS-1:0] signal decodes the voltage combination selected for the I/O voltage logic. Host controllers that support only SD standard or standard versions before eMMC4.41 do not program MMC_VOLT_REG. Only host controllers that support all three versions (1.8, 1.2 V) can program MMC_VOLT_REG and connect biu_volt_reg_1_2

SDMMC_EMMC_DDR_REG

Address: Operational Base + offset (0x0010c)

eMMC 4.5 DDR START Bit Detection Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	HALF_START_BIT Control for start bit detection mechanism inside DWC_mobile_storage based on duration of start bit; each bit refers to one slot. For eMMC 4.5, start bit can be: 0: Full cycle (HALF_START_BIT = 0) 1: Less than one full cycle (HALF_START_BIT = 1) Set HALF_START_BIT=1 for eMMC 4.5 and above; set to 0 for SD applications.

SDMMC_ENABLE_SHIFT

Address: Operational Base + offset (0x00110)

Enable Phase Shift Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>enable_shift Control for the amount of phase shift provided on the default enables in the design. Two bits are assigned for each card/slot. For example, bits[1:0] control slot0 and indicate the following.</p> <ul style="list-style-type: none"> 00: Default phase shift 01: Enables shifted to next immediate positive edge 10: Enables shifted to next immediate negative edge 11: Reserved

SDMMC_FIFO_BASE

Address: Operational Base + offset (0x00200)

FIFO Base Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>fifo_base_addr fifo base addr</p>

13.5 Interface Description

The interface and IOMUX setting for SDMMC, SDIO, EMMC are shown as follows.

13.5.1 SDMMC IOMUX

Table 13-4 IOMUX Settings for SDMMC

Module Pin	Direction	Pad Name	IOMUX Setting
sdmmc_cclk	O	IO_MMC0clkout_GPIO1c0	GRF_GPIO1C_IOMUX[1:0]=2'b01
sdmmc_ccmd	I/O	IO_MMC0cmd_GPIO1b7	GRF_GPIO1B_IOMUX[15:14]=2'b01
sdmmc_cdata0	I/O	IO_MMC0d0_UART2tx_GPIO1c2	GRF_GPIO1C_IOMUX[5:4]=2'b01
sdmmc_cdata1	I/O	IO_MMC0d1_UART2rx_GPIO1c3	GRF_GPIO1C_IOMUX[7:6]=2'b01
sdmmc_cdata2	I/O	IO_MMC0d2_JTAGtck1_GPIO1c4	GRF_GPIO1C_IOMUX[9:8]=2'b01
sdmmc_cdata3	I/O	IO_MMC0d3_JTAGtms1_GPIO1c5	GRF_GPIO1C_IOMUX[11:10]=2'b01
sdmmc_cdetect_n	I	IO_MMC0detn_GPIO1c1	GRF_GPIO1C_IOMUX[3:2]=2'b01
sdmmc_wpri	I	IO_MMC0wrprt_GPIO1a7	GRF_GPIO1A_IOMUX[15:14]=2'b01
sdmmc_pwren	O	IO_MMC0pwren_GPIO1b6	GRF_GPIO1B_IOMUX[13:12]=2'b01

13.5.2 SDIO IOMUX

Table 13-5 IOMUX Settings for SDIO

Module Pin	Direction	Pad Name	IOMUX Setting
sdio_cclk	O	IO_I2Sclk_MMC1clkout_XIN32k_GPIO1a0	GRF_GPIO1A_IOMUX[1:0]=2'b10
sdio_ccmd	I/O	IO_I2C1tpsda_MMC1cmd_GPIO0a3	GRF_GPIO1A_IOMUX[7:6]=2'b10
sdio_cdata0	I/O	IO_I2Ssclk_MMC1d0_PMICsleep1_GPIO1a1	GRF_GPIO1A_IOMUX[3:2]=2'b10
sdio_cdata1	I/O	IO_I2Slrckrx_MMC1d1_GPIO1a2	GRF_GPIO1A_IOMUX[5:4]=2'b10
sdio_cdata2	I/O	IO_I2Ssdo_MMC1d2_GPIO1a4	GRF_GPIO1A_IOMUX[9:8]=2'b10
sdio_cdata3	I/O	IO_I2Ssdi_MMC1d3_GPIO1a5	GRF_GPIO1A_IOMUX[11:10]=2'b10
sdio_pwren	O	IO_MMCI1pwren_GPIO0d6	GRF_GPIO0D_IOMUX[13:12]=2'b01

13.5.3 eMMC IOMUX

Table 13-6 IOMUX Settings for eMMC

Module Pin	Direction	Pad Name	IOMUX Setting
emmc_cclk	O	IO_NANDdqs_EMMCclkout_GPIO2_a7	GRF_GPIO2A_IOMUX[15:14]=2'b10
emmc_ccmd	I/O	IO_NANDcs2_EMMCCmd_GPIO1c6	GRF_GPIO1C_IOMUX[13:12]=2'b10 & GRF_SOC_CON1[6]=1'b0
		IO_NANDrdy_EMMCCmd1_SFClk_GPIO2a4	GRF_GPIO2A_IOMUX[9:8]=2'b10 & GRF_SOC_CON1[6]=1'b1
emmc_cdata0	I/O	IO_NANDd0_EMMCd0_SFcd0_GPI_O1d0	GRF_GPIO1D_IOMUX[1:0]=2'b10
emmc_cdata1	I/O	IO_NANDd1_EMMCd1_SFcd1_GPI_O1d1	GRF_GPIO1D_IOMUX[3:2]=2'b10
emmc_cdata2	I/O	IO_NANDd2_EMMCd2_SFcd2_GPI_O1d2	GRF_GPIO1D_IOMUX[5:4]=2'b10
emmc_cdata3	I/O	IO_NANDd3_EMMCd3_SFcd3_GPI_O1d3	GRF_GPIO1D_IOMUX[7:6]=2'b10
emmc_cdata4	I/O	IO_NANDd4_EMMCd4_SPI1rxdi_GPIO1d4	GRF_GPIO1D_IOMUX[9:8]=2'b10
emmc_cdata5	I/O	IO_NANDd5_EMMCd5_SPI1txdi_GPIO1d5	GRF_GPIO1D_IOMUX[11:10]=2'b10
emmc_cdata6	I/O	IO_NANDd6_EMMCd6_SPI1csn0_GPIO1d6	GRF_GPIO1D_IOMUX[13:12]=2'b10
emmc_cdata7	I/O	IO_NANDd7_EMMCd7_SPI1csn1_GPIO1d7	GRF_GPIO1D_IOMUX[15:14]=2'b10
emmc_rstn	O	IO_NANDcs3_EMMCrstnout_GPIO1c7	GRF_GPIO1C_IOMUX[15:14]=2'b10
emmc_pwren	O	IO_NANDwp_EMMCPwren_GPIO2a_5	GRF_GPIO2A_IOMUX[11:10]=2'b10

Notes: Direction: **I**- Input, **O**- Output, **I/O**- Input/Output

13.6 Application Notes

13.6.1 Card-Detect and Write-Protect Mechanism

Following figure illustrates how the SD/MMC card detection and write-protect signals are connected. Most of the SD/MMC sockets have card-detect pins. When no card is present, card_detect_n is 1 due to the pull-up. When the card is inserted, the card-detect pin is shorted to ground, which makes card_detect_n go to 0. Similarly in SD cards, when the write-protect switch is toward the left, it shorts the write_protect port to ground.

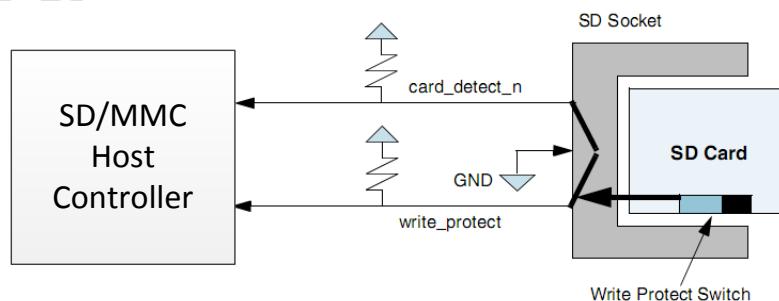


Fig. 13-6 SD/MMC Card-Detect and Write-Protect

13.6.2 SD/MMC Termination Requirement

Following Figure illustrates the SD/MMC termination requirements, which is required to pull up ccmd and cdata lines on the device bus. The recommended specification for pull-up on the ccmd line (Rcmd) is 4.7K - 100K for MMC, and 10K - 100K for an SD. The recommended pull-up on the cdata line (Rdat) is 50K - 100K.

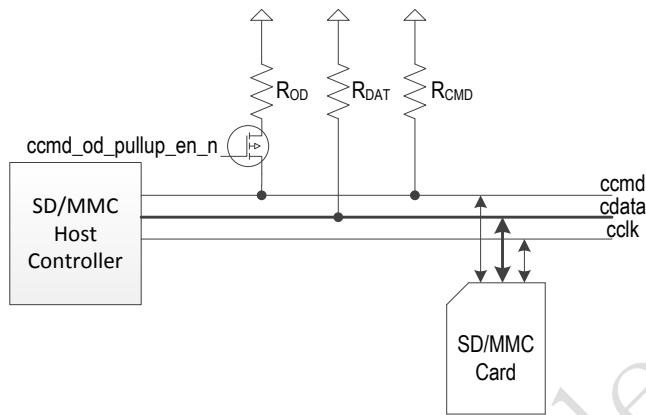


Fig. 13-7 SD/MMC Card Termination

1. Rcmd and Rod Calculation

The SD/MMC card enumeration happens at a very low frequency – 100-400KHz. Since the MMC bus is a shared bus between multiple cards, during enumeration open-drive mode is used to avoid bus conflict. Cards that drive 0 win over cards that drive “z”. The pull-up in the command line pulls the bus to 1 when all cards drive “z”. During normal data transfer, the host chooses only one card and the card driver switches to push-pull mode.

For example, if enumeration is done at 400KHz and the total bus capacitance is 200 pf, the pull-up needed during enumeration is:

$$\begin{aligned} 2.2 \text{ RC} &= \text{rise-time} = 1/400\text{KHz} \\ R &= 1/(2.2 * C * 100\text{KHz}) \\ &= 1/(2.2 \times 200 \times 10^{**-12} \times 400 \times 10^{**3}) \\ &= 1/(17.6 \times 10^{**-5}) \\ &= 5.68\text{K} \end{aligned}$$

The R_{OD} and R_{CMD} should be adjusted in such a way that the effective pull-up is at the maximum 5.68K during enumeration. If there are only a few cards in the bus, a fixed R_{CMD} resistor is sufficient and there is no need for an additional R_{OD} pull-up during enumeration. You should also ensure the effective pull-up will not violate the I_{OL} rating of the drivers.

In SD mode, since each card has a separate bus, the capacitance is less, typically in the order of 20-30pf (host capacitance + card capacitance + trace + socket capacitance). For example, if enumeration is done at 400KHz and the total bus capacitance is 20pf, the pull-up needed during enumeration is:

$$\begin{aligned} 2.2 \text{ RC} &= \text{rise-time} = 1/400\text{KHz} \\ R &= 1/(2.2 * C * 100\text{KHz}) \\ &= 1/(2.2 \times 20 \times 10^{**-12} \times 400 \times 10^{**3}) \end{aligned}$$

$$\begin{aligned}
 &= 1/(1.76 \times 10^{**-5}) \\
 &= 56.8K
 \end{aligned}$$

Therefore, a fixed 56.8K permanent Rcmd is sufficient in SD mode to enumerate the cards.

The driver of the SD/MMC on the “command” port needs to be only a push-pull driver. During enumeration, the SD/MMC emulates an open-drain driver by driving only a 0 or a “z” by controlling the ccmd_out and ccmd_out_en signals.

13.6.3 Software/Hardware Restriction

Before issuing a new data transfer command, the software should ensure that the card is not busy due to any previous data transfer command. Before changing the card clock frequency, the software must ensure that there are no data or command transfers in progress.

If the card is enumerated in SDR50, or DDR50 mode, then the application must program the use_hold_reg bit[29] in the CMD register to 1'b0 (phase shift of cclk_in_drv = 0) or 1'b1 (phase shift of cclk_in_drv>0). If the card is enumerated in SDR12 or SDR25 mode, the application must program the use_hold_reg bit[29] in the CMD register to 1'b1.

This programming should be done for all data transfer commands and non-data commands that are sent to the card. When the use_hold_reg bit is programmed to 1'b0, the Host Controller bypasses the Hold Registers in the transmit path. The value of this bit should not be changed when a Command or Data Transfer is in progress. For more details on using use_hold_reg and the implementation requirements for meeting the Card input hold time, refer to “Recommended Usage” in following table.

Table 13-7 Recommended Usage of use_hold_reg

No.	Speed Mode	use_hold_reg	cclk_in (MHz)	clk_in_drv (MHz)	clk_divider	Phase shift
1	SDR104	1'b0	200	200	0	0
2	SDR104	1'b1	200	200	0	Tunable> 0
3	SDR50	1'b0	100	100	0	0
4	SDR50	1'b1	100	100	0	Tunable> 0
5	DDR50 (8bit)	1'b0	100	100	1	0
6	DDR50 (8bit)	1'b1	100	100	1	Tunable> 0
7	DDR50 (4bit)	1'b0	50	50	0	0
8	DDR50 (4bit)	1'b1	50	50	0	Tunable> 0
9	SDR25	1'b1	50	50	0	Tunable> 0
10	SDR12	1'b1	50	50	1	Tunable> 0

To avoid glitches in the card clock outputs, the software should use the following steps when changing the card clock frequency:

- 1) Before disable the clocks, ensure that the card is not busy due to any previous data command. To determine this, check for 0 in bit9 of STATUS register.

2) Update the Clock Enable register to disable all clocks. To ensure completion of any previous command before this update, send a command to the CIU to update the clock registers by setting:

- start_cmd bit
- “update clock registers only” bits
- “wait_previous data complete” bit

Wait for the CIU to take the command by polling for 0 on the start_cmd bit.

- 3) Set the start_cmd bit to update the Clock Divider and/or Clock Source registers, and send a command to the CIU in order to update the clock registers; wait for the CIU to take the command.
- 4) Set start_cmd to update the Clock Enable register in order to enable the required clocks and send a command to the CIU to update the clock registers; wait for the CIU to take the command.

In non-DMA mode, while reading from a card, the Data Transfer Over (RINTSTS[3]) interrupt occurs as soon as the data transfer from the card is over. There still could be some data left in the FIFO, and the RX_WMark interrupt may or may not occur, depending on the remaining bytes in the FIFO. Software should read any remaining bytes upon seeing the Data Transfer Over (DTO) interrupt. While using the external DMA interface for reading from a card, the DTO interrupt occurs only after all the data is flushed to memory by the DMA interface unit.

While writing to a card in external DMA mode, if an undefined-length transfer is selected by setting the Byte Count Register to 0, the DMA logic will likely request more data than it will send to the card, since it has no way of knowing at which point the software will stop the transfer. The DMA request stops as soon as the DTO is set by the CIU.

If the software issues a controller_reset command by setting control register bit[0] to 1, all the CIU state machines are reset; the FIFO is not cleared. The DMA sends all remaining bytes to the host. In addition to a card-reset, if a FIFO reset is also issued, then:

- Any pending DMA transfer on the bus completes correctly
- DMA data read is ignored
- Write data is unknown(x)

Additionally, if dma_reset is also issued, any pending DMA transfer is abruptly terminated. When the DW-DMA is used, the DMA controller channel should also be reset and reprogrammed.

If any of the previous data commands do not properly terminate, then the software should issue the FIFO reset in order to remove any residual data, if any, in the FIFO. After asserting the FIFO reset, you should wait until this bit is cleared.

One data-transfer requirement between the FIFO and host is that the number of transfers should be a multiple of the FIFO data width (32bits). For example, you want to write only 15 bytes to an SD/MMC card (BYTCNT), the host should write 16 bytes to the FIFO or program the DMA to do 16-byte transfers. The software can still program the Byte Count register to only 15, at which point only 15 bytes will be transferred to the card. Similarly, when 15 bytes are read from a card, the host should still read all 16 bytes from the FIFO.

It is recommended that you not change the FIFO threshold register in the middle of data transfers.

13.6.4 Programming Sequence

1. Initialization

Following figure illustrates the initialization flow.

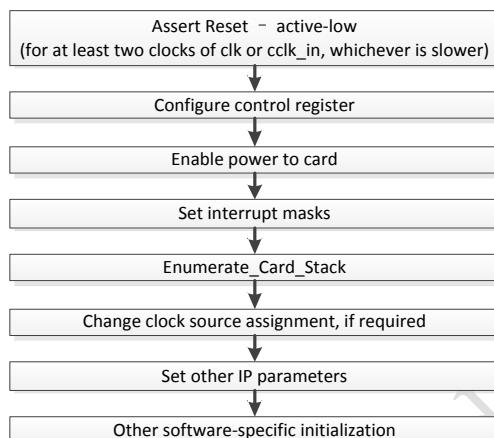


Fig. 13-8 Host Controller Initialization Sequence

Once the power and clocks are stable, `reset_n` should be asserted(active-low) for at least two clocks of `clk` or `cclk_in`, whichever is slower. The reset initializes the registers, ports, FIFO-pointers, DMA interface controls, and state-machines in the design. After power-on reset, the software should do the following:

- 1) Configure control register – For MMC mode, enable the open-drain pullup by setting `enable_OD_pullup`(bit24) in the control register.
- 2) Enable power to cards – Before enabling the power, confirm that the voltage setting to the voltage regulators is correct. Enable power to the connected cards by setting the corresponding bit to 1 in the Power Enable register. Wait for the power ramp-up time.
- 3) Set masks for interrupts by clearing appropriate bits in the Interrupt Mask register. Set the global `int_enable` bit of the Control register. It is recommended that you write `0xffff_ffff` to the Raw Interrupt register in order to clear any pending interrupts before setting the `int_enable` bit.
- 4) Enumerate card stack – Each card is enumerated according to card type; for details, refer to “Enumerated Card Stack”. For enumeration, you should restrict the clock frequency to 400KHz.
- 5) Changing clock source assignment – set the card frequency using the clock-divider and clock-source registers; for details, refer to “Clock Programming”. MMC cards operate at a maximum of 20MHz (at maximum of 52MHz in high-speed mode). SD mode operates at a maximum of 25MHz (at maximum of 50MHz in high-speed mode).
- 6) Set other parameters, which normally do not need to be changed with every command, with a typical value such as timeout values in `cclk_out` according to SD/MMC specifications.
 - `ResponseTimeOut` = `0x64`
 - `DataTimeOut` = highest of one of the following:

$(10*((TAAC*Fop)+(100*NSAC))$

Host FIFO read/write latency from FIFO empty/full

- Set the debounce value to 25ms(default:0xffff) in host clock cycle units in the DEBNCE register.
- FIFO threshold value in bytes in the FIFOTH register.

2. Enumerated Card Stack

The card stack does the following:

- Enumerates all connected cards
- Sets the RCA for the connected cards
- Reads card-specific information
- Stores card-specific information locally

Enumeration depends on the operating mode of the SD/MMC card; the card type is first identified and the appropriate card enumeration routine is called.

- 1) Check if the card is connected.
- 2) Clear the card type register to set the card width as a single bit. For the given card number, clear the corresponding bits in the card_type register. Clear the register bit for a 1-bit, 4-bit bus width. For example, for card number=1, clear bit 0 and bit 16 of the card_type register.
- 3) Set clock frequency to $F_{OD}=400\text{KHz}$, maximum – Program clock divider0 (bits 0-7 in the CLKDIV register) value to one-half of the cclk_in frequency divided by 400KHz. For example, if cclk_in is 20MHz, then the value is $20,000/(2*400)=25$.
- 4) Identify the card type; that is, SD, MMC, or SDIO.
 - a. Send CMD5 first. If a response is received, then the card is SDIO
 - b. If not, send CMD8 with the following Argument


```
Bit[31:12] = 20'h0 //reserved bits
          Bit[11:8] = 4'b0001 //VHS value
          Bit[7:0] = 8'b10101010 //Preferred Check Pattern by SD2.0
```
 - c. If Response is received the card supports High Capacity SD2.0 then send ACMD41 with the following Argument


```
Bit[31] = 1'b0; //Reserved bits
          Bit[30] = 1'b1; //High Capacity Status
          Bit[29:24] = 6'h0; //Reserved bits
          Bit[23:0] = Supported Voltage Range
```
 - d. If Response is received for ACMD41 then the card is SD. Otherwise the card is MMC.
 - e. If response is not received for initial CMD8 then card does not support High Capacity SD2.0, then issue CMD0 followed by ACMD41 with the following Argument


```
Bit[31] = 1'b0; //Reserved bits
          Bit[30] = 1'b0; //High Capacity Status
          Bit[29:24] = 6'h0; //Reserved bits
          Bit[23:0] = Supported Voltage Range
```

- 5) Enumerate the card according to the card type.
- 6) Use a clock source with a frequency = Fod (that is, 400KHz) and use the following enumeration command sequence:
 - SD card – Send CMD0, CMD8, ACMD41, CMD2, CMD3.
 - MMC – Send CMD0, CMD1, CMD2, CMD3.

3. Power Control

You can implement power control using the following registers, along with external circuitry:

- Control register bits card_voltage_a and card_voltage_b – Status of these bits is reflected at the IO pins. The bits can be used to generate or control the supply voltage that the memory cards require.
- Power enable register – Control power to individual cards.

Programming these two register depends on the implemented external circuitry. While turning on or off the power enable, you should confirm that power supply settings are correct. Power to all cards usually should be disabled while switching off the power.

4. Clock Programming

The Host Controller supports one clock sources. The clock to an individual card can be enabled or disabled. Registers that support this are:

- CLKDIV – Programs individual clock source frequency. CLKDIV limited to 0 or 1 is recommended.
- CLKSRC – Assign clock source for each card.
- CLKENA – Enables or disables clock for individual card and enables low-power mode, which automatically stops the clock to a card when the card is idle for more than 8 clocks.

The Host Controller loads each of these registers only when the start_cmd bit and the Update_clk_regs_only bit in the CMD register are set. When a command is successfully loaded, the Host Controller clears this bit, unless the Host Controller already has another command in the queue, at which point it gives an HLE(Hardware Locked Error).

Software should look for the start_cmd and the Update_clk_regs_only bits, and should also set the wait_prvdata_complete bit to ensure that clock parameters do not change during data transfer. Note that even though start_cmd is set for updating clock registers, the Host Controller does not raise a command_done signal upon command completion.

The following shows how to program these registers:

- 1) Confirm that no card is engaged in any transaction; if there is a transaction, wait until it finishes.
- 2) Stop all clocks by writing xxxx0000 to the CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.

- 3) Program the CLKDIV and CLKSRC registers, as required. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
- 4) Re-enable all clocks by programming the CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.

5. No-Data Command With or Without Response Sequence

To send any non-data command, the software needs to program the CMD register @0x2C and the CMDARG register @0x28 with appropriate parameters. Using these two registers, the Host Controller forms the command and sends it to the command bus. The Host Controller reflects the errors in the command response through the error bits of the RINTSTS register.

When a response is received – either erroneous or valid – the Host Controller sets the command_done bit in the RINTSTS register. A short response is copied in Response Register0, while along response is copied to all four response registers @0x30, 0x34, 0x38, and 0x3C. The Response3 register bit 31 represents the MSB, and the Response0 register bit 0 represents the LSB of a long response.

For basic commands or non-data commands, follow these steps:

- 1) Program the Command register @0x28 with the appropriate command argument parameter.
- 2) Program the Command register @0x2C with the settings in following table.

Table 13-8 Command Settings for No-Data Command

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used; ref to "use_hold_reg" on CMD register
update_clk_regs_only	0	No clock parameters update command
data_expected	0	No data command
card number	0	Actual card number(one controller only connect one card, the num is No. 0)
cmd_index	command-index	-
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
wait_prvdata_complete	1	Before sending command on command line, host should wait for completion of any data command in process, if any (recommended to always set this bit, unless the current command is to query status or stop data transfer when transfer is in progress)
check_response_crc	1	If host should crosscheck CRC of response received

- 3) Wait for command acceptance by host. The following happens when the command is loaded into the Host Controller:
 - Host Controller accepts the command for execution and clears the start_cmd bit in the CMD register, unless one command is in process, at which point the Host Controller can load and keep the second command in the buffer.
 - If the Host Controller is unable to load the command – that is, a command is already in progress, a second command is in the buffer, and a third command is attempted – then it generates an HLE (hardware-locked error).
- 4) Check if there is an HLE.
- 5) Wait for command execution to complete. After receiving either a response from a card or response timeout, the Host Controller sets the command_done bit in the RINTSTS register. Software can either poll for this bit or respond to a generated interrupt.
- 6) Check if response_timeout error, response_CRC error, or response error is set. This can be done either by responding to an interrupt raised by these errors or by polling bits 1, 6, and 8 from the RINTSTS register @0x44. If no response error is received, then the response is valid. If required, the software can copy the response from the response registers @0x30-0x3C.

Software should not modify clock parameters while a command is being executed.

6. Data Transfer Commands

Data transfer commands transfer data between the memory card and the Host Controller. To send a data command, the Host Controller needs a command argument, total data size, and block size. Software can receive or send data through the FIFO.

Before a data transfer command, software should confirm that the card is not busy and is in a transfer state, which can be done using the CMD13 and CMD7 commands, respectively.

For the data transfer commands, it is important that the same bus width that is programmed in the card should be set in the card type register @0x18.

The Host Controller generates an interrupt for different conditions during data transfer, which are reflected in the RINTSTS register @0x44 as:

- 1) Data_Transfer_Over (bit 3) – When data transfer is over or terminated. If there is a response timeout error, then the Host Controller does not attempt any data transfer and the “Data Transfer Over” bit is never set.
- 2) Transmit_FIFO_Data_request (bit 4) – FIFO threshold for transmitting data was reached; software is expected to write data, if available, in FIFO.
- 3) Receive_FIFO_Data_request (bit 5) – FIFO threshold for receiving data was reached; software is expected to read data from FIFO.
- 4) Data starvation by Host timeout (bit 10) – FIFO is empty during transmission or is full during reception. Unless software writes data for empty condition or reads data for full condition, the Host Controller cannot continue with data transfer. The clock to the card has been stopped.

- 5) Data read timeout error (bit 9) – Card has not sent data within the timeout period.
- 6) Data CRC error (bit 7) – CRC error occurred during data reception.
- 7) Start bit error (bit 13) – Start bit was not received during data reception.
- 8) End bit error (bit 15) – End bit was not received during data reception or for a write operation; a CRC error is indicated by the card.

Conditions 6, 7, and 8 indicate that the received data may have errors. If there was a response timeout, then no data transfer occurred.

7. Single-Block or Multiple-Block Read

Steps involved in a single-block or multiple-block read are:

- 1) Write the data size in bytes in the BYTCNT register @0x20.
- 2) Write the block size in bytes in the BLKSIZ register @0x1C. The Host Controller expects data from the card in blocks of size BLKSIZ each.
- 3) Program the CMDARG register @0x28 with the data address of the beginning of a data read.
- 4) Program the Command register with the parameters listed in following table. For SD and MMC cards, use CMD17 for a single-block read and CMD18 for a multiple-block read. For SDIO cards, use CMD53 for both single-block and multiple-block transfers.

Table 13-9 Command Setting for Single or Multiple-Block Read

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used; ref to "use_hold_reg" on CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number (one controller only connect one card, the num is No.0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	0	Read from card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0- Sends command immediately 1- Sends command after previous data transfer ends
check_response_crc	1	0- Host Controller should not check response CRC 1- Host Controller should check response CRC

After writing to the CMD register, the Host Controller starts executing the command; when the command is sent to the bus, the command_done interrupt is generated.

- Software should look for data error interrupts; that is, bits 7, 9, 13, and 15 of the RINTSTS register. If required, software can terminate the data transfer by sending a STOP command.
- Software should look for Receive_FIFO_Data_request and/or data starvation by host timeout conditions. In both cases, the software should read data from the FIFO and make space in the FIFO for receiving more data.
- When a Data_Transfer_Over interrupt is received, the software should read the remaining data from the FIFO.

8. Single-Block or Multiple-Block Write

Steps involved in a single-block or multiple-block write are:

- 1) Write the data size in bytes in the BYTCNT register @0x20.
- 2) Write the block size in bytes in the BLKSIZ register @0x1C; the Host Controller sends data in blocks of size BLKSIZ each.
- 3) Program CMDARG register @0x28 with the data address to which data should be written.
- 4) Write data in the FIFO; it is usually best to start filling data the full depth of the FIFO.
- 5) Program the Command register with the parameters listed in following table.

Table 13-10 Command Settings for Single or Multiple-Block Write

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used; ref to "use_hold_reg" on CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number (one controller only connect one card, the num is No. 0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	1	Write to card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0- Sends command immediately 1- Sends command after previous data transfer ends
check_response_crc	1	0- Host Controller should not check response CRC 1- Host Controller should check response CRC

After writing to the CMD register, Host Controller starts executing a command; when the command is sent to the bus, a command_done interrupt is generated.

- Software should look for data error interrupts; that is, for bits 7, 9, and 15 of the RINTSTS register. If required, software can terminate the data transfer by sending the STOP command.
- Software should look for Transmit_FIFO_Data_Request and/or timeout conditions from data starvation by the host. In both cases, the software should write data into the FIFO.
- When a Data_Transfer_Over interrupt is received, the data command is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the Host Controller should send the STOP command, if necessary. Completion of the AUTO-STOP command is reflected by the Auto_command_done interrupt – bit 14 of the RINTSTS register. A response to AUTO_STOP is stored in RESP1 @0x34.

9. Stream Read

A stream read is like the block read mentioned in “Single-Block or Multiple-Block Read”, except for the following bits in the Command register:

```
transfer_mode = 1; //Stream transfer  
cmd_index = CMD20;
```

A stream transfer is allowed for only a single-bit bus width.

10. Stream Write

A stream write is exactly like the block write mentioned in “Single-Block or Multiple-Block Write”, except for the following bits in the Command register:

```
transfer_mode = 1;//Stream transfer  
cmd_index = CMD11;
```

In a stream transfer, if the byte count is 0, then the software must send the STOP command. If the byte count is not 0, then when a given number of bytes completes a transfer, the Host Controller sends the STOP command. Completion of this AUTO_STOP command is reflected by the Auto_command_done interrupt. A response to an AUTO_STOP is stored in the RESP1 register@0x34.

A stream transfer is allowed for only a single-bit bus width.

11. Packed Commands

In order to reduce overhead, read and write commands can be packed in groups of commands—either all read or all write—that transfer the data for all commands in the group in one transfer on the bus.

Packed commands can be of two types:

- Packed Write: CMD23 →CMD25
- Packed Read: CMD23 → CMD25 → CMD23 → CMD18

Packed commands are put in packets by the application software and are transparent to the core.

12. Sending Stop or Abort in Middle of Transfer

The STOP command can terminate a data transfer between a memory card and

the Controller, while the ABORT command can terminate an I/O data transfer for only the SDIO_IOONLY and SDIO_COMBO cards.

- Send STOP command – Can be sent on the command line while a data transfer is in progress; this command can be sent at any time during a data transfer.

You can also use an additional setting for this command in order to set the Command register bits (5-0) to CMD12 and set bit 14 (stop_abort_cmd) to 1. If stop_abort_cmd is not set to 1, the Controller does not know that the user stopped a data transfer. Reset bit 13 of the Command register (wait_prvdata_complete) to 0 in order to make the Controller send the command at once, even though there is a data transfer in progress.

- Send ABORT command – Can be used with only an SDIO_IOONLY or SDIO_COMBO card. To abort the function that is transferring data, program the function number in ASx bits (CCCR register of card, address 0x06, bits (0-2) using CMD52.

13. Suspend or Resume Sequence

In an SDIO card, the data transfer between an I/O function and the Controller can be temporarily halted using the SUSPEND command; this may be required in order to perform a high-priority data transfer with another function. When desired, the data transfer can be resumed using the RESUME command.

The following functions can be implemented by programming the appropriate bits in the CCCR register (Function 0) of the SDIO card. To read from or write to the CCCR register, use the CMD52 command.

- SUSPEND data transfer – Non-data command
 - 1) Check if the SDIO card supports the SUSPEND/RESUME protocol; this can be done through the SBS bit in the CCCR register @0x08 of the card.
 - 2) Check if the data transfer for the required function number is in process; the function number that is currently active is reflected in bits 0-3 of the CCCR register @0x0D. Note that if the BS bit (address 0xc::bit 0) is 1, then only the function number given by the FSx bits is valid.
 - 3) To suspend the transfer, set BR (bit 2) of the CCCR register @0x0C.
 - 4) Poll for clear status of bits BR (bit 1) and BS (bit 0) of the CCCR @0x0C. The BS (Bus Status) bit is 1 when the currently-selected function is using the data bus; the BR (Bus Release) bit remains 1 until the bus release is complete. When the BR and BS bits are 0, the data transfer from the selected function has been suspended.
- RESUME data transfer – This is a data command
 - 1) Check that the card is not in a transfer state, which confirms that the bus is free for data transfer.
 - 2) If the card is in a disconnect state, select it using CMD7. The card status can be retrieved in response to CMD52/CMD53 commands.
 - 3) Check that a function to be resumed is ready for data transfer; this can be confirmed by reading the RFx flag in CCCR @0x0F. If RF = 1, then the

function is ready for data transfer.

- 4) To resume transfer, use CMD52 to write the function number at FSx bits (0-3) in the CCCR register @0x0D. Form the command argument for CMD52 and write it in CMDARG @0x28.
- 5) Write the block size in the BLKSIZ register @0x1C; data will be transferred in units of this block size.
- 6) Write the byte count in the BYTCNT register @0x20. This is the total size of the data; that is, the remaining bytes to be transferred. It is the responsibility of the software to handle the data.
- 7) Program Command register; similar to a block transfer.
- 8) When the Command register is programmed, the command is sent and the function resumes data transfer. Read the DF flag (Resume Data Flag). If it is 1, then the function has data for the transfer and will begin a data transfer as soon as the function or memory is resumed. If it is 0, then the function has no data for the transfer.
- 9) If the DF flag is 0, then in case of a read, the Host Controller waits for data. After the data timeout period, it gives a data timeout error.

14. Read_Wait Sequence

Read_wait is used with only the SDIO card and can temporarily stall the data transfer—either from function or memory—and allow the host to send commands to any function within the SDIO device. The host can stall this transfer for as long as required. The Host Controller provides the facility to signal this stall transfer to the card. The steps for doing this are:

- 1) Check if the card supports the read_wait facility; read SRW (bit 2) of the CCCR register @0x08. If this bit is 1, then all functions in the card support the read_wait facility. Use CMD52 to read this bit.
- 2) If the card supports the read_wait signal, then assert it by setting the read_wait (bit 6) in the CTRL register @0x00.
- 3) Clear the read_wait bit in the CTRL register.

15. Controller/DMA/FIFO Reset Usage

- Controller reset – Resets the controller by setting the controller_reset bit (bit 0) in the CTRL register; this resets the CIU and state machines, and also resets the BIU-to-CIU interface. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.
- FIFO reset - Resets the FIFO by setting the fifo_reset bit (bit 1) in the CTRL register; this resets the FIFO pointers and counters of the FIFO. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.

In external DMA transfer mode, even when the FIFO pointers are reset, if there is a DMA transfer in progress, it could push or pop data to or from the FIFO; the DMA itself completes correctly. In order to clear the FIFO, the software should issue an additional FIFO reset and clear any FIFO underrun or overrun errors in the RAWINTS register caused by the DMA transfers after the FIFO was reset.

16. Card Read Threshold

When an application needs to perform a Single or Multiple Block Read command, the application must program the CardThrCtl register with the appropriate Card Read Threshold size (CardRdThreshold) and set the Card Read Threshold Enable (CardRdThrEnable) bit to 1'b1. This additional programming ensures that the Host controller sends a Read Command only if there is space equal to the CardRDThreshold available in the Rx FIFO. This in turn ensures that the card clock is not stopped in the middle a block of data being transmitted from the card. The Card Read Threshold can be set to the block size of the transfer, which guarantees that there is a minimum of one block size of space in the RxFIFO before the controller enables the card clock. The Card Read Threshold is required when the Round Trip Delay is greater than 0.5cclk_in period.

17. Error Handling

The Host Controller implements error checking; errors are reflected in the RAWINTS register@0x44 and can be communicated to the software through an interrupt, or the software can poll for these bits. Upon power-on, interrupts are disabled (int_enable in the CTRL register is 0), and all the interrupts are masked (bits 0-31 of the INTMASK register; default is 0).

Error handling:

- Response and data timeout errors – For response timeout, software can retry the command. For data timeout, the Host Controller has not received the data start bit – either for the first block or the intermediate block – within the timeout period, so software can either retry the whole data transfer again or retry from a specified block onwards. By reading the contents of the TCBCNT later, the software can decide how many bytes remain to be copied.
- Response errors – Set when an error is received during response reception. In this case, the response that copied in the response registers is invalid. Software can retry the command.
- Data errors – Set when error in data reception are observed; for example, data CRC, start bit not found, end bit not found, and so on. These errors could be set for any block-first block, intermediate block, or last block. On receipt of an error, the software can issue a STOP or ABORT command and retry the command for either whole data or partial data.
- Hardware locked error – Set when the Host Controller cannot load a command issued by software. When software sets the start_cmd bit in the CMD register, the Host Controller tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
- FIFO underrun/overrun error – If the FIFO is full and software tries to write data in the FIFO, then an overrun error is set. Conversely, if the FIFO is empty and the software tries to read data from the FIFO, an underrun error is set. Before reading or writing data in the FIFO, the software should read the fifo_empty or fifo_full bits in the Status register.
- Data starvation by host timeout – Raised when the Host Controller is waiting for software intervention to transfer the data to or from the FIFO, but the software does not transfer within the stipulated timeout period. Under this

condition and when a read transfer is in process, the software should read data from the FIFO and create space for further data reception. When a transmit operation is in process, the software should fill data in the FIFO in order to start transferring data to the card.

- CRC Error on Command – If a CRC error is detected for a command, the CE-ATA device does not send a response, and a response timeout is expected from the Host Controller. The ATA layer is notified that an MMC transport layer error occurred.

Notes: During a multiple-block data transfer, if a negative CRC status is received from the device, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the RINTSTS register. It then continues further data transmission until all the bytes are transmitted.

13.6.5 Voltage Switching

The Host Controller supports SD 3.0 Ultra High Speed (UHS-1) and is capable of voltage switching in SD-mode, which can be applied to SD High-Capacity (SDHC) and SD Extended Capacity (SDXC) cards. UHS-1 supports only 4-bit mode.

SD 3.0 UHS-1 supports the following transfer speed modes for UHS-50 and/or UHS-104 cards:

- DS – default-speed up to 25MHz, 3.3V signaling
- HS – high-speed up to 50MHz, 3.3V signaling
- SDR12 – SDR up to SDR 25MHz, 1.8V signaling
- SDR25 – SDR up to 50MHz, 1.8V signaling
- SDR50 – SDR up to 100MHz, 1.8V signaling
- DDR50 – DDR up to 50MHz, 1.8V signaling

Voltage selection can be done in only SD mode. The first CMD0 selects the bus mode-either SD mode or SPI mode. The card must be in SD mode in order for 1.8V signaling mode to apply, during which time the card cannot be switched to SPI mode or 3.3V signaling without a power cycle.

If the System BIOS in an embedded system already knows that it is connected to an SD 3.0 card, then the driver programs the Controller to initiate ACMD41. The software knows from the response of ACMD41 whether or not the card supports voltage switching to 1.8V.

- If bit 32 of ACMD41 response is 1'b1: card supports voltage switching and next command-CMD11-invokes voltage switching sequence. After CMD11 is started, the software must program the VOLT_REG register in the CSR space with the appropriate card number.
- If bit 32 of ACMD41 response is 1'b0: card does not support voltage switching and CMD11 should not be started.

If the card and host controller accept voltage switching, then they support UHS-1 modes of data transfer. After the voltage switch to 1.8V, SDR12 is the default speed.

Since the UHS-1 can be used in only 4-bit mode, the software must start ACMD6 and change the card data width to 4-bit mode; ACMD6 is driven in any of the UHS-1 speeds. If the host wants to select the DDR mode of data transfer, then the software must program the DDR_REG register in the CSR space with the appropriate card number.

To choose from any of the SDR or DDR modes, appropriate values should be programmed in the CLKDIV register.

1. Voltage Switch Operation

The Voltage Switch operation must be performed in SD mode only.

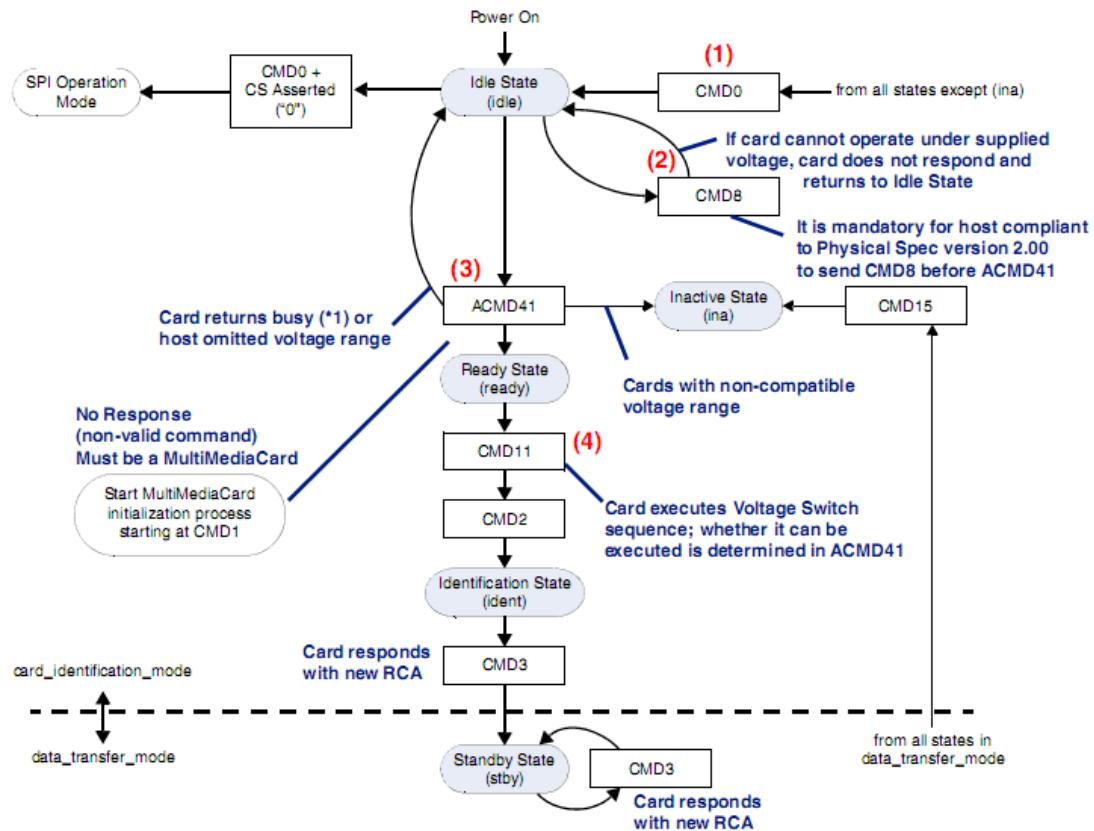


Fig. 13-9 Voltage Switching Command Flow Diagram

The following outlines the steps for the voltage switch programming sequence

- 1) Software Driver starts CMD0, which selects the bus mode as SD.
- 2) After the bus is in SD card mode, CMD8 is started in order to verify if the card is compatible with the SD Memory Card Specification, Version 2. 00. CMD8 determines if the card is capable of working within the host supply voltage specified in the VHS (19:16) field of the CMD; the card supports the current host voltage if a response to CMD8 is received.
- 3) ACMD 41 is started. The response to this command informs the software if the card supports voltage switching; bits 38, 36, and 32 are checked by the card argument of ACMD41; refer to following figure.

47	46	45-40	39	38	37	36	35-33	32	31-16	15-08	07-01	00
S	D	Index	Busy 31	HCS 30	(FB) 29	XPC 28	Reserved 27-25	S18R 24	OCR 23-08	Reserved 07-00	CRC7	E
0	1	101001	0	X	0	X	000	X	xxxxh	0000000	xxxxxx	1

Host Capacity Support
 0b: SDSC-only Host
 1b: SDHC or SDXC supported

SCXC Power Control
 0b: Power saving
 1b: Maximum performance

S18R: Switching to 1.8V Request
 0b: Use current signal voltage
 1b: Switch to 1.8V signal voltage

Fig. 13-10 ACMD41 Argument

- Bit 30 informs the card if host supports SDHC/SDXC or not; this bit should be set to 1'b1.
- Bit 28 can be either 1 or 0.
- Bit 24 should be set to 1'b1, indicating that the host is capable of voltage switching; refer to Figure 17-16.

47	46	45-40	39	38	37	36-33	32	31-16	15-08	07-01	00
S	D	Index	Busy 31	CCS 30	Rsvd 29	Reserved 28-25	S18R 24	OCR 23-08	Reserved 07-00	CRC7	E
0	0	111111	X	X	0	0000	X	xxxxh	0000000	1111111	1

Busy Status
 0b: On Initialization
 1b: Initialization complete

Card Capacity Status
 0b: SDSC
 1b: SDHC or SDXC

S18R: Switching to 1.8V Accepted
 0b: Continues current voltage signalling
 1b: Ready for switching signal voltage

Fig. 13-11 ACMD41 Response(R3)

- Bit 30 – If set to 1'b1, card supports SDHC/SDXC; if set to 1'b0, card supports only SDSC
 - Bit 24 – If set to 1'b1, card supports voltage switching and is ready for the switch
 - Bit 31 – If set to 1'b1, initialization is over; if set to 1'b0, means initialization in process
- 4) If the card supports voltage switching, then the software must perform the steps discussed for either the "Voltage Switch Normal Scenario" or the "Voltage Switch Error Scenario".

1. Voltage Switch Normal Scenario

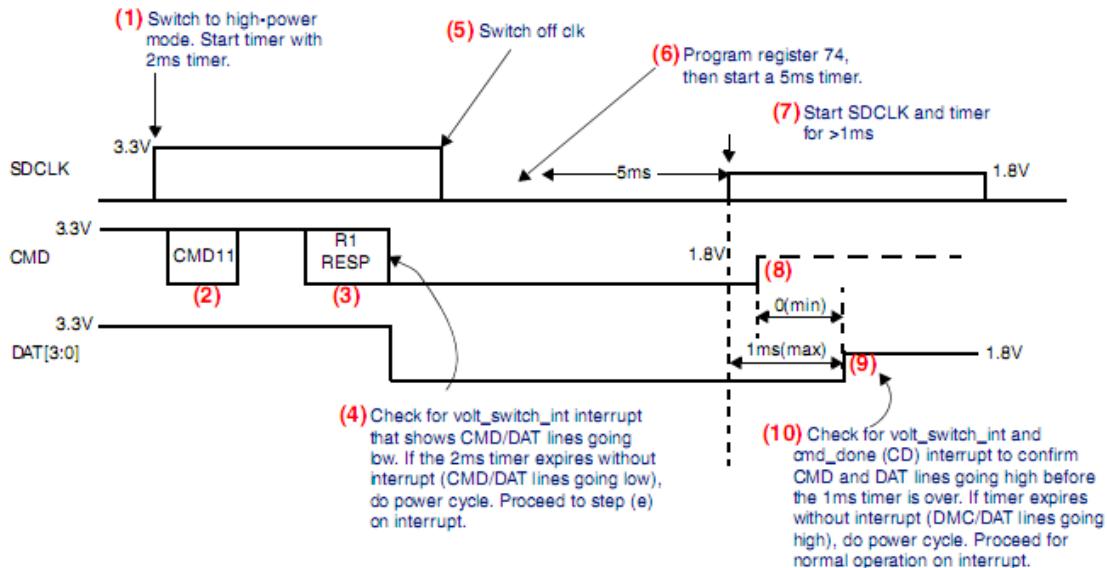


Fig. 13-12 Voltage Switch Normal Scenario

- The host programs CLKENA—cclk_low_power register—with zero (0) for the corresponding card, which makes the host controller move to high-power mode. The application should start a timer with a recommended value of 2ms; this value of 2 ms is determined as below: Total clk required for CMD11 = 48 clks

Total clk required for RESP R1 = 48 clks

Maximum clk delay between MCD11 end to start of RESP1 = 60 clks

Total = 48+48 + 60 = 160

Minimum frequency during enumeration is 100 KHz; that is, 10us

Total time = 160 * 10us = 1600us = 1. 6ms ~ 2ms

- The host issues CMD11 to start the voltage switch sequence. Set bit 28 to 1'b1 in CMD when setting CMD11; for more information on setting bits, refer to "Boot Operation".
- The card returns R1 response; the host controller does not generate cmd_done interrupt on receiving R1 response.
- The card drives CMD and DAT [3:0] to low immediately after the response. The host controller generates interrupt (VOLT_SWITCH_INT) once the CMD or DAT [3:0] line goes low. The application should wait for this interrupt. If the 2ms timer expires without an interrupt (CMD/DAT lines going low), do a power cycle.

Note: Before doing a power cycle, switch off the card clock by programming CLKENA register

Proceed to step (5) on getting an interrupt (VOLT_SWITCH_INT).

Note: This interrupt must be cleared once this interrupt is received. Additionally, this interrupt should not be masked during the voltage switch sequence.

If the timer expires without interrupt (CMD/DAT lines going low), perform a power cycle. Proceed to step (5) on interrupt.

- 5) Program the CLKENA, cclk_enable register, with 0 for the corresponding card; the host stops supplying SDCLK.
- 6) Program VOLT_REG to the required values for the corresponding card. The application must program the newly-defined VOLT_REG register to assign 1 for the bit corresponding to the card number. The application should start a timer > 5ms.
- 7) After the 5ms timer expires, the host voltage regulator is stable. Program CLKENA, cclk_enable register, with 1 for the corresponding card; the host starts providing SDCLK at 1. 8V; this can be at zero time after VOLT_REG has been programmed. When the CLKENA register is programmed, the application should start another timer > 1ms.
- 8) By detecting SDCLK, the card drives CMD to high at 1. 8V for at least one clock and then stops driving (tri-state); CMD is triggered by the rising edge of SDCLK (SDR timing).
- 9) If switching to 1. 8V signaling is completed successfully, the card drives DAT [3:0] to high at 1. 8V for at least one clock and then stops driving (tri-state); DAT [3:0] is triggered by the rising edge of SDCLK (SDR timing). DAT[3:0] must be high within 1ms from the start of SDCLK.
- 10) The host controller generates a voltage switch interrupt (VOLT_SWITCH_INT) and a command done (CD) interrupt once the CMD and DAT[3:0] lines go high. The application should wait for this interrupt to confirm CMD and DAT lines going high before the 1ms timer is done.

If the timer expires without the voltage switch interrupt (VOLT_SWITCH_INT), a power cycle should be performed. Program the CLKENA register to stop the clock for the corresponding card number. Wait for the cmd_done (CD) interrupt. Proceed for normal operation on interrupt. After the sequence is completed, the host and the card start communication in SDR12 timing.

2. Voltage Switch Error Scenario

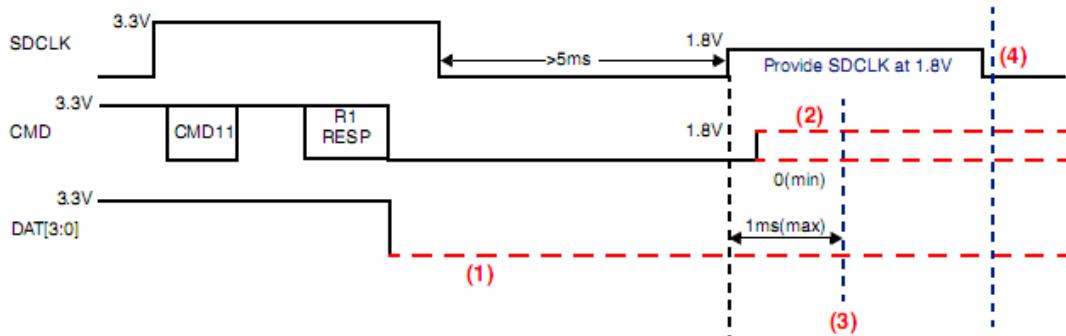


Fig. 13-13 Voltage Switch Error Scenario

- 1) If the interrupt (VOLT_SWITCH_INT) does not come, then the 2 ms timer should time out and a power cycle should be initiated.

Note: Before performing a power cycle, switch off the card clock by programming CLKENA register; no cmd_done (CD) interrupt is generated.

Additionally, if the card detects a voltage error at any point in between steps (5)

and (7) in Figure 17-17, the card keeps driving DAT[3:0] to low until card power off.

- 2) CMD can be low or tri-state.
- 3) The host controller generates a voltage switch interrupt once the CMD and DAT[3:0] lines go high. The application should check for an interrupt to confirm CMD and DAT lines going high before the 1 ms timer is done.

If the 1 ms timer expires without interrupt (VOLT_SWITCH_INT) and cmd_done (CD), a power cycle should be performed. Program the CLKENA register to stop SDCLK of the corresponding card. Wait for the cmd_done interrupt. Proceed for normal operation on interrupt.

- 4) If DAT[3:0] is low, the host drives SDCLK to low and then stops supplying the card power.

Note: The card checks voltages of its own regulator output and host signals to ensure they are less than 2.5V. Errors are indicated by (1) and (2) in Figure 7-18.

- If voltage switching is accepted by the card, the default speed is SDR12.
- Command Done is given:
 - If voltage switching is properly done, CMD and DAT line goes high.
 - If switching is not complete, the 1ms timer expires, and the card clk is switched off.

Note: No other CMD should be driven before the voltage switching operation is completed and Command Done is received.

- The application should use CMD6 to check and select the particular function; the function appropriate-speed should be selected.

After the function switches, the application should program the correct value in the CLKDIV register, depending on the function chosen. Additionally, if Function 0x4 of the Access mode is chosen—that is, DDR50, then the application should also program 1'b1 in DDR_REG for the card number that has been selected for DDR50 mode.

13.6.6 Back-End Power

Each device needs one bit to control the back-end power supply for an embedded device; this bit does not control the VDDH of the host controller. A back_end_power register enables software programming for back-end power. The value on this register is output to the back_end_power signal, which can be used to switch power on and off the embedded device.

13.6.7 DDR Operation

1. 4-bit DDR Programming Sequence

DDR programming should be done only after the voltage switch operation has completed. The following outlines the steps for the DDR programming sequence:

- 1) Once the voltage switch operation is complete, the user must program VOLT_REG to the required values for the corresponding card.
 - To start a card to work in DDR mode, the application must program a bit of the newly defined VOLT_REG[31:16] register with a value of 1'b1.
 - The bit that the user programs depends on which card is to be accessed in DDR mode.
- 2) To move back to SDR mode, a power cycle should be run on the card—putting the card in SDR12 mode—and only then should VOLT_REG[31:16] be set back to 1'b0 for the appropriate card.

2. 8-bit DDR Programming Sequence

The following outlines the steps for the 8-bit DDR programming sequence:

- 1) The cclk_in signal should be twice the speed of the required cclk_out. Thus, if the cclk_out signal is required to be 50 MHz, the cclk_in signal should be 100 MHz.
- 2) The CLKDIV register should always be programmed with a value higher than zero (0); that is, a clock divider should always be used for 8-bit DDR mode.
- 3) The application must program the UHS_REG [31:16] register (DDR_REG bits) by assigning it with a value of 1 for the bit corresponding to the card number; this causes the selected card to start working in DDR mode.
- 4) Depending on the card number, the CTYPE [31:16] bits should be set in order to make the host work in the 8-bit mode.

3. eMMC4.5 DDR START Bit

The eMMC4.5 changes the START bit definition in the following manner:

- 1) Receiver samples the START bit on the rising edge.
- 2) On the next rising edge after sampling the START bit, the receiver must sample the data.
- 3) Removes requirement of the START bit and END bit to be high for one full cycle.

Notes: The Host Controller does not support a START bit duration higher than one clock cycle. START bit durations of one or less than one clock cycle are supported and can be defined at the time of startup by programming the EMMC_DDR_REG register.

Following figure illustrates cases for the definition change of the START bit with eMMC4.5; it also illustrates how some of these cases can fail in sampling when higher-value delays are considered for I/O PADs.

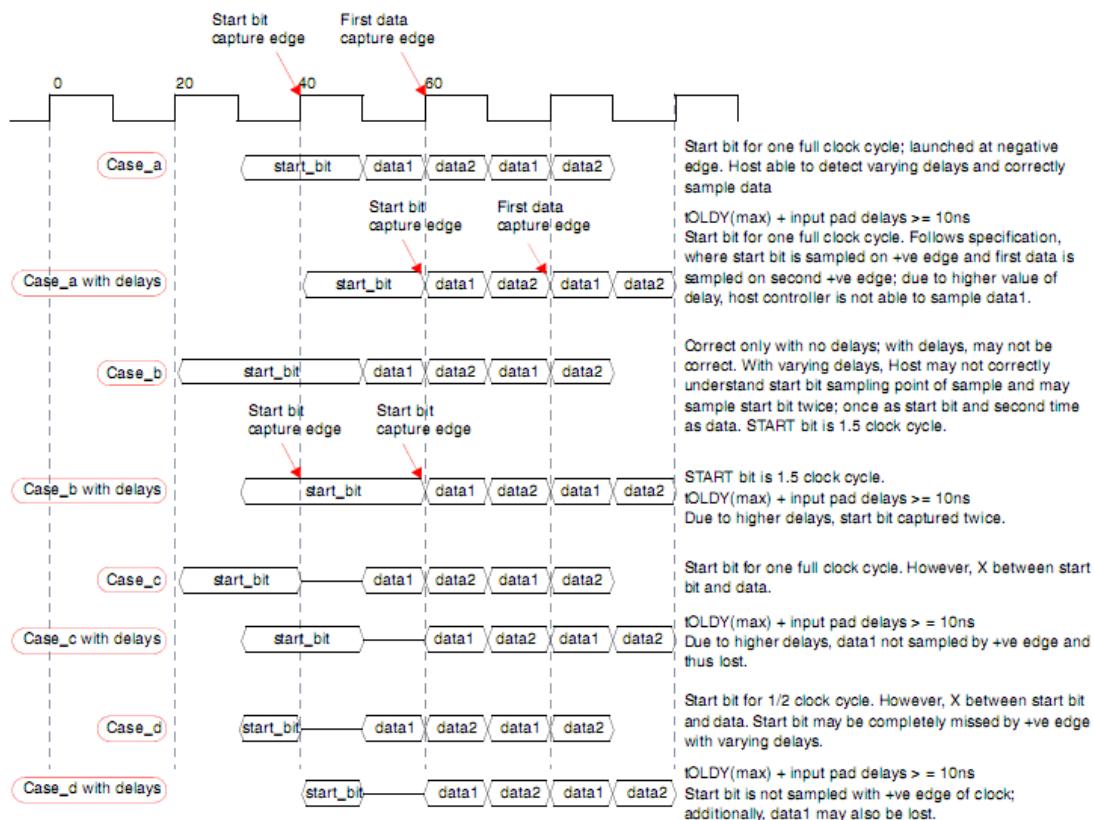


Fig. 13-14 CASES for eMMC 4.5 START bit

4. Reset Command/Moving from DDR50 to SDR12

To reset the mode of operation from DDR50 to SDR12, the following sequence of operations has to be done by the application:

- 1) Issue CMD0.

When CMD0 is received, the card changes from DDR50 to SDR12.

- 2) Program the CLKDIV register with an appropriate value.

- 3) Set DDR_REG to 0.

Note: The VOLT_REG register should not be programmed to 0 while switching from DDR50 to SDR12, since the card is still operating in 1.8V mode after receiving CMD0.

13.6.8 H/W Reset Operation

When the RST_n signal goes low, the card enters a pre-idle state from any state other than the inactive state.

H/W Reset Programming Sequence

The following outlines the steps for the H/W reset programming sequence:

- 1) Program CMD12 to end any transfer in process.
- 2) Wait for DTO, even if no response is sent back by the card.

3) Set the following resets:

- DMA reset– CTRL[2]
- FIFO reset – CTRL[1] bits

Note: The above steps are required only if a transfer is in process.

- 4) Program the CARD_RESET register with a value of 0; this can be done at any time when the card is connected to the controller. This programming asserts the RST_n signal and resets the card.
- 5) Wait for minimum of 1 μ s or cclk_in period, whichever is greater
- 6) After a minimum of 1 μ s, the application should program a value of 0 into the CARD_RESET register. This de-asserts the RST_n signal and takes the card out of reset.
- 7) The application can program a new CMD only after a minimum of 200 μ s after the de-assertion of the RST_n signal, as per the MMC 4.41 standard.

Note: For backward compatibility, the RST_n signal is temporarily disabled in the card by default. The host may need to set the signal as either permanently enabled or permanently disabled before it uses the card.

13.6.9 FBE Scenarios

An FBE occurs due to an AHB error response on the AHB bus. This is a system error, so the software driver should not perform any further programming to the Host. The only recovery mechanism from such scenarios is to do one of the following:

- Issue a hard reset by asserting the reset_n signal
- Do a program controller reset by writing to the CTRL[0] register

FIFO Overflow and Underflow

During normal data transfer conditions, FIFO overflow and underflow will not occur. However if there is a programming error, then FIFO overflow/underflow can result. For example, consider the following scenarios.

- For transmit: PBL=4, Tx watermark = 1. For the above programming values, if the FIFO has only one location empty, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pushes into the FIFO. This will result in a FIFO overflow interrupt.
- For receive: PBL=4, Rx watermark = 1. For the above programming values, if the FIFO has only one location filled, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pops to the FIFO. This will result in a FIFO underflow interrupt.

The driver should ensure that the number of bytes to be transferred as indicated in the descriptor should be a multiple of 4bytes with respect to H_DATA_WIDTH=32. For example, if the BYTCNT = 13, the number of bytes indicated in the descriptor should be 16 for H_DATA_WIDTH=32.

Programming of PBL and Watermark Levels

The DMAC performs data transfers depending on the programmed PBL and threshold values.

Table 13-11 PBL and Watermark Levels

PBL (Number of transfers)	Tx/Rx Watermark Value
1	greater than or equal to 1
4	greater than or equal to 4
8	greater than or equal to 8
16	greater than or equal to 16
32	greater than or equal to 32
64	greater than or equal to 64
128	greater than or equal to 128
256	greater than or equal to 256

13.6.10 Variable Delay/Clock Generation

Variable delay mechanism for the cclk_in_drv is optional, but it can be useful in order to meet a range of hold-time requirements across modes. Variable delay mechanism for the cclk_in_sample is mandatory and is required to achieve the correct sampling point for data.

cclk_in/cclk_in_sample/ cclk_in_drv is generated by Clock Generation Unit (CLKGEN) with variable delay mechanism, which includes Phase Shift Unit and Delay Line Unit selectable.

The Phase Shift Unit can shift cclk_in_sample/cclk_in_drv by 0/90/180/270-degree relative to cclk_in, controlled by *sample_degree/drv_degree*.

The Delay Line Unit can shift cclk_in_sample/cclk_in_drv in the unit of 40ps~80ps for every delay element. The delay unit number is determined by *sample_delaynum/drv_delaynum*, and enabled by *sample_sel/drv_sel*.

cclk_in is generated by cclkin divided by 2. cclk_in_drv and cclk_in_sample clocks are phase-shifted with delayed versions of cclk_in. All clocks are recommended to have a 50% duty cycle; DDR modes must have 50% duty cycles.

The architecture is as follows.

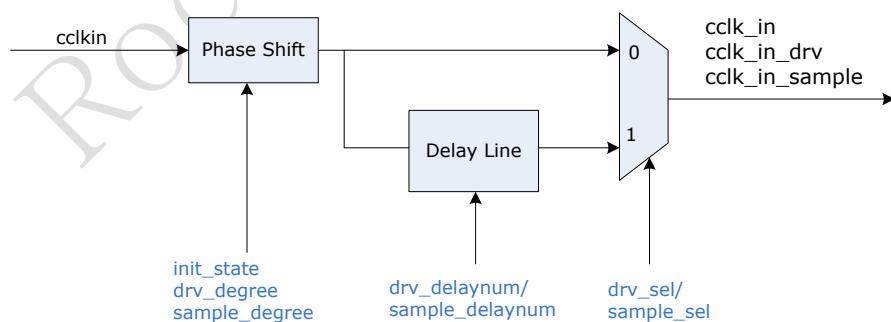


Fig. 13-15 Clock Generation Unit

The control signals for different Host Controller instance are shown as follows:

Table 13-12 Configuration for SDMMC Clock Generation

Signal Name	Source	Default	Description
init_state	CRU_SDMMC_CON0[0]	0	Soft initial state for phase shift.
drv_degree [1:0]	CRU_SDMMC_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum [7:0]	CRU_SDMMC_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_SDMMC_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree [1:0]	CRU_SDMMC_CON1[1:0]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum [7:0]	CRU_SDMMC_CON1[9:2]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_SDMMC_CON1[10]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

Table 13-13 Configuration for SDIO Clock Generation

Signal Name	Source	Default	Description
init_state	CRU_SDIO0_CON0[0]	0	Soft initial state for phase shift.
drv_degree [1:0]	CRU_SDIO0_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum [7:0]	CRU_SDIO0_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_SDIO0_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree [1:0]	CRU_SDIO0_CON1[1:0]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum [7:0]	CRU_SDIO0_CON1[9:2]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_SDIO0_CON1[10]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

Table 13-14 Configuration for eMMC Clock Generation

Signal Name	Source	Default	Description
init_state	CRU_EMMC_CON0[0]	0	Soft initial state for phase shift.

drv_degree [1:0]	CRU_EMMC_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum [7:0]	CRU_EMMC_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_EMMC_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree [1:0]	CRU_EMMC_CON1[1:0]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum [7:0]	CRU_EMMC_CON1[9:2]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_EMMC_CON1[10]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

The following outlines the steps for clock generation sequence:

- 1) Assert init_state to soft reset the CLKGEN.
- 2) Configure drv_degree/sample_degree.
- 3) If fine adjustment required, delay line can be used by configuring drv_delaynum/sample_delaynum and drv_sel/sample_sel.
- 4) Dis-assert init_state to start CLKGEN.

13.6.11 Variable Delay Tuning

Tuning is defined by SD and MMC cards to determine the correct sampling point required for the host, especially for the speed modes SDR104 and HS200 where the output delays from the cards can be up to 2 UI. Tuning is required for other speed modes-such as DDR50-even though the output delay from the card is less than one cycle.

Command for tuning is different for different cards.

- SD Memory Card:
 - CMD19 – SD card for SDR50 and SDR104 speed modes. Tuning data is defined by card specifications.
 - CMD6 – SD card for speed modes not supporting CMD19. Tuning data is the 64byte SD status.
- Multimedia Card:
 - CMD21 – MMC card for HS200 speed mode. Tuning data is defined by card specifications.

- CMD8 – MMC card for speed modes not supporting CMD21. Tuning data is 512 byte ExtCSD data.

The following is the procedure for variable delay tuning:

- 1) Set a phase shift of 0-degree on cclk_in_sample.
- 2) Send the Tuning command to the card; the card in turn sends an R1 response on the CMD line and tuning data on the DAT line.
- 3) If the host sees any of the errors—start bit error, data crc error, end bit error, data read time-out, response crc error, response error—then the sampling point is incorrect.
- 4) Send CMD12 to bring the host controller state machines to idle.
 - The card may treat CMD12 as an invalid command because the card has successfully sent the tuning data, and it cannot send a response.
 - The host controller may generate a response time-out interrupt that must be cleared by software.
- 5) Repeat steps 2) to 4) by increasing the phase shift value or delay element number on cclk_in_sample until the correct sampling point is received such that the host does not see any of the errors.
- 6) Mark this phase shift value as the starting point of the sampling window.
- 7) Repeat steps 2 to 4 by increasing the phase shift value or delay element number on cclk_in_sample until the host sees the errors starting to come again or the phase shift value reaches 360-degree.
- 8) Mark the last successful phase shift value as the ending point of the sampling window.

A window is established where the tuning block is matched. For example, for a scenario where the tuning block is received correctly for a phase shift window of 90-degree and 180-degree, then an appropriate sampling point is established as 135-degree. Once a sampling point is established, no errors should be visible in the tuning block.

13.6.12 Package Command

In order to reduce overhead, read and write commands can be packed in groups of commands—either all read or all write—that transfer the data for all commands in the group in one transfer on the bus.

Packed commands can be of two types:

- Packed Write: CMD23 → CMD25
- Packed Read: CMD23 → CMD25 → CMD23 → CMD18

Packed commands are put in packets by the application software and are transparent to the core. For more information on packed commands, refer to the eMMC specification.

13.6.13 Card Detection Method

There are many methods for SDMMC/SDIO card detection.

(1) Method1: Using CDETECT register, which is value on card_detect_n input port. 0 represents presence of card.

(2) Method2: Using card detection unit, outputting host interrupt (*IRQ_ID[46]*). The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value in DEBNCE[23:0]. Following figure illustrates the timing for card-detect signals.

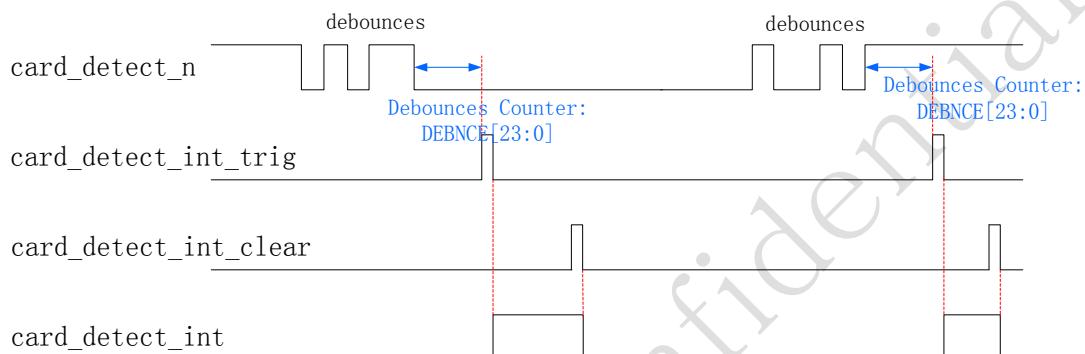


Fig. 13-16 Card Detection Method 2

- Method3: Using card detection unit in GRF, outputting *sdmmc_detect_dual_edge_int(IRQ_ID[86])*, only available for SDMMC. Similar to Method2, except that the debounce is selecting from 5ms/15ms/35ms/50ms; and the insertion/removal detection interrupt can be enabled or cleared respectively. The detailed register information is:

Table 13-15 Register for SDMMC Card Detection Method 3

Signal Name	Source	Default	Description
sd_detectn_rise_edge_irq_en	GRF_SOC_CON0[2]	0	sdmmc detect_n signal rise edge interrupt enable. 1'b1: enable 1'b0: disable
sd_detectn_rise_edge_irq_pd	GRF_SOC_CON0[0]	0	sdmmc detect_n rise edge interrupt pending status. Write 1 to clear the status.
sd_detectn_fall_edge_irq_en	GRF_SOC_CON0[3]	0	sdmmc detect_n signal fall edge interrupt enable. 1'b1: enable 1'b0: disable
sd_detectn_fall_edge_irq_pd	GRF_SOC_CON0[1]	0	sdmmc detect_n fall edge interrupt pending status. Write 1 to clear the status.
grf_filter_cnt_sel	GRF_SOC_CON0[5:4]	0	the counter select for sd card detect filter: 2'b00: 5ms 2'b01: 15ms 2'b10: 35ms 2'b11: 50ms

- Method4: Using card_detect_n for interrupt source, connecting to

IRQ_ID[78] directly.

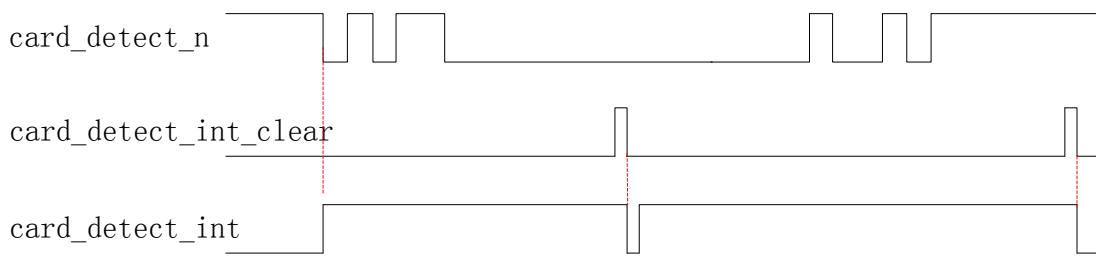


Fig. 13-17 Card Detection Method 4

13.6.14 SDMMC IOMUX With JTAG

The IO for sdmmc_cdata2/sdmmc_cdata3 is shared with jtag_tck/jtag_tms. The condition of usage for SDMMC or JTAG usage is as follows.

- If GRF_SOC_CON0[8](grf_force_jtag) is equal to 1 and sdmmc card is not detected within detection time(defined in GRF_MMC_DET_CNT, in the unit of XIN24M clock), the GPIOs are used for JTAG.
- Otherwise, the GPIOs' usage is defined by IOMUX configuration.

Chapter 14 Power Management Unit (PMU)

14.1 Overview

In order to meet high performance and low power requirements, a power management unit is designed for saving power when RK3128 in low power mode. The RK3128 PMU is dedicated for managing the power of the whole chip.

14.1.1 Features

- Support 2 voltage domains including VD_CORE, VD_LOGIC
- Support 4 separate power domains in the whole chip, which can be power up/down by software based on different application scenes
- In low power mode, the pmu could power up/down vd_core by hardware
- Support CORTEX-A7 core source clock gating in low power mode
- Support Logic Bus source clock gating in low power mode
- Support global interrupt disable in low power mode
- Support pd_pmu_clock switch to 32KHz or pvtm clock in low power mode
- Support DDR self-refresh in low power mode
- Support DDR controller clock auto gating in low power mode
- Support to send idle requests to all NIU in the SoC (details will be described later)
- A group of configurable counter in PMU for HW control (such as PMIC, Core power up/down and so on)
- Support varies configurable wakeup source for low power mode

14.2 Block Diagram

14.2.1 power domain partition

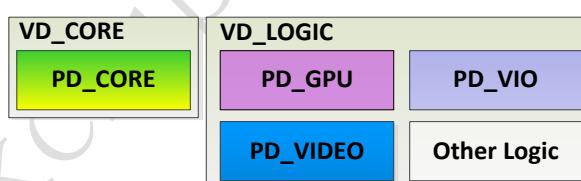


Fig.14-1Power Domain Partition

The above diagram describes the power domain and voltage domain partition, and the following table lists all the power domains.

Table 14-1RK3128 Power Domain and Voltage Domain Summary

Voltage Domain	Power Domain	Description
VD_CORE	PD_CORE	A7 logic
VD_LOGIC	PD_BUS	include pd_bus, pd_peri and other system control unit (GRF, CRU and so on)
	PD_VIO	Video input/output system, include VOP, VIP, IEP, RGA, EBC, MIPI-DSI, HDMI
	PD_VIDEO	Video Encode&Decode , include VEPU, VDPU
	PD_GPU	GPU

14.2.2 PMU block diagram

The following figure is the PMU block diagram. The PMU includes the 3 following sections:

- APB interface and register, which can accept the system configuration
- Low Power State Control, which generate low power control signals.
- Power Switch Control, which control all power domain switch

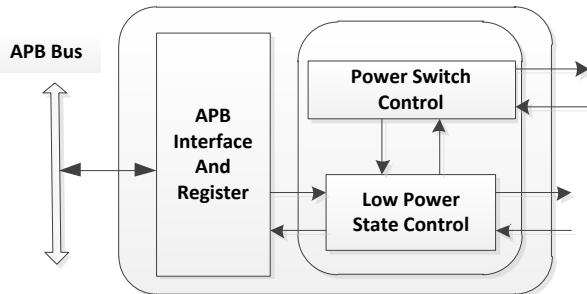


Fig.14-2PMU Bock Diagram

14.3 Power Switch Timing Requirement

The following table describe the switch time for power down and power up progress of each power domain. This table gives the time range, and each power domain switch time will be more than the min time and less than the max time.

Table 14-2 Power Switch Timing

Power domain	type	Power down Switch Timing①(ns)	Power up Switch Timing①(ns)
PD_CORE	min	170.3	132.4
	max	306.7	237.5
	max	199.0	156.1
PD_VIO	min	280.6	217.5
	max	518.5	407.8
PD_VIDEO	min	315.4	244.2
	max	586.2	460.4
PD_GPU	min	470.2	364
	max	871.4	684.1

Notes:the power switch timing is just the chip power electrical parameter, this is not the parameter for the software to determine the power domain status. The software need to check each power domain status register to determine the power status.

14.4 Function Description

14.4.1 Normal Mode

First of all, we define two modes of power for chip, normal mode and low power mode.

In normal mode, the PMU can power off/on all power domain (except pd_core)

by setting PMU_PWRDN_CON register. At same, pmu can send idle request for every power domain by setting PMU_IDLE_REQ register.

Don't set pd_core power off or send idle_req_core and idle_req_sys in normal mode. This will cause the system to not work properly.

Basically, there are 2 configurations that software can do in normal mode to save power.

- Configure DDR to self-refresh, DDR IO retention and DDR IO power off
- Power down power domains

The first one will save power consumption of using DDR controller and DDR IO. For avoiding confliction, the software must make sure the execution code of this step is not in DDR.

The second one will save power of the power domain which software is shutting down.

14.4.2 Low Power Mode

PMU can work in the Low Power Mode by setting bit[0] of PMU_PWRMODE_CON register. After setting the register, PMU would enter the Low Power mode. In the low power mode, pmu will auto power on/off the specified power domain, send idle req to specified power domain, disable/enable config bus clock and so on. All of above are configurable by setting corresponding registers.

Table 14-3 Low Power State

Num	Hardware Flow	Description of Flow	
0	NORMAL	nomal status	
1	TRANS_NO_FIN	wait transfer to finish	bit[22:16] of PMU_PWRMODE_CON
2	SREF_ENTER	enter to self refresh	bit[8:7] of PMU_PWRMODE_CON
3	CORE_CLK_DIS	core clock gating	bit[1] of PMU_PWRMODE_CON
4	BUS_CLK_DIS	cfgbus clock disable	bit[2] of PMU_PWRMODE_CON
5	CLOCK_LF	switch to 32KHz or pvtm	bit[5] of PMU_PWRMODE_CON
6	CORE_PWRDN	vd core power down	bit[4] of PMU_PWRMODE_CON
7	WAIT_WAKEUP	wait wakeup	
8	WAIT_24M	wait 24MHz stable	bit[6] of PMU_PWRMODE_CON
9	CLOCK_HF	switch to 24MHz	
10	BUS_CLK_EN	cfgbus clock enable	
11	CORE_CLK_EN	core时钟恢复	
12	SREF_EXIT	exit self refresh	
13	CORE_PWRUP	vd core power up	
14	TRANS_RESTORE	restroe transfer	

The Low Power mode have three steps:

- Enter Low Power mode, there are some sub-steps in the enter step, every sub-step can be enable/disable by setting the corresponding register.
- Wait wakeup, there is only armint wakeup source by setting PMU_WAKEUP_CFG[0] register
- Exit Low Power mode, the sub-step are executed depend on whether they were executed in enter low power step.

14.4.3 Wakeup source

There is only one wakeup source, armint which can trigger PMU from power mode to normal mode.

If software expect PMU be woken up from power mode it should be enabled by

write 1 to PMU_WAKEUP_CFG[0] register before entering into power mode.

14.5 Register Description

14.5.1 Register Summary

Name	Offset	Size	Reset Value	Description
PMU_WAKEUP_CFG	0x0000	W	0x00000000	PMU wake-up source configuration register
PMU_PWRDN_CON	0x0004	W	0x00000000	System power gating configuration register
PMU_PWRDN_ST	0x0008	W	0x00000000	System power gating status register
PMU_IDLE_REQ	0x000c	W	0x00000000	PMU Noc idle req control
PMU_IDLE_ST	0x0010	W	0x00000000	PMU Noc idle status
PMU_PWRMODE_CON	0x0014	W	0x00000000	PMU configuration register in power mode flow
PMU_PWR_STATE	0x0018	W	0x00000000	PMU Low power mode state
PMU_OSC_CNT	0x001c	W	0x00005dc0	24MHz OSC stabilization counter threshold
PMU_CORE_PWRDN_CNT	0x0020	W	0x00005dc0	CORE domain power down waiting counter in sleep mode
PMU_CORE_PWRUP_CNT	0x0024	W	0x00005dc0	CORE domain power up waiting counter in sleep mode
PMU_SFT_CON	0x0028	W	0x00000000	PMU Software control in normal mode
PMU_DDR_SREF_ST	0x002c	W	0x00000000	PMU DDR self refresh status
PMU_INT_CON	0x0030	W	0x00000000	PMU interrupt configuration register
PMU_INT_ST	0x0034	W	0x00000000	PMU interrupt status register
PMU_SYS_REG0	0x0038	W	0x00000000	PMU system register0
PMU_SYS_REG1	0x003c	W	0x00000000	PMU system register1
PMU_SYS_REG2	0x0040	W	0x00000000	PMU system register2
PMU_SYS_REG3	0x0044	W	0x00000000	PMU system register3

14.5.2 Detail Register Description

PMU_WAKEUP_CFG

Address: Operational Base + offset (0x0000)

PMU wake-up source configuration register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	armint_wakeup_en ARM interrupt wake-up enable 1'b0: disable 1'b1: enable

PMU_PWRDN_CON

Address: Operational Base + offset (0x0004)

System power gating configuration register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	PD_VIO_DWN_EN Power domain VIO power down enable 1'b0: power on 1'b1: power off
2	RW	0x0	PD_VIDEO_DWN_EN Power domain VIDEO power down enable 1'b0: power on 1'b1: power off
1	RW	0x0	PD_GPU_DWN_EN Power domain GPU power down enable 1'b0: power on 1'b1: power off
0	RW	0x0	CORE_PWROFF_EN software config power off pd_core 1'b1: power off 1'b0: not power off

PMU_PWRDN_ST

Address: Operational Base + offset (0x0008)

System power gating status register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	pd_vio_pwr_st Power domain VIO power status 1'b0: power on 1'b1: power off
2	RW	0x0	pd_video_pwr_st Power domain VIDEO power status 1'b0: power on 1'b1: power off
1	RW	0x0	pd_gpu_pwr_st Power domain GPU power status 1'b0: power on 1'b1: power off
0	RW	0x0	pd_core_pwr_st Power domain core power status 1'b0: power on 1'b1: power off

PMU_IDLE_REQ

Address: Operational Base + offset (0x000c)

PMU Noc idle req control

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	idle_req_crypto_cfg software config crypto domain flush transaction request 1'b1: idle req 1'b0: not idle req

Bit	Attr	Reset Value	Description
6	RW	0x0	idle_req_msch_cfg software configmsch domain flush transaction request 1'b1: idle req 1'b0: not idle req
5	RW	0x0	idle_req_sys_cfg software configsystem domain flush transaction request 1'b1: idle req 1'b0: not idle req
4	RW	0x0	idle_req_core_cfg software configcore domain flush transaction request 1'b1: idle req 1'b0: not idle req
3	RW	0x0	idle_req_gpu_cfg software configgpu domain flush transaction request 1'b1: idle req 1'b0: not idle req
2	RW	0x0	idle_req_vio_cfg software configvio domain flush transaction request 1'b1: idle req 1'b0: not idle req
1	RW	0x0	idle_req_video_cfg software configvideo domain flush transaction request 1'b1: idle req 1'b0: not idle req
0	RW	0x0	idle_req_peri_cfg software configperi domain flush transaction request 1'b1: idle req 1'b0: not idle req

PMU_IDLE_ST

Address: Operational Base + offset (0x0010)

PMU Noc idle status

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RW	0x0	idle_ack_crypto crypto domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
22	RW	0x0	idle_ack_msch msch domain flush transaction acknowledge 1'b0: no ack 1'b1: ack

Bit	Attr	Reset Value	Description
21	RW	0x0	idle_ack_sys sys domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
20	RW	0x0	idle_ack_core core domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
19	RW	0x0	idle_ack_gpu gpu domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
18	RW	0x0	idle_ack_vio vio domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
17	RW	0x0	idle_ack_video video domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
16	RW	0x0	idle_ack_peri peri domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
15:8	RO	0x0	reserved
7	RW	0x0	IDLE_CRYPTO crypto domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
6	RW	0x0	IDLE_MSCH msch domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
5	RW	0x0	IDLE_SYS sys domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
4	RW	0x0	IDLE_CORE core domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
3	RW	0x0	IDLE_GPU gpu domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
2	RW	0x0	IDLE_VIO vio domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish

Bit	Attr	Reset Value	Description
1	RW	0x0	IDLE_VIDEO video domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
0	RW	0x0	IDLE_PERI peri domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish

PMU_PWRMODE_CON

Address: Operational Base + offset (0x0014)

PMU configuration register in power mode flow

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RW	0x0	clr_peri issue idle_req_peri in low power mode 1'b0: not issue 1'b1: issue
22	RW	0x0	clr_video issue idle_req_video in low power mode 1'b0: not issue 1'b1: issue
21	RW	0x0	clr_vio issue idle_req_vio in low power mode 1'b0: not issue 1'b1: issue
20	RW	0x0	clr_gpu issue idle_req_gpu in low power mode 1'b0: not issue 1'b1: issue
19	RW	0x0	clr_core issue idle_req_core in low power mode 1'b0: not issue 1'b1: issue
18	RW	0x0	clr_sys issue idle_req_sys in low power mode 1'b0: not issue 1'b1: issue
17	RW	0x0	clr_msch issue idle_req_msch in low power mode 1'b0: not issue 1'b1: issue
16	RW	0x0	clr_crypto issue idle_req_crypto in low power mode 1'b0: not issue 1'b1: issue
15:10	RW	0x0	ddr0io_ret_de_req ddr0io retention de-assert request 1'b0: de-assert request 1'b1: not de-assert request

Bit	Attr	Reset Value	Description
9	RW	0x0	pmu_int_en pmuint enable 1'b1: enable 1'b0: disable
8	RW	0x0	ddr_gating_en ddrc auto gating in low power mode 1'b0: disable 1'b1: enable
7	RW	0x0	sref_enter_en DDR enter self-refresh enable in low power mode 1'b0: disable DDR enter self-refresh 1'b1: enable DDR enter self-refresh
6	RW	0x0	wait_osc_24m wait 24MHz OSC when wakeup in low power mode 1'b0: disable 1'b1: enable
5	RW	0x0	pmu_use_if pmu domain clock switch to low clock enable 1'b0: not switch to low clock 1'b1: switch to low clock
4	RW	0x0	core_pd_en core power off enable in low power mode 1'b0: core power on 1'b1: core power off
3	RW	0x0	global_int_disable Global interrupt disable 1'b0: enable global interrupt 1'b1: disable global interrupt
2	RW	0x0	clk_bus_src_gate_en config bus clock source gating enable in idle mode 1'b0: enable 1'b1: disable
1	RW	0x0	clk_core_src_gate_en A7 core clock source gating enable in idle mode 1'b0: enable 1'b1: disable
0	RW	0x0	power_mode_en power mode flow enable 1'b0: disable 1'b1: enable

PMU_PWR_STATE

Address: Operational Base + offset (0x0018)

PMU Low power mode state

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	TRANS_RESTORE noc trans restore 1'b0: state not happened 1'b1: state happened
13	RW	0x0	CORE_PWRUP pd core power up state 1'b0: state not happened 1'b1: state happened
12	RW	0x0	SREF_EXIT ddr exit self-refresh 1'b0: state not happened 1'b1: state happened
11	RW	0x0	CORE_CLK_EN pd_core source clock enable 1'b0: state not happened 1'b1: state happened
10	RW	0x0	BUS_CLK_EN config bus source clock enable 1'b0: state not happened 1'b1: state happened
9	RW	0x0	CLOCK_HF pd_pmu switch to normal clock 1'b0: state not happened 1'b1: state happened
8	RW	0x0	WAIT_24M wait 24M osc stable 1'b0: state not happened 1'b1: state happened
7	RW	0x0	WAIT_WAKEUP wait wakeup state 1'b0: state not happened 1'b1: state happened
6	RW	0x0	CORE_PWRDN pd core down state 1'b0: state not happened 1'b1: state happened
5	RW	0x0	CLOCK_LF pd_pmu switch to low speed clock 1'b0: state not happened 1'b1: state happened
4	RW	0x0	BUS_CLK_DIS config bus source clock disable 1'b0: state not happened 1'b1: state happened
3	RW	0x0	CORE_CLK_DIS pd_core source clock disable 1'b0: state not happened 1'b1: state happened
2	RW	0x0	SREF_ENTER ddrselfrefresh enter 1'b0: state not happened 1'b1: state happened

Bit	Attr	Reset Value	Description
1	RW	0x0	TRANS_NO_FIN transfer no finish 1'b0: state not happened 1'b1: state happened
0	RW	0x0	NORMAL normal state 1'b0: state not happened 1'b1: state happened

PMU_OSC_CNT

Address: Operational Base + offset (0x001c)

24MHz OSC stabilization counter threshold

Bit	Attr	Reset Value	Description
31:0	RW	0x05dc0	osc_stabl_cnt_thresh 24MHz OSC stabilization counter threshold

PMU_CORE_PWRDWN_CNT

Address: Operational Base + offset (0x0020)

CORE domain power down waiting counter in sleep mode

Bit	Attr	Reset Value	Description
31:0	RW	0x05dc0	core_pwrdown_cnt_thresh CORE domain power down waiting counter threshold

PMU_CORE_PWRUP_CNT

Address: Operational Base + offset (0x0024)

CORE domain power up waiting counter in sleep mode

Bit	Attr	Reset Value	Description
31:0	RW	0x05dc0	core_pwrup_cnt_thresh CORE domain power up waiting counter threshold

PMU_SFT_CON

Address: Operational Base + offset (0x0028)

PMU Software control in normal mode

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	clk_core_src_gating_cfg pd core source clock disable 1'b1: disable 1'b0: enable
3	RW	0x0	clk_bus_src_gating_cfg config bus source clock disable 1'b1: disable 1'b0: enable
2	RW	0x0	upctl_c_sysreq_cfg software config enter DDR self-refresh by lowpower interface 1'b1: request enter self-refresh 1'b0: not enter self-refresh

Bit	Attr	Reset Value	Description
1	RW	0x0	pmu_lf_ena_cfg software config PMU domain clock switch to low clock 1'b1: switch to low speed clock 1'b0: not switch
0	RW	0x0	low_clk_sel low clock in low power mode select 1'b0: clock pvtm 1'b1: 32KHz clock

PMU_DDR_SREF_ST

Address: Operational Base + offset (0x002c)

PMU DDR self refresh status

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	upctl_c_sysack DDR enter self-refresh acknowledge 1'b0: no ack 1'b1: ack
0	RW	0x0	upctl_c_active DDR enter self-refresh 1'b0: no active 1'b1: active

PMU_INT_CON

Address: Operational Base + offset (0x0030)

PMU interrupt configuration register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	pd_core_int_en Power domain core power switch interrupt enable 1'b0: disable 1'b1: enable
4	RW	0x0	pd_gpu_int_en Power domain gpu power switch interrupt enable 1'b0: disable 1'b1: enable
3	RW	0x0	pd_video_int_en Power domain video power switch interrupt enable 1'b0: disable 1'b1: enable
2	RW	0x0	pd_vio_int_en Power domain vio power switch interrupt enable 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
1	RW	0x0	wakeup_int_en wakeup status interrupt enable 1'b0: disable 1'b1: enable
0	RW	0x0	armint_wakeup_int_en ARM interrupt wakeup status interrupt enable 1'b0: disable 1'b1: enable

PMU_INT_ST

Address: Operational Base + offset (0x0034)

PMU interrupt status register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	pd_core_int_st Power domain core power switch status 1'b0: no power switch happen 1'b1: power switch happen
4	RW	0x0	pd_gpu_int_st Power domain gpu power switch status 1'b0: no power switch happen 1'b1: power switch happen
3	RW	0x0	pd_video_int_st Power domain video power switch status 1'b0: no power switch happen 1'b1: power switch happen
2	RW	0x0	pd_vio_int_st Power domain vio power switch status 1'b0: no power switch happen 1'b1: power switch happen
1	RW	0x0	pwrmode_wakeup_event_trig power mode flow wakeup 1'b0: no wakeup 1'b1: wakeup
0	RW	0x0	armint_wakeup_event_trig ARM interrupt wake-up enent trigger 1'b0: no wakeup 1'b1: wakeup

PMU_SYS_REG0

Address: Operational Base + offset (0x0038)

PMU system register0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg0 PMU system register0

PMU_SYS_REG1

Address: Operational Base + offset (0x003c)

PMU system register1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg1 PMU system register1

PMU_SYS_REG2

Address: Operational Base + offset (0x0040)
 PMU system register2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg2 PMU system register2

PMU_SYS_REG3

Address: Operational Base + offset (0x0044)
 PMU system register3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg3 PMU system register3

14.6 Timing Diagram

14.6.1 Each domain power switch timing

The following figure is the each domain power down and power up timing.

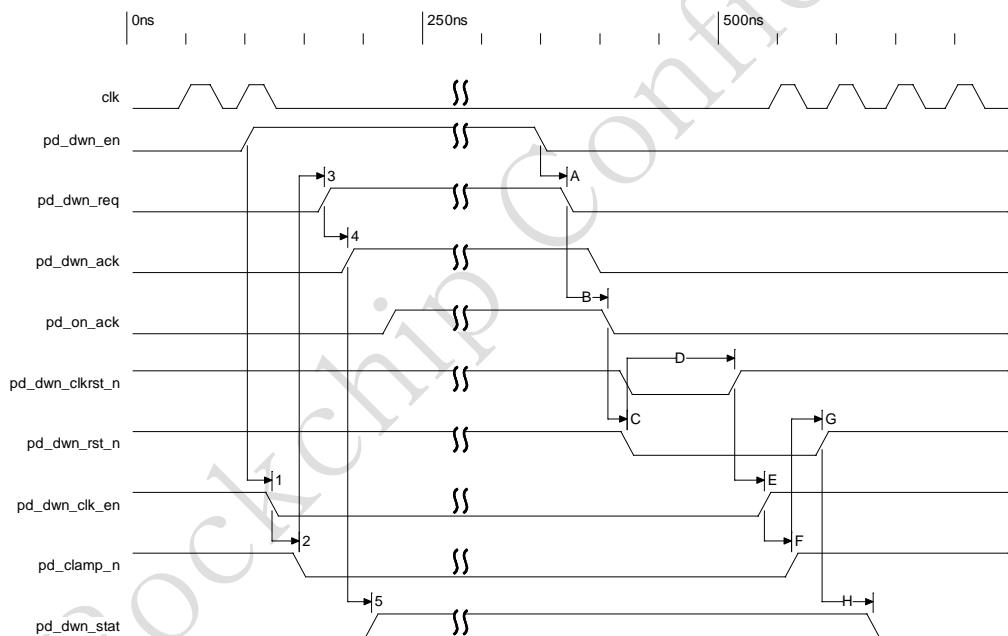


Fig.14-3 Each Domain Power Switch Timing

14.6.2 External wakeup PAD timing

The PMU supports a lot of external wakeup sources, such as SD/MMDC, USBDEV, SIM0/1 detect wakeup, GPIO0 wakeup source and so on. All these external wakeup sources must meet the timing requirement (at least 200us) when the wakeup event is asserted. The following figure gives the timing information.

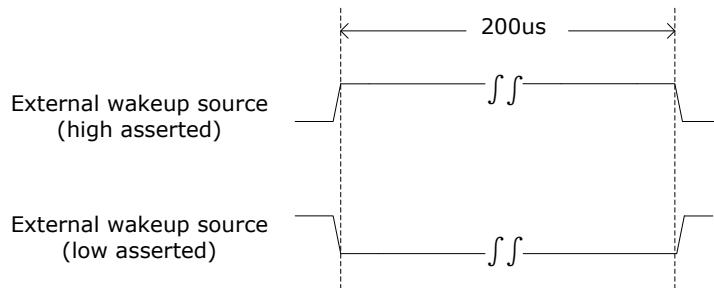


Fig.14-4 External Wakeup Source PAD Timing

14.7 Application Notes

14.7.1 Recommend configurations for power mode.

The PMU is a design with great flexibilities, but just for facilities and inheritances, a group of recommend configurations will be shown below for software. And for convenience, we will define several modes.

The RK3128 can support following 5 recommended power modes:

- normal
- idle mode
- deep idle mode
- sleep mode
- power off mode

The following table lists the detailed description of the modes.

Table 14-4 Power Domain Status Summary in all Work Mode

Item	mode1(idle)	mode2(sleep)	configurable
Core	standby	power down	
Logic Clock	enable	disable	Yes
Logic Clock Source	24MHz	32KHz/PVTM	Yes
PLL	working	power down	Yes
DDR	working	self refresh	Yes

Normal mode

In this mode, you can power off/on or enable/disable the following power domain to save power: PD_VIO/PD_VIDEO/PD_GPU

Idle mode

This mode is used when the core do not have load for a shot while such as waiting for interrupt and the software want to save power by gating Cortex-A7 source clock.

In idle mode, core1/2/3 of Cortex-A7 should be either power off or in WFI/WFE state. The core0 of A7 should be in WFI/WFE state. The configurations of core clock source gating and disable global interrupt are presented. The Cortex-A7 can waked up by an interrupt.

Sleep mode

The sleep mode can power off all power domains except TOP Logic. The VD_CORE is turned off externally, and other domains power off by software.

In sleep mode the clock of PD_PMU can be switched from 24MHz to low speedclock optionally by hardware. The low speed clock can be selected from clock pvtm and 32KHz clock.

In sleep mode all PLLs power down mandatorily to save power by software.

In sleep mode OSC can be disabled optionally by software.

In sleep mode DDR self-refresh can be issued by hardware mandatorily.

14.7.2 System Register

PMU support 4 words register: PMU_SYS_REG0, PMU_SYS_REG1, PMU_SYS_REG2, PMU_SYS_REG3. These registers are always on no matter what low power mode. So software can use these registers to retain some information which is useful after wakeup from any mode.

14.7.3 Configuration Constraint

In order to shut down the power domains correctly, the software must obey the rules bellow:

- Send NIU request to the NIU in power domain that you want to shut down.
- Querying PMU_IDLE_ST register to get the information until the pacific NIU is in idle state.
- Send power request to the power domain through PMU_PWRDN_CON register.
- Querying PMU_PWRDN_ST register to make sure the pacific power domain is power down.

The power domains controlled only by software are showing below:

PD_VIO, PD_GPU, PD_VIDEO.

So you must power off these power domains before enter low power mode if you need.

Chapter 15 Embedded SRAM

15.1 Overview

The Embedded SRAM is the AXI slave device, which supports read and write access to provide system fast access data storage.

15.1.1 Features supported

- Provide 8KB access space
- Support 64bit AXI bus

15.1.2 Features not supported

- Don't support AXI lock transaction
- Don't support AXI exclusive transaction
- Don't support AXI cache function
- Don't support AXI protection function

15.2 Block Diagram

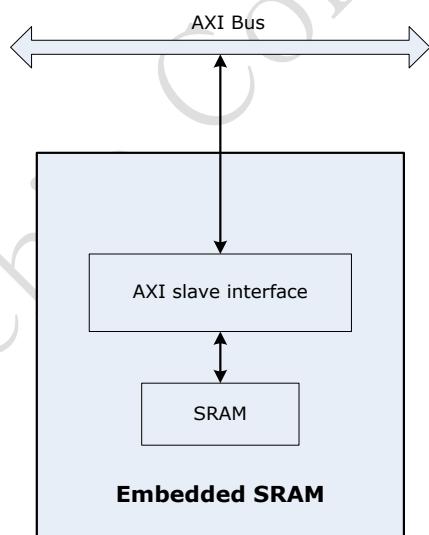


Fig. 15-1 Embedded SRAM block diagram

15.3 Function Description

15.3.1 AXI slave interface

The AXI slave interface is bridge which translate AXI bus access to SRAM interface.

15.3.2 Embedded SRAM access path

The Embedded SRAM can only be accessed by Cortex-A7.

15.3.3 Remap

The Embedded SRAM support remapping.

Before remap, the Embedded SRAM address range is 0x1008_0000 ~ 0x1008_1fff.

After set remap, (controlled by GRF_SOC_CON0[12]), the system can still access the Embedded SRAM by the old address. at same time, the system also can access the Embedded SRAM by the new address 0x0000_0000 ~ 0x0000_1fff (include the boot_addr)

Chapter 16 GPU (Graphics Process Unit)

16.1 Overview

The GPU is a hardware accelerator for 2D and 3D graphics systems. Its triangle rate can be 30 Mtris/s, pixel rate can be 600Mpix/s.

The GPU supports the following graphics standards:

- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.1

The GPU consists of:

- 2 Pixel Processors (PPs)
- a geometry Processor (GP)
- a Level2 Cache controller (L2)
- a Memory Management Unit (MMU) for each GP and PP included in the GPU

The GPU contains a 64-bit APB bus and a 64-bit AXI bus. CPU configures GPU through APB bus, GPU read and write data through AXI bus.

16.2 Block Diagram

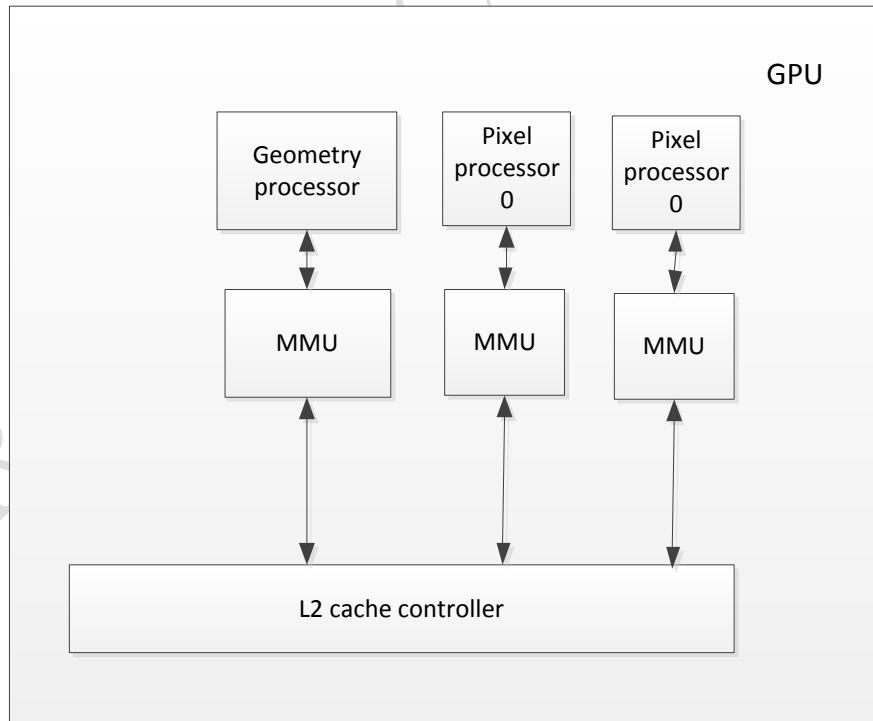


Fig. 16-1 GPU block diagram

As Fig. 16-1 shows, The GPU contains one geometry processor, 2 pixel processors, 3 MMU and a L2 cache controller.

The pixel processor features are:

- Each pixel processor used processes a different tile, enabling a faster turnaround.
- Programmable fragment shader
- Alpha blending
- Complete non-power-of-2 texture support
- Cube mapping
- Fast dynamic branching
- Fast trigonometric functions, including arctangent
- Full floating-point arithmetic
- Framebuffer blend with destination alpha
- Indexable texture samplers
- Line, quad, triangle and point sprites
- No limit on program length
- Perspective correct texturing
- Point sampling, bilinear and trilinear filtering
- Programmable mipmap level-of-detail biasing and replacement
- Stencil buffering, 8-bit
- Two-sided stencil
- Unlimited dependent texture reads
- 4-level hierarchical Z and stencil operations
- Up to 512 times Full scene Anti-Aliasing (FSAA). 4x multisampling times 128x supersampling
- 4-bit per texel compressed texture format

The geometry processor features are:

- Programmable vertex shader
- Flexible input and output formats
- Autonomous operation tile list generation
- Indexed and no-indexed geometry input
- Primitive constructions with points, lines, triangles and quads

The L2 cache controller features are:

- 32KB size
- 4-way set-associative
- Supports up to 32 outstanding AXI transactions
- Implements a standard pseudo-LRU algorithm
- Cache line and line fill burst size is 64 bytes
- Support eight to 64 bytes uncached read bursts and write bursts

- 64-bit interface to memory sub-system
- Support for hit-under-miss and miss-under-miss with the only limitation of AXI ordering rules.

The MMU features are:

- Accesses control registers through the bus infrastructure to configure the memory system
- Each processor has its own MMU to control and translate memory access that the GPU initiates

16.3 Function description

Please refer to the document DDI0437F_mali_400_mp_gpu_r1p1_trm.pdf.

16.4 Register description

The GPU address range is 0x1009_0000 ~ 0x100a_0000.

Please refer to the document DDI0437F_mali_400_mp_gpu_r1p1_trm.pdf.

16.5 Timing Diagram

The GPU only has a clock input, which is called gpu_aclk. Gpu_aclk is generated from the CRU module as shows below

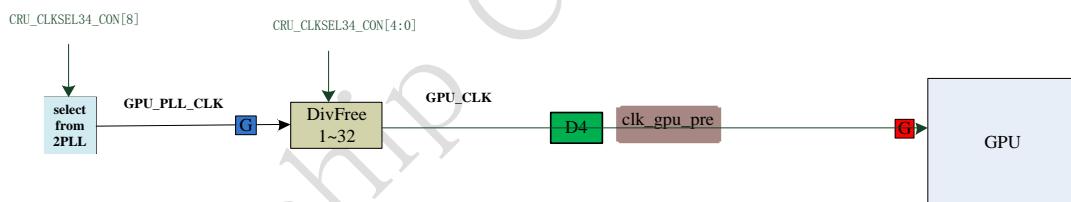


Fig. 16-2 GPU_ACLK generate block diagram

We can configure CPLL, GPLL and CRU register CRU_CLKSEL34_CON to control the gpu_aclk frequency.

16.6 Interface description

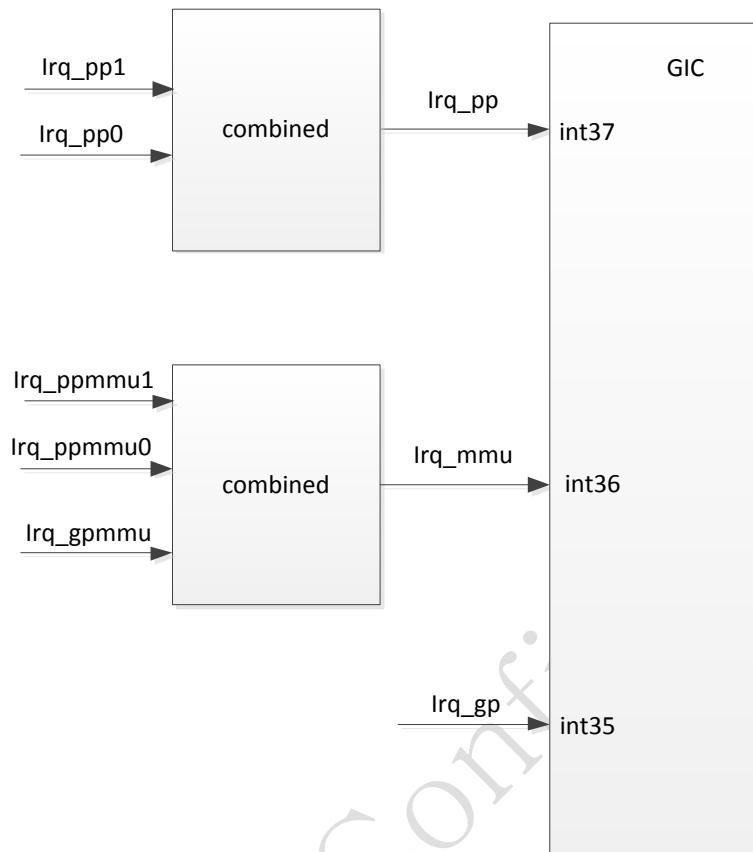


Fig. 16-3 GPU interrupt connection

The GPU now has three interrupt output. `irq_ppmmu` and `irq_gpmmu` is combined to `irq_mmu`.

16.7 Application Notes

Rockchip Confidential

Chapter 17 VPU_Combo

17.1 Overview

VPU_Combo is composed by the H.265 (HEVC) decoder and the H.264 encoder/decoder to realize the high quality video decoding. VPU_Combo is connected to the AHB bus through an AHB slave and the AXI bus through an AXI master. The register configuration is fed into the VPU_Combo through the AHB slave interface while the stream data is transacted between DDR and VPU_Combo through the AXI master interface.

To reduce the area, H.265 decoder and H.264 encoder/decoder not only share several pieces of the internal memories, but also share the bus master and slave interfaces. Therefore VPU_Combo has no any possibility to have the H.265 video decoding and H.264 video encoding/decoding to work simultaneously. As the internal memory sharing utilizes the time division mechanism based on one frame of the video, VPU_Combo can be free to choose H.264 decoding/encoding mode or H.265 decoding mode after one frame is completed.

In order to improve large data transaction performance, VPU embeds MMU (memory management unit) and supports the cacheable bus operation.

VPU_Combo supports the next-generation video coding standard HEVC (High Efficiency Video Coding, aka H.265) full-HD decoding up to 60fps. With HEVC standard, the data compression ratio can be doubled compared to H.264/MEPG-4 at the same video quality or alternatively to provide substantially improved video at the same bit rate.

17.1.1 Features

- Supports HEVC Main Profile up to Level 4.1 High Tier: 1920x1080@60 fps
 - MMU embedded
 - Supports frame timeout interrupt, frame finish interrupt and bitstream error interrupt
 - Supports RLC write mode, RLC mode and Normal Mode
- Supports decoding of the following standards
 - Error detection and concealment support for all video formats
 - Output data structure after decoder is YCbCr 4:2:0 semi-planar to have more efficient bus usage
 - H.264 up to HP level 5.1 : 1080p@60fps (1920x1088)
 - SVC: base layer only for Baseline/High Profile
 - MVC: Stereo High
 - MPEG-4: Simple Profile up to Level 6; Advanced Profile up to Level 5
 - MPEG-2: Main Profile up to High Level
 - MPEG-1: Main Profile up to High Level
 - H.263: Level 10-70 for Profile 0, and image size up to 720x576
 - Sorenson Spark: Bistream version 0 and 1
 - JPEG: Baseline interleaved, and supports ROI (region of image) decode
 - RV: RV8, RV9, V10
 - VP7: up to version 3
 - VP8: version 2(webM)
 - WebP
 - DivX: DivX Home Theater Profile Qualified, DivX4, DivX5, DivX6
 - For H.264, Image cropping not supported
- Built-in post processor in H.264 decoder supports:
 - Stand-alone mode: rotation, deinterlace, RGB conversion, scaling, dithering and alpha blending

- Pipe-lining mode:, RGB conversion, scaling, dithering and alpha blending
- Supports encoding of the following standards:
 - H.264: up to HP level 4.1
 - JPEG: Baseline (DCT sequential)
- Built-in pre-processor in H.264 encoder supports:
 - Cropping, rotation, YCbCr conversion

17.2 Block Diagram

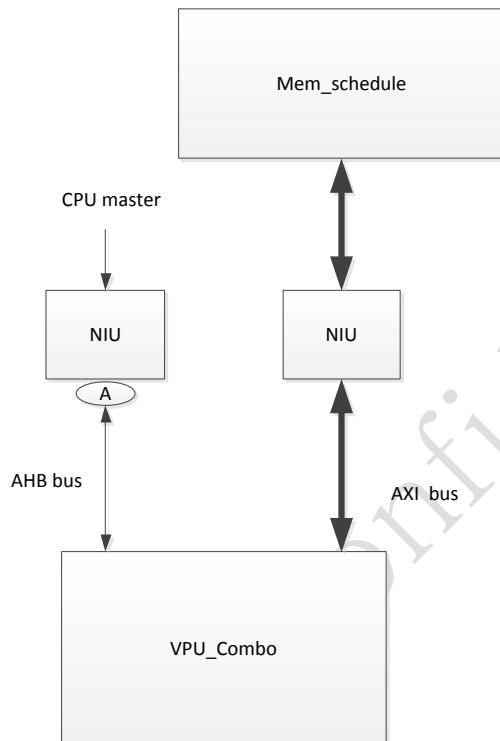


Fig.17-1VPU Combo in SOC

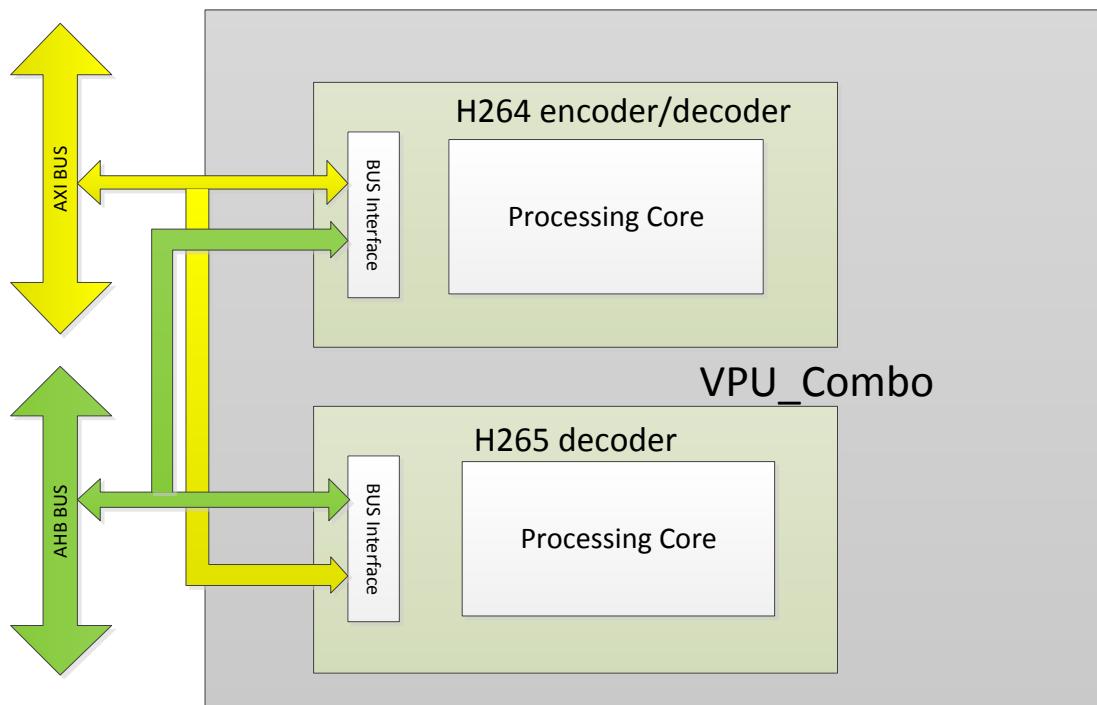


Fig.17-2VPU Combo Block Diagram

As shown in Fig,CPU accesses to HEVC register bank through 32-bit AHB bus. Bitstream and compressed video data are fed into HEVC core though 128-bit AXI read channel, and after several steps of decoding process, decoded pictures are transferred to designated location in the DDR through 64-bit AXI write channel.

CPU accesses to H264 register bank through 32-bit AHB bus. Video data are fed into H.264 core though 64-bit AXI read channel, and after several steps of decoding process, process results are transferred to designated location in the DDR through 64-bit AXI write channel.

17.3 Function Description

17.3.1 HEVC Standard

High Efficiency Video Coding (HEVC) is a video compression standard, a successor to H.264/MPEG-4 AVC (Advanced Video Coding), that was jointly developed by the ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) as ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.265. MPEG and VCEG established a Joint Collaborative Team on Video Coding (JCT-VC) to develop the HEVC standard.

HEVC was designed to substantially improve coding efficiency compared to H.264/MPEG-4 AVC HP, i.e. to reduce bitrate requirements by half with comparable image quality, at the expense of increased computational complexity. HEVC was designed with the goal of allowing video content to have a data compression ratio of up to 1000:1. Depending on the application

requirements HEVC encoders can trade off computational complexity, compression rate, robustness to errors, and encoding delay time. Two of the key features where HEVC was improved compared to H.264/MPEG-4 AVC was support for higher resolution video and improved parallel processing methods.

17.3.2 HEVC Coding Tools

1. Coding tree unit

HEVC replaces macroblocks, which were used with previous standards, with Coding Tree Units (CTUs) which can use a larger block structures of up to 64x64 pixels and can better sub-partition the picture into variable sized structures. HEVC initially divides the picture into CTUs which can be 64x64, 32x32, or 16x16 with a larger pixel block size usually increasing the coding efficiency.

2. Parallel processing tools

Tiles allow for the picture to be divided up into a grid of rectangular regions that can independently be decoded/encoded and the main purpose of tiles is to allow for parallel processing. Tiles can be independently decoded and can even allow for random access to specific regions of a picture in a video stream.

Wavefront parallel processing (WPP) is when a slice is divided into rows of CTUs in which the first row is decoded normally but each additional row requires that decisions be made in the previous row. WPP has the entropy encoder use information from the preceding row of CTUs and allows for a method of parallel processing that may allow for better compression than tiles.

Tiles and WPP are allowed but are optional. If tiles are present they must be at least 64 pixels high and 256 pixels wide with a level specific limit on the number of tiles allowed.

Slices can for the most part be decoded independently from each other with the main purpose of tiles being re-synchronization in case of data loss in the video stream. Slices can be defined as self-contained in that prediction is not made across slice boundaries. When in-loop filtering is done on a picture though information across slice boundaries may be required.[1] Slices are CTUs decoded in the order of the raster scan and different coding types can be used for slices such as I types, P types, or B types.

Dependent slices can allow for data related to tiles or WPP to be accessed more quickly by the system than if the entire slice had to be decoded.[1] The main purpose of dependent slices is to allow for low delay video encoding due to its lower latency.

3. Entropy coding

HEVC uses a context-adaptive binary arithmetic coding (CABAC) algorithm that is fundamentally similar to CABAC in H.264/MPEG-4 AVC. CABAC is the only entropy encoder method that is allowed in HEVC while there are two entropy encoder methods allowed by H.264/MPEG-4 AVC. CABAC and the entropy coding of transform coefficients in HEVC were designed for a higher throughput than H.264/MPEG-4 AVC. For instance, the number of context coded bins have been reduced by 8x and the CABAC bypass-mode has been improved in terms of its design to increase throughput. Another improvement with HEVC is that the dependencies between the coded data has been changed to further increase throughput. Context modeling in HEVC has also been improved so that CABAC can better select a context that increases efficiency when compared to H.264/MPEG-4 AVC.

4. Intra prediction

HEVC specifies 33 directional modes for intra prediction compared to the 8 directional modes for intra prediction specified by H.264/MPEG-4 AVC. HEVC also specifies planar and DC intra prediction modes.[1] The intra prediction modes use data from neighboring prediction blocks that have been previously decoded.

5. Motion compensation

For the interpolation of fractional luma sample positions HEVC uses separable application of one-dimensional half-sample interpolation with an 8-tap filter or quarter-sample interpolation with a 7-tap filter while, in comparison, H.264/MPEG-4 AVC uses a two-stage process that first derives values at half-sample positions using separable one-dimensional 6-tap interpolation followed by integer rounding and then applies linear interpolation between values at nearby half-sample positions to generate values at quarter-sample positions.[1] HEVC has improved precision due to the longer interpolation filter and the elimination of the intermediate rounding error. For 4:2:0 video, the chroma samples are interpolated with separable one-dimensional 4-tap filtering to generate eighth-sample precision, while in comparison H.264/MPEG-4 AVC uses only a 2-tap bilinear filter (also with eighth-sample precision).

As in H.264/MPEG-4 AVC, weighted prediction in HEVC can be used either with uni-prediction (in which a single prediction value is used) or bi-prediction (in which the prediction values from two prediction blocks are combined).

6. Motion vector prediction

HEVC defines a signed 16-bit range for both horizontal and vertical motion vectors (MVs). This was added to HEVC at the July 2012 HEVC meeting with the mvLX variables. HEVC horizontal/vertical MVs have a range of -32768 to 32767 which given the quarter pixel precision used by HEVC allows for a MV range of -8192 to 8191.75 luma samples. This compares to H.264/MPEG-4 AVC which allows for a horizontal MV range of -2048 to 2047.75 luma samples and a vertical MV range of -512 to 511.75 luma samples.

HEVC allows for two MV modes which are Advanced Motion Vector Prediction (AMVP) and merge mode. AMVP uses data from the reference picture and can also use data from adjacent prediction blocks. The merge mode allows for the MVs to be inherited from neighboring prediction blocks. Merge mode in HEVC is similar to "skipped" and "direct" motion inference modes in H.264/MPEG-4 AVC but with two improvements. The first improvement is that HEVC uses index information to select one of several available candidates. The second improvement is that HEVC uses information from the reference picture list and reference picture index.

7. Inverse transforms

HEVC specifies four transform units (TUs) sizes of 4x4, 8x8, 16x16, and 32x32 to code the prediction residual. A CTB may be recursively partitioned into 4 or more TUs.[1] TUs use integer basis functions that are similar to the discrete cosine transform (DCT). In addition 4x4 luma transform blocks that belong to an

intra coded region are transformed using an integer transform that is derived from discrete sine transform (DST). This provides a 1% bit rate reduction but was restricted to 4x4 luma transform blocks due to marginal benefits for the other transform cases. Chroma uses the same TU sizes as luma so there is no 2x2 transform for chroma.

8. Loop filters

HEVC specifies two loop filters that are applied sequentially, with the deblocking filter (DBF) applied first and the sample adaptive offset (SAO) filter applied afterwards. Both loop filters are applied in the inter-picture prediction loop, i.e. the filtered image is stored in the decoded picture buffer (DPB) as a reference for inter-picture prediction.

8.1 Deblocking filter

The DBF is similar to the one used by H.264/MPEG-4 AVC but with a simpler design and better support for parallel processing.[1] In HEVC the DBF only applies to a 8x8 sample grid while with H.264/MPEG-4 AVC the DBF applies to a 4x4 sample grid. DBF uses a 8x8 sample grid since it causes no noticeable degradation and significantly improves parallel processing because the DBF no longer causes cascading interactions with other operations. Another change is that HEVC only allows for three DBF strengths of 0 to 2. HEVC also requires that the DBF first apply horizontal filtering for vertical edges to the picture and only after that does it apply vertical filtering for horizontal edges to the picture. This allows for multiple parallel threads to be used for the DBF.

8.2 Sample adaptive offset

The SAO filter is applied after the DBF and is designed to allow for better reconstruction of the original signal amplitudes by applying offsets stored in a lookup table in the bitstream. Per CTB the SAO filter can be disabled or applied in one of two modes: edge offset mode or band offset mode. The edge offset mode operates by comparing the value of a sample to two of its eight neighbors using one of four directional gradient patterns. Based on a comparison with these two neighbors, the sample is classified into one of five categories: minimum, maximum, an edge with the sample having the lower value, an edge with the sample having the higher value, or monotonic. For each of the first four categories an offset is applied. The band offset mode applies an offset based on the amplitude of a single sample. A sample is categorized by its amplitude into one of 32 bands (histogram bins). Offsets are specified for four consecutive of the 32 bands, because in flat areas which are prone to banding artifacts, sample amplitudes tend to be clustered in a small range.[1][135] The SAO filter was designed to increase picture quality, reduce banding artifacts, and reduce ringing artifacts.

17.3.3 MMU

The MMU divides memory into 4KB pages, where each page can be individually configured. For each page the following parameters are specified:

- Address translation of virtual memory, this enables the processor to work using address that differ from the physical address in the memory system.
- The permitted types of accesses to that page. Each page can permit read,

write, both, or none.

The MMU use 2-level page table structure:

1. The first level, the page directory consists of 1024 directory table entries(DTEs), each pointing to a page table.
2. The second level, the page table consists of 1024 page table entries(PTEs), each pointing to a page in memory.

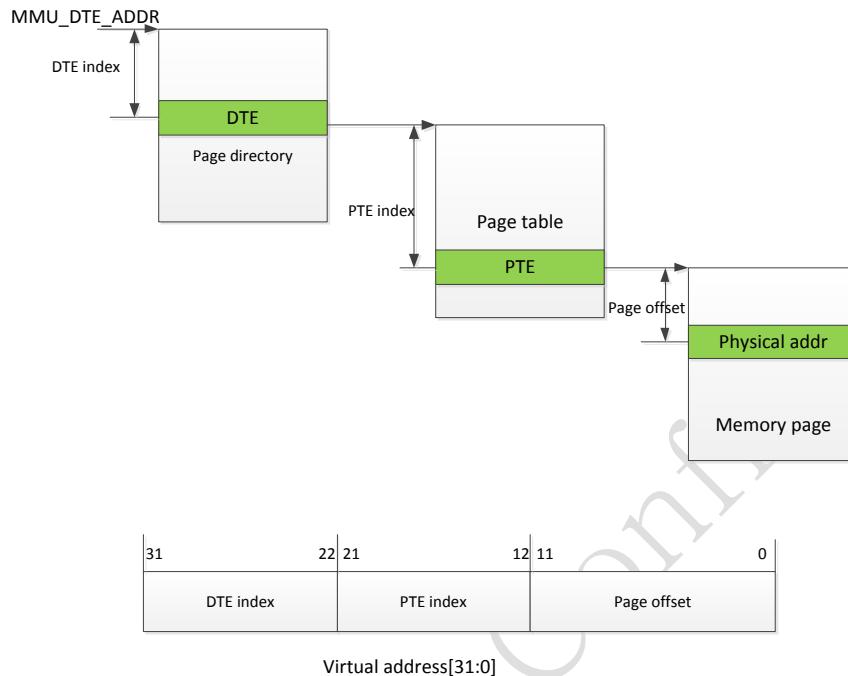


Fig.17-3structure of two-level page table

17.3.4 HEVC Working Mode

There are three working modes to be selected for HEVC decoder: RLC Mode, RLC Write Mode, Normal Mode.

The key differences among three working modes are whether CABAC module and Post-CABAC module are involved into the hardware decoding process.

For RLC mode, CABAC are bypassed and the input bitstream to the Post-CABAC module should be already decoded.

For RLC write mode, the decoded results by CABAC are output to the DDR, and the following decoding processes are stopped.

As for the normal mode, all the modules are involved into the decoding process, and complete decoding results are output. Normally, this mode should be selected.

17.3.5 H.264 decoder

The input of the decoder is H.264 standard bit stream in either plain NAL unit format or byte stream format. The input format in use will be automatically detected. The H.264 video encoding allows the use of multiple reference pictures, which means that the decoding order of the pictures may be different.

from their display order. The decoder can perform internally the display reordering of the decoded pictures or it can skip this and output all the pictures as soon as they are decoded.

The decoder has two operating modes: in the primary mode the HW performs entropy decoding, and in the secondary mode SW performs entropy decoding. Secondary mode is used in H.264 ASO or Slice Group stream decoding.

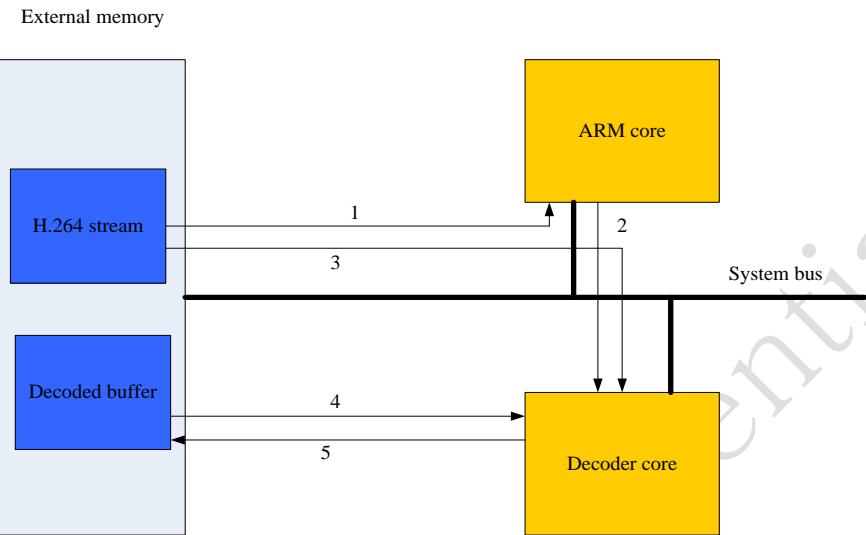


Fig.17-4 Dataflow of HW performs entropy decoding in video decoder

The dataflow of HW performs entropy decoding is as Fig.17-4 shown. The decoder software (SW) starts decoding the first picture by parsing the stream headers (1). Software then setups the hardware control registers (picture size, stream start address etc.) and enables the hardware (2). Hardware decodes the picture by reading stream (3) and the reference pictures (required for inter picture decoding)(4) from the external memory. Hardware writes the decoded output picture to memory one macroblock at a time (5). When the picture has been fully decoded or the hardware has run out of stream data, it gives an interrupt with a proper status flag and provides stream and address for software to continue and returns to initial state.

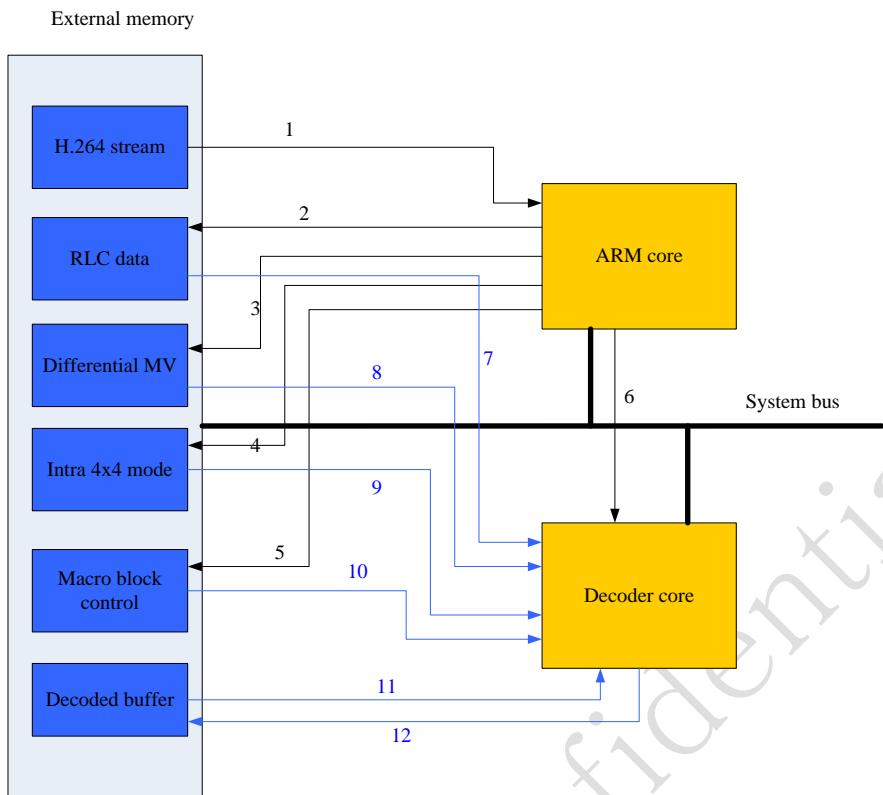


Fig.17-5 Dataflow of SW performs entropy decoding in video decoder

SW entropy decoding mode (RLC mode) changes the input data format that is transferred from SW to HW. The dataflow of this mode is as Fig.17-5. In this case the decoder software starts decoding the first picture by parsing the stream headers (1), and by performing entropy decoding. Software then writes the following items to external memory:

Run-length-code (RLC) data (2)

Differential motion vectors (3)

Intra 4x4 prediction modes (4)

Macroblock control data (5)

Last step for the software is to write the hardware control registers and to enable the hardware (6).

Hardware decodes the picture by buffering control data for several macroblocks at a time, and reading then appropriate amount of RLC data, differential motion vectors and intra modes depending on each macroblock type (7)-(10). For the rest of the decoding process (11)-(12), the functionality is identical to the HW entropy decoding mode. When the picture has been fully decoded, hardware can raise an interrupt and write the status bits in the status register.

17.4 Register description

This section describes the control/status registers of the VPU_Combo.

If HEVC base is vpu_base, then the HEVC base address is vpu_base. The read MMU master base address is vpu_base+0x440, the write MMU base address is vpu_base + 0x480, the cache control base address is vpu_base + 0x400.

If H.264 base is vpu_base,then VEPU base address is vpu_base,VDPU base address is vpu_base + 0x400, MMU base address is vpu_base+0x800, the VDPU cache control base address is vpu_base + 0xc00.

17.4.1 VDP_HEVC Register Summary

Name	Offset	Size	Reset Value	Description
hevc_swreg0_id	0x0000	W	0x68761100	ID register (read only)
hevc_swreg1_int	0x0004	W	0x00200022	interrupt and decoder enable register
hevc_swreg2_sysctrl	0x0008	W	0x00000000	Data input and output endian setting and sys ctrl
hevc_swreg3_picpar	0x000c	W	0x00000000	picture parameters
hevc_swreg4_strm_rlc_base	0x0010	W	0x00000000	the stream or rlc data base address
hevc_swreg5_stream_rlc_len	0x0014	W	0x00000000	amount of stream bytes or rlc data byte in the input buffer or the
hevc_swreg6_cabactbl_base	0x0018	W	0x00000000	the base address of cabac table
hevc_swreg7_decout_base	0x001c	W	0x00000000	base address of decoder output picture base address
hevc_swreg8_y_virstride	0x0020	W	0x00000000	the ouput picture y virtual stride
hevc_swreg9_yuv_virstride	0x0024	W	0x00000000	the ouput picture yuv virtual stride
hevc_swreg10_refer0_base	0x0028	W	0x00000000	base address for reference picture index 0
hevc_swreg11_refer1_base	0x002c	W	0x00000000	base address for reference picture index 1
hevc_swreg12_refer2_base	0x0030	W	0x00000000	base address for reference picture index 2
hevc_swreg13_refer3_base	0x0034	W	0x00000000	base address for reference picture index 3
hevc_swreg14_refer4_base	0x0038	W	0x00000000	base address for reference picture index 4
hevc_swreg15_refer5_base	0x003c	W	0x00000000	base address for reference picture index 5
hevc_swreg16_refer6_base	0x0040	W	0x00000000	base address for reference picture index 6
hevc_swreg17_refer7_base	0x0044	W	0x00000000	base address for reference picture index 7
hevc_swreg18_refer8_base	0x0048	W	0x00000000	base address for reference picture index 8
hevc_swreg19_refer9_base	0x004c	W	0x00000000	base address for reference picture index 9
hevc_swreg20_refer10_base	0x0050	W	0x00000000	base address for reference picture index 10
hevc_swreg21_refer11_base	0x0054	W	0x00000000	base address for reference picture index 11
hevc_swreg22_refer12_base	0x0058	W	0x00000000	base address for reference picture index 12

Name	Offset	Size	Reset Value	Description
hevc_swreg23_refer_13_base	0x005c	W	0x00000000	base address for reference picture index 13
hevc_swreg24_refer_14_base	0x0060	W	0x00000000	base address for reference picture index 14
hevc_swreg25_refer_0_poc	0x0064	W	0x00000000	the poc of reference picture index 0
hevc_swreg26_refer_1_poc	0x0068	W	0x00000000	the poc of reference picture index 1
hevc_swreg27_refer_2_poc	0x006c	W	0x00000000	the poc of reference picture index 2
hevc_swreg28_refer_3_poc	0x0070	W	0x00000000	the poc of reference picture index 3
hevc_swreg29_refer_4_poc	0x0074	W	0x00000000	the poc of reference picture index 4
hevc_swreg30_refer_5_poc	0x0078	W	0x00000000	the poc of reference picture index 5
hevc_swreg31_refer_6_poc	0x007c	W	0x00000000	the poc of reference picture index 6
hevc_swreg32_refer_7_poc	0x0080	W	0x00000000	the poc of reference picture index 7
hevc_swreg33_refer_8_poc	0x0084	W	0x00000000	the poc of reference picture index 8
hevc_swreg34_refer_9_poc	0x0088	W	0x00000000	the poc of reference picture index 9
hevc_swreg35_refer_10_poc	0x008c	W	0x00000000	the poc of reference picture index 10
hevc_swreg36_refer_11_poc	0x0090	W	0x00000000	the poc of reference picture index 11
hevc_swreg37_refer_12_poc	0x0094	W	0x00000000	the poc of reference picture index 12
hevc_swreg38_refer_13_poc	0x0098	W	0x00000000	the poc of reference picture index 13
hevc_swreg39_refer_14_poc	0x009c	W	0x00000000	the poc of reference picture index 14
hevc_swreg40_cur_poc	0x00a0	W	0x00000000	the poc of cur picture
hevc_swreg41_rlcwrite_base	0x00a4	W	0x00000000	the base address or rlcwrite base addr
hevc_swreg42_pps_base	0x00a8	W	0x00000000	the base address of pps
hevc_swreg43_rps_base	0x00ac	W	0x00000000	the base address of rps
hevc_swreg44_cabac_error_en	0x00b0	W	0x00000000	cabac error enable config
hevc_swreg45_cabac_error_status	0x00b4	W	0x00000000	cabac error status
hevc_swreg46_cabac_error_ctu	0x00b8	W	0x00400000	cabac error ctu
hevc_swreg47_sao_ctu_position	0x00bc	W	0x00000000	sao ctu position

Name	Offset	Size	Reset Value	Description
hevc_swreg64_performace_cycle	0x0100	W	0x00000000	hevc performance cycle
hevc_swreg65_axi_d dr_rdata	0x0104	W	0x00000000	axi ddr read data num
hevc_swreg66_axi_d dr_wdata	0x0108	W	0x00000000	axi ddr write data number

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

17.4.2 VDP_HEVC Detail Register Description

hevc_swreg0_id

Address: Operational Base + offset (0x0000)

ID register (read only)

Bit	Attr	Reset Value	Description
31:16	RO	0x6876	prod_num product number The ascii code of "hv", which is 0x6876
15	RO	0x0	reserved
14	RW	0x0	codec_flag codec flag 0: only dec 1: dec + enc
13	RO	0x0	reserved
12	RW	0x1	profile hevc profile 0: Main 1: Main10
11:9	RO	0x0	reserved
8	RO	0x1	level level 0: FHD 1: UHD
7:0	RO	0x00	minor_ver minor version minor version

hevc_swreg1_int

Address: Operational Base + offset (0x0004)

interrupt and decoder enable register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_softreset_rdy when it is 1'b1, it says that softreset has been done

Bit	Attr	Reset Value	Description
21	RW	0x1	sw_force_softreset_valid when sw_force_softreset_valid is 1'b1, sw_softrst_en will always be valid to the system no matter that whether the axi bus is idle; when sw_force_softreset_valid is 1'b0, sw_softrst_en will only be valid when the axi bus is idle.
20	RW	0x0	sw_softrst_en_p softreset enable softreset enable signal write 1 to soft reset, write 0 invalid puls register
19	RO	0x0	reserved
18	RW	0x0	sw_cabu_end_sta cabac decode end status cabac decode end status
17	RW	0x0	sw_colmv_ref_error_sta colmv ref error status colmv ref error status when it is 1'b1, it means that inter module read the invalid dpb frame
16	RO	0x0	reserved
15	RW	0x0	sw_dec_timeout_sta decoder timeout interrupt status When high the decoder has been idling for too long. it will self reset the hardware only when sw_dec_timeout_e is 1'b1, this bit is valid
14	RW	0x0	sw_dec_error_sta status bit of input stream error when high, an error is found in input data stream decoding. It will self reset the hardware
13	RW	0x0	sw_dec_bus_sta bus error status When this bit is high, there is error on the axi bus, it will self reset hardware
12	RW	0x0	sw_dec_rdy_sta decoder ready status when this bit is high, decoder has decoded a picture
11:10	RO	0x0	reserved
9	RW	0x0	sw_dec_irq_raw the raw status of sw_dec_irq the raw status of sw_dec_irq,SW should reset this bit after interrupt is handled

Bit	Attr	Reset Value	Description
8	RO	0x0	sw_dec_irq decoder IRQ when high, decoder requests an interrupt. $sw_dec_irq = sw_dec_irq_raw \&& (sw_dec_irq_dis == 1'b0)$
7	RW	0x0	sw_stmerror_waitdecfifo_empty whether the stream error process wait the decfifo empty when it is 1'b0, the stream error process will no wait the ca2decfifo empty when it is 1'b1, the stream error process will wait the ca2decfifo empty
6	RO	0x0	reserved
5	RW	0x1	sw_dec_timeout_e Timeout interrupt enable If enabled HW may return timeout interrupt in case HW gets stucked while decoding picture.
4	RW	0x0	sw_dec_irq_dis decoder IRQ disable When hight, there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt status
3:2	RO	0x0	reserved
1	RW	0x1	sw_dec_clkgate_e decoder dynamic clock gating enable 0 = clock is running for all structures 1 = clock is gated for decoder structures that are not used
0	RW	0x0	sw_dec_e decoder enable Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when picture is processed or stream error is detected or bus error or time out interrupt is given

hevc_swreg2_sysctrl

Address: Operational Base + offset (0x0008)

Data input and output endian setting and sys ctrl

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:12	RW	0x00	sw_strm_start_bit exact bit of stream start exact bit of streamd start word where decoding can be started (asosiates with sw_str_rlc_base)
11	RW	0x0	sw_rlc_mode rlc mode enable 0 = HW decodes video from bit stream 1 = HW decodes video from RLC input data

Bit	Attr	Reset Value	Description
10	RW	0x0	sw_rlc_mode_direct_write cabac decode output direct write cabac decode output direct write enable when this bit is enable , all the module other than cabac and busifd are not work
9	RO	0x0	reserved
8	RW	0x0	sw_out_cbcr_swap output cbcr swap 1'b0: cb(u) is in the lower address, cr(v) is in the higher address 1'b1: cb(u) is in the higher address, cr(v) is in the lower address sw_in_cbcr_swap is the same with sw_out_cbcr_swap
7	RW	0x0	sw_out_swap32_e decoder output data and dpb input data 32bit swap may be used for 64 or 128 bit environment 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped
6	RW	0x0	sw_out_endian dec output data and colmv , dpb data and colmv input endian 0 = little endian 1 = big endian for litter enadian , a data 0x12345678, 0x78 is stored in lower address, 0x12 is stored in higher address
5	RW	0x0	sw_str_swap64_e stream 64bit data swap may be used for 128 bit environment 0 = no swapping of 64 bit words 1 = 64 bit data words are swapped
4	RW	0x0	sw_str_swap32_e stream 32bit data swap may be used for 64 or 128 bit environment 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped
3	RW	0x0	sw_str_endian stream data input endian mode 0 = little endian 1 = big endian for litter enadian , a data 0x12345678, 0x78 is stored in lower address, 0x12 is stored in higher address
2	RW	0x0	sw_in_swap64_e input 64bit data swap for other than stream and dpb data may be used for 128 bit environment 0 = no swapping of 64 bit words 1 = 64 bit data words are swapped

Bit	Attr	Reset Value	Description
1	RW	0x0	sw_in_swap32_e input 32bit data swap for other than stream and dpb data may be used for 64 or 128 bit environment 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped
0	RW	0x0	sw_in_endian decoder input endian mode for other than stream and dpb data 0 = little endian 1 = big endian for litter enadian , a data 0x12345678, 0x78 is stored in lower address, 0x12 is stored in higher address

hevc_swreg3_picpar

Address: Operational Base + offset (0x000c)
picture parameters

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:21	RW	0x000	sw_slice_num slice number in a frame slice number in a frame (0~199, when it is 0, it real means 1 slice in a frame) just only used for rps read. 2013.11.27 change the meaning from count from 1, so it will be in 1~200 2013.11.30 sw_slice_num max value is change to 600, so sw_slice_num expand to 10bit
20:12	RW	0x000	sw_uv_hor_virstride Field0000 Abstract picture horizontal virtual stride (the unit is 128bit) the max is $(4096 \times 1.5 + 128) / 16 = 0x188$ suggest this register to config to even for advance ddr performance
11:9	RO	0x0	reserved
8:0	RW	0x000	sw_y_hor_virstride picture horizontal virtual stride picture horizontal virtual stride (the unit is 128bit) the max is $(4096 \times 1.5 + 128) / 16 = 0x188$ suggest this register to config to even for advance ddr performance

hevc_swreg4_strm_rlc_base

Address: Operational Base + offset (0x0010)
the stream or rlc data base address

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_strm_rlc_base the stream or rlc data base address when swreg2.sw_rlc_mode =1, it is base address for rlc data when swreg2.sw_rlc_mode =0, it is base address for stream , after a frame is decoded ready or error (stream error , time out , bus error) , it is the last address of the stream the address should 128bit align
3:0	RO	0x0	reserved

hevc_swreg5_stream_rlc_len

Address: Operational Base + offset (0x0014)

amount of stream bytes or rlc data byte in the input buffer or the

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RW	0x0000000	sw_stream_len amount of stream (unit is 8bit) in the input buffer amount of stream 8bits in the input buffer the max of sw_stream_len : $4096 \times 2304 \times 1.5 \times 1.5 = 0x1440000$ 128bits unit: $0x1440000 / 16 = 0x144000$ it is count from 0 2013.10.15 change to 23bit for zty's suggestion 2013.10.28, amount of stream data bytes in input buffer. it is count from 1, change to 27bits

hevc_swreg6_cabactbl_base

Address: Operational Base + offset (0x0018)

the base address of cabac table

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_cabactbl_base the base address of cabac table the base address of cabac table the address should 128bit align
3:0	RO	0x0	reserved

hevc_swreg7_decout_base

Address: Operational Base + offset (0x001c)

base address of decoder output picture base address

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_decout_base base address of decoder output picture addr base address of decoder output picture the address should be 128bit align
3:0	RO	0x0	reserved

hevc_swreg8_y_virstride

Address: Operational Base + offset (0x0020)
the ouput picture y virtual stride

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	sw_y_virstride the output picture y virtual stride the output picture y virtual stride (the unit is 128bit) the max: $(4096 \times 1.5 + 128) \times 2304 = 0xdc8000$ we can know the sw_mvout_base = sw_decout_base + (sw_y_virstride << 4)

hevc_swreg9_yuv_virstride

Address: Operational Base + offset (0x0024)
the ouput picture yuv virtual stride

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:0	RW	0x0000000	sw_yuv_virstride the ouput picture yuv virtual stride the output picture yuv virtual stride (the unit is 128bit) the max : $(4096 \times 1.5 + 128) \times 2304 \times 1.5 = 0x14ac000$ we can know the sw_mvout_base = sw_decout_base + (sw_yuv_virstride << 4)

hevc_swreg10_refer0_base

Address: Operational Base + offset (0x0028)
base address for reference picture index 0

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer0_base base address for reference picture index0 base address for reference picture index 0 (the address should be 128bit align)
3:0	RW	0x0	sw_ref_valid_0_3 valid flag for picture index 0 ~3 valid flag for picture index 0 ~3

hevc_swreg11_refer1_base

Address: Operational Base + offset (0x002c)

base address for reference picture index 1

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer1_base base address for reference picture index 1 base address for reference picture index 1 (the address should be 128bit align)
3:0	RW	0x0	sw_ref_valid_4_7 valid flag for picture index 4 ~7 valid flag for picture index 4 ~7

hevc_swreg12_refer2_base

Address: Operational Base + offset (0x0030)

base address for reference picture index 2

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer2_base base address for reference picture index 2 base address for reference picture index 2 (the address should be 128bit align)
3:0	RW	0x0	sw_ref_valid_8_11 valid flag for picture index 8~11 valid flag for picture index 8~11

hevc_swreg13_refer3_base

Address: Operational Base + offset (0x0034)

base address for reference picture index 3

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer3_base base address for reference picture index 3 base address for reference picture index 3 (the address should be 128bit align)
3	RO	0x0	reserved
2:0	RW	0x0	sw_ref_valid_12_14 valid flag for picture index 12~14 valid flag for picture index 12~14

hevc_swreg14_refer4_base

Address: Operational Base + offset (0x0038)

base address for reference picture index 4

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer4_base base address for reference picture index 4 base address for reference picture index 4(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg15_refer5_base

Address: Operational Base + offset (0x003c)

base address for reference picture index 5

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer5_base base address for reference picture index 5 base address for reference picture index 5(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg16_refer6_base

Address: Operational Base + offset (0x0040)
base address for reference picture index 6

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer6_base base address for reference picture index 6 base address for reference picture index 6(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg17_refer7_base

Address: Operational Base + offset (0x0044)
base address for reference picture index 7

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer7_base base address for reference picture index 7 base address for reference picture index 7(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg18_refer8_base

Address: Operational Base + offset (0x0048)
base address for reference picture index 8

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer8_base base address for reference picture index 8 base address for reference picture index 8(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg19_refer9_base

Address: Operational Base + offset (0x004c)
base address for reference picture index 9

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer9_base base address for reference picture index 9 base address for reference picture index 9(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg20_refer10_base

Address: Operational Base + offset (0x0050)
 base address for reference picture index 10

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer10_base base address for reference picture index 10 base address for reference picture index 10(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg21_refer11_base

Address: Operational Base + offset (0x0054)
 base address for reference picture index 11

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer11_base base address for reference picture index 11 base address for reference picture index 11(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg22_refer12_base

Address: Operational Base + offset (0x0058)
 base address for reference picture index 12

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer12_base base address for reference picture index 12 base address for reference picture index 12(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg23_refer13_base

Address: Operational Base + offset (0x005c)
 base address for reference picture index 13

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer13_base base address for reference picture index 13 base address for reference picture index 13(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg24_refer14_base

Address: Operational Base + offset (0x0060)
 base address for reference picture index 14

Bit	Attr	Reset Value	Description
31:4	RW	0x0000000	sw_refer14_base base address for reference picture index 14 base address for reference picture index 14(the address should be 128bit align)
3:0	RO	0x0	reserved

hevc_swreg25_refer0_poc

Address: Operational Base + offset (0x0064)
the poc of reference picture index 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer0_poc the poc of reference picture index 0 the poc of reference picture index 0

hevc_swreg26_refer1_poc

Address: Operational Base + offset (0x0068)
the poc of reference picture index 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer1_poc the poc of reference picture index 1 the poc of reference picture index 1

hevc_swreg27_refer2_poc

Address: Operational Base + offset (0x006c)
the poc of reference picture index 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer2_poc the poc of reference picture index 2 the poc of reference picture index 2

hevc_swreg28_refer3_poc

Address: Operational Base + offset (0x0070)
the poc of reference picture index 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer3_poc the poc of reference picture index 3 the poc of reference picture index 3

hevc_swreg29_refer4_poc

Address: Operational Base + offset (0x0074)
the poc of reference picture index 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer4_poc the poc of reference picture index 4 the poc of reference picture index 4

hevc_swreg30_refer5_poc

Address: Operational Base + offset (0x0078)
the poc of reference picture index 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer5_poc the poc of reference picture index 5 the poc of reference picture index 5

hevc_swreg31_refer6_poc

Address: Operational Base + offset (0x007c)

the poc of reference picture index 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer6_poc the poc of reference picture index 6 the poc of reference picture index 6

hevc_swreg32_refer7_poc

Address: Operational Base + offset (0x0080)

the poc of reference picture index 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer7_poc the poc of reference picture index 7 the poc of reference picture index 7

hevc_swreg33_refer8_poc

Address: Operational Base + offset (0x0084)

the poc of reference picture index 8

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer8_poc the poc of reference picture index 8 the poc of reference picture index 8

hevc_swreg34_refer9_poc

Address: Operational Base + offset (0x0088)

the poc of reference picture index 9

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer9_poc the poc of reference picture index 9 the poc of reference picture index 9

hevc_swreg35_refer10_poc

Address: Operational Base + offset (0x008c)

the poc of reference picture index 10

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer10_poc the poc of reference picture index 10 the poc of reference picture index 10

hevc_swreg36_refer11_poc

Address: Operational Base + offset (0x0090)

the poc of reference picture index 11

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer11_poc the poc of reference picture index 11 the poc of reference picture index 11

hevc_swreg37_refer12_poc

Address: Operational Base + offset (0x0094)

the poc of reference picture index 12

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer12_poc the poc of reference picture index 12 the poc of reference picture index 12

hevc_swreg38_refer13_poc

Address: Operational Base + offset (0x0098)

the poc of reference picture index 13

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer13_poc the poc of reference picture index 13 the poc of reference picture index 13

hevc_swreg39_refer14_poc

Address: Operational Base + offset (0x009c)

the poc of reference picture index 14

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer14_poc the poc of reference picture index 14 the poc of reference picture index 14

hevc_swreg40_cur_poc

Address: Operational Base + offset (0x00a0)

the poc of cur picture

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_cur_poc the poc of the cur picture the poc of the cur picture

hevc_swreg41_rlcwrite_base

Address: Operational Base + offset (0x00a4)

the base address or rlcwrite base addr

Bit	Attr	Reset Value	Description
31:3	RW	0x00000000	sw_rlcwrite_base the base address of rlcwrite the base address of rlcwrite(the address should 64bit align) cabac output write to this rlcwrite base address when sw_rlc_mode_direct_write in swreg2_sysctrl is valid

Bit	Attr	Reset Value	Description
2:0	RO	0x0	reserved

hevc_swreg42_pps_base

Address: Operational Base + offset (0x00a8)
the base address of pps

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_pps_base the base address of pps the base address of pps (the address should 128bit align) it is for storing sps(sequence parameter set) and pps(picture parameter set)
3:0	RO	0x0	reserved

hevc_swreg43_rps_base

Address: Operational Base + offset (0x00ac)
the base address of rps

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_rps_base rps base address rps(reference picture set) base address (the address should 128bit align)
3:0	RO	0x0	reserved

hevc_swreg44_cabac_error_en

Address: Operational Base + offset (0x00b0)
cabac error enable config

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:0	RW	0x00000000	sw_cabac_error_e cabac error enable regs cabac error enable regs

hevc_swreg45_cabac_error_status

Address: Operational Base + offset (0x00b4)
cabac error status

Bit	Attr	Reset Value	Description
31:28	RW	0x0	sw_colmv_error_ref_picidx colmv error ref picidx when sw_colmv_ref_error_sta is 1'b1, these bits are used for tell which dpb frame is invalid but is read by inter module
27:0	RW	0x00000000	sw_cabac_error_status cabac error status cabac error status

hevc_swreg46_cabac_error_ctu

Address: Operational Base + offset (0x00b8)
cabac error ctu

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:16	RW	0x40	sw_streamfifo_space2full stream fifo space to full It is for debug use, to tell the stream fifo space to full
15:8	RW	0x00	sw_cabac_error_ctu_yoffset cabac error ctu yoffset cabac error ctu yoffset
7:0	RW	0x00	sw_cabac_error_ctu_xoffset cabac error ctu xoffset cabac error ctu xoffset

hevc_swreg47_sao_ctu_position

Address: Operational Base + offset (0x00bc)
sao ctu position

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:16	RW	0x000	sw_saowr_yoffset saowr y offset saowr y offset , its unit is 4 pixels
15:9	RO	0x0	reserved
8:0	RW	0x000	sw_saowr_xoffet saowr x address offset saowr x address offset, its unit is 128bit

hevc_swreg64_performance_cycle

Address: Operational Base + offset (0x0100)
hevc performance cycle

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_performance_cycle hevc running cycle hevc running cycle if just want to analys a frame performance cycle, should set the register 0 before start a frame

hevc_swreg65_axi_ddr_rdata

Address: Operational Base + offset (0x0104)
axi ddr read data num

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_axi_ddr_rdata axi ddr rdata num axi ddr rdata num, the unit is byte

hevc_swreg66_axi_ddr_wdata

Address: Operational Base + offset (0x0108)
 axi ddr write data number

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_axi_ddr_wdata hevc write data byte num hevc write data byte num

This section describes the control/status registers of the design.

If VPU base is vpu_base, then the VEPU base address is vpu_base, the VDPU base address is vpu_base + 0x400.

The MMU base address is vpu_base+0x800, the VDPU cache control base address is vpu_base + 0xc00.

The table below is the description for vepu and vdpu and mmu and pre-cache.

17.4.3 VDP_HEVC MMU Register Summary

Name	Offset	Size	Reset Value	Description
hevc_mmu_DTE_DR	0x00000	W	0x00000000	MMU current page Table address
hevc_mmu_STATUS	0x00004	W	0x00000018	MMU status register
hevc_mmu_COMMAND	0x00008	W	0x00000000	MMU command register
hevc_mmu_PAGE_FAULT_ADDR	0x0000c	W	0x00000000	MMU logical address of last page fault
hevc_mmu_ZAP_ON_E_LINE	0x00010	W	0x00000000	MMU Zap cache line register
hevc_mmu_INT_RAWSTAT	0x00014	W	0x00000000	MMU raw interrupt status register
hevc_mmu_INT_CLEAR	0x00018	W	0x00000000	MMU raw interrupt status register
hevc_mmu_INT_MASK	0x0001c	W	0x00000000	MMU raw interrupt status register
hevc_mmu_INT_STATUS	0x00020	W	0x00000000	MMU raw interrupt status register
hevc_mmu_AUTO_GATING	0x00024	W	0x00000001	mmu auto gating

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

17.4.4 VDP_HEVC MMU Detail Register Description

hevc_mmu_DTE_ADDR

Address: Operational Base + offset (0x00000)

MMU current page Table address

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MMU_DTE_ADDR mmu dte base addr mmu dte base addr , the address must be 4kb aligned

hevc_mmu_STATUS

Address: Operational Base + offset (0x00004)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGE_FAULT_BUS_ID Field0000 Abstract Index of master responsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE Field0000 Abstract The direction of access for last page fault: 0 = Read 1 = Write
4	RO	0x1	REPLAY_BUFFER_EMPTY Field0000 Abstract The MMU replay buffer is empty
3	RO	0x1	MMU_IDLE Field0000 Abstract The MMU is idle when accesses are being translated and there are no unfinished translated accesses.
2	RO	0x0	STAIL_ACTIVE Field0001 Abstract MMU stall mode currently enabled. The mode is enabled by command
1	RO	0x0	PAGE_FAULT_ACTIVE Field0000 Abstract MMU page fault mode currently enabled . The mode is enabled by command.
0	RO	0x0	PAGING_ENABLED Field0000 Abstract Paging is enabled

hevc_mmu_COMMAND

Address: Operational Base + offset (0x00008)

MMU command register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	WO	0x0	MMU_CMD Field0000 Abstract MMU_CMD. This can be: 0: MMU_ENABLE_PAGING 1: MMU_DISABLE_PAGING 2: MMU_ENABLE_STALL 3: MMU_DISABLE_STALL 4: MMU_ZAP_CACHE 5: MMU_PAGE_FAULT_DONE 6: MMU_FORCE_RESET

hevc_mmu_PAGE_FAULT_ADDR

Address: Operational Base + offset (0x0000c)

MMU logical address of last page fault

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	PAGE_FAULT_ADDR Field0000 Abstract address of last page fault

hevc_mmu_ZAP_ONE_LINE

Address: Operational Base + offset (0x00010)

MMU Zap cache line register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE Field0000 Abstract address to be invalidated from the page table cache

hevc_mmu_INT_RAWSTAT

Address: Operational Base + offset (0x00014)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR Field0000 Abstract read bus error
0	RW	0x0	PAGE_FAULT Field0000 Abstract page fault

hevc_mmu_INT_CLEAR

Address: Operational Base + offset (0x00018)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	WO	0x0	READ_BUS_ERROR Field0000 Abstract read bus error
0	WO	0x0	PAGE_FAULT Field0000 Abstract page fault

hevc_mmu_INT_MASK

Address: Operational Base + offset (0x0001c)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR Field0000 Abstract read bus error enable an interrupt source if the corresponding mask bit is set to 1
0	RW	0x0	PAGE_FAULT Field0000 Abstract page fault enable an interrupt source if the corresponding mask bit is set to 1

hevc_mmu_INT_STATUS

Address: Operational Base + offset (0x00020)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR Field0000 Abstract read bus error
0	RO	0x0	PAGE_FAULT Field0000 Abstract page fault

hevc_mmu_AUTO_GATING

Address: Operational Base + offset (0x00024)

mmu auto gating

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_gating mmu auto gating when it is 1'b1, the mmu will auto gating it self

17.4.5 VDP_HEVC_Pref_cache Register Summary

Name	Offset	Size	Reset Value	Description

Name	Offset	Size	Reset Value	Description
pref_cache_VERSIO N	0x00000	W	0xcac20101	VERSION register
pref_cache_SIZE	0x00004	W	0x07110206	L2 cache SIZE
pref_cache_STATUS	0x00008	W	0x00000000	Status register
pref_cache_COMMAN D	0x00010	W	0x00000000	Command setting register
pref_cache_CLEAR_P AGE	0x00014	W	0x00000000	clear page register
pref_cache_MAX_RE ADS	0x00018	W	0x0000001c	maximum read register
pref_cache_ENABLE	0x0001c	W	0x00000003	enables cacheable accesses and cache read allocation
pref_cache_PERFCNT _SRC0	0x00020	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT _VAL0	0x00024	W	0x00000000	performance counter 0 value register
pref_cache_PERFCNT _SRC1	0x00028	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT _VAL1	0x0002c	W	0x00000000	performance counter 1 value register

Notes: **Size:** **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

17.4.6 VDP_HEVC Pref_cache Detail Register Description

pref_cache_VERSION

Address: Operational Base + offset (0x00000)

VERSION register

Bit	Attr	Reset Value	Description
31:16	RO	0xcac2	PRODUCT_ID Field0000 Abstract Field0000 Description
15:8	RO	0x01	VERSION_MAJOR Field0000 Abstract Field0000 Description
7:0	RO	0x01	VERSION_MINOR Field0000 Abstract Field0000 Description

pref_cache_SIZE

Address: Operational Base + offset (0x00004)

L2 cache SIZE

Bit	Attr	Reset Value	Description
31:24	RO	0x07	External_bus_width Field0000 Abstract Log2 external bus width in bits

Bit	Attr	Reset Value	Description
23:16	RO	0x11	CACHE_SIZE Field0000 Abstract Log2 cache size in bytes
15:8	RO	0x02	ASSOCIATIVITY Field0000 Abstract Log2 associativity
7:0	RO	0x06	LINE_SIZE Field0000 Abstract Log2 line size in bytes

pref_cache_STATUS

Address: Operational Base + offset (0x00008)

Status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	DATA_BUSY Field0000 Abstract set when the cache is busy handling data
0	RW	0x0	CMD_BUSY Field0000 Abstract set when the cache is busy handling commands

pref_cache_COMMAND

Address: Operational Base + offset (0x00010)

Command setting register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	COMMAND Field0000 Abstract The possible command is 1 = Clear entire cache

pref_cache_CLEAR_PAGE

Address: Operational Base + offset (0x00014)

clear page register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	CLEAR_PAGE Field0000 Abstract writing an address, invalidates all lines in that page from the cache

pref_cache_MAX_READS

Address: Operational Base + offset (0x00018)

maximum read register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x1c	MAX_READS Field0000 Abstract Limit the number of outstanding read transactions to this amount

pref_cache_ENABLE

Address: Operational Base + offset (0x0001c)

enables cacheable accesses and cache read allocation

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	sw_cache_clk_disgate cache clk disgate cache clk disgate when it is 1'b0, enable cache clk auto clkgating when it is 1'b1, disable cache clk auto clkgating
2	RW	0x0	sw_readbuffer_counter_reject_en counter reject enable default is 1'b0, for enhance cacheable read performnace in readbuffer. 1'b1: normal origin counter reject
1	RW	0x1	permit_cache_read_allocate cache read allocate 1'b1: permit cache read allocate
0	RW	0x1	permit_cacheable_access cacheable access 1'b1: permit cacheable access

pref_cache_PERFCNT_SRC0

Address: Operational Base + offset (0x00020)

performance counter 0 source register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	PERFCNT_SRC0 Field0000 Abstract This register holds all the possible source values for Performance Counter 0 0: disabled 1: total clock cycles 2: active clock cycles 3: read transactions, master 4: word reads, master 5: read transactions, slave 6: word reads, slave 7: read hit, slave 8: read misses, slave 9: read invalidates, slave 10: cacheable read transactions, slave 11: bad hit nmber, slave

pref_cache_PERFCNT_VAL0

Address: Operational Base + offset (0x00024)

performance counter 0 value register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERFCNT_VAL0 Field0000 Abstract Performance counter 0 value

pref_cache_PERFCNT_SRC1

Address: Operational Base + offset (0x00028)

performance counter 0 source register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	PERFCNT_SRC1 Field0000 Abstract This register holds all the possible source values for Performance Counter 1 0: disabled 1: total clock cycles 2: active clock cycles 3: read transactions, master 4: word reads, master 5: read transactions, slave 6: word reads, slave 7: read hit, slave 8: read misses, slave 9: read invalidates, slave 10: cacheable read transactions, slave 11: bad hit nmber, slave

pref_cache_PERFCNT_VAL1

Address: Operational Base + offset (0x0002c)

performance counter 1 value register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERFCNT_VAL1 Field0000 Abstract Performance counter 1 value

17.4.7 VEPU Register Summary

Name	Offset	Size	Reset Value	Description
VEPU_swreg0	0x00000	W	0x48310000	product ID
VEPU_swreg1	0x00004	W	0x00000000	interrupt control and status
VEPU_swreg2	0x00008	W	0x00000010	axi control
VEPU_swreg3	0x0000c	W	0x00000000	test control register
VEPU_swreg4	0x00010	W	0x00000000	reserverd
VEPU_swreg5	0x00014	W	0x00000000	addr_output_stream

Name	Offset	Size	Reset Value	Description
VEPU_swreg6	0x00018	W	0x00000000	base address for output control
VEPU_swreg7	0x0001c	W	0x00000000	base address for reference luma
VEPU_swreg8	0x00020	W	0x00000000	base address for reference chroma
VEPU_swreg9	0x00024	W	0x00000000	base address for reconstructed luma
VEPU_swreg10	0x00028	W	0x00000000	base address for reconstructed chroma
VEPU_swreg11	0x0002c	W	0x00000000	base addr for input luma
VEPU_swreg12	0x00030	W	0x00000000	base address for input cb
VEPU_swreg13	0x00034	W	0x00000000	base address for input cr
VEPU_swreg14	0x00038	W	0x00000000	enc control
VEPU_swreg15	0x0003c	W	0x00000000	input image control
VEPU_swreg16	0x00040	W	0x00000000	encoder control regsite 0
VEPU_swreg17	0x00044	W	0x00000000	encoder control register 1
VEPU_swreg18	0x00048	W	0x00000000	encoder control register 2
VEPU_swreg19	0x0004c	W	0x00000000	encoder control register 3
VEPU_swreg20	0x00050	W	0x00000000	JPEG control regsite
VEPU_swreg21	0x00054	W	0x00000000	encoder control register 4
VEPU_swreg22	0x00058	W	0x00000000	stream header remainder bits MSB
VEPU_swreg23	0x0005c	W	0x00000000	stream header remainder bits LSB
VEPU_swreg24	0x00060	W	0x00000000	stream buffer limit
VEPU_swreg25	0x00064	W	0x00000000	MAD control register
VEPU_swreg27	0x0006c	W	0x00000000	QP register
VEPU_swreg28	0x00070	W	0x00000001	checkpoint 1 and 2
VEPU_swreg29	0x00074	W	0x00000000	checkpoint 3 and 4
VEPU_swreg30	0x00078	W	0x00000000	checkpoint 5 and 6
VEPU_swreg31	0x0007c	W	0x00000000	checkpoint 7 and 8
VEPU_swreg32	0x00080	W	0x00000000	checkpoint 9 and 10
VEPU_swreg33	0x00084	W	0x00000000	checkpoint word error 1 and 2
VEPU_swreg34	0x00088	W	0x00000000	checkpoint word error 1 and 2
VEPU_swreg35	0x0008c	W	0x00000000	checkpoint word error 1 and 2
VEPU_swreg36	0x00090	W	0x00000000	checkpoint delta QP register
VEPU_swreg37	0x00094	W	0x00000000	rlc control
VEPU_swreg38	0x00098	W	0x00000000	mb control register
VEPU_swreg39	0x0009c	W	0x00000000	Base address for next pic
VEPU_swreg40	0x000a0	W	0x00000000	Stabilization minimum value
VEPU_swreg41	0x000a4	W	0x00000000	Stabilization motion sum
VEPU_swreg42	0x000a8	W	0x00000000	stab_matrix1 and stab_gmv_hor

Name	Offset	Size	Reset Value	Description
VEPU_swreg43	0x000ac	W	0x00000000	stab_matrix2 and stab_gmv_ver
VEPU_swreg44	0x000b0	W	0x00000000	stab_matrix3
VEPU_swreg45	0x000b4	W	0x00000000	stab_matrix4
VEPU_swreg46	0x000b8	W	0x00000000	stab_matrix5
VEPU_swreg47	0x000bc	W	0x00000000	stab_matrix6
VEPU_swreg48	0x000c0	W	0x00000000	stab_matrix7
VEPU_swreg49	0x000c4	W	0x00000000	stab_matrix8
VEPU_swreg50	0x000c8	W	0x00000000	stab_matrix9
VEPU_swreg51	0x000cc	W	0x00000000	cabac_table_addr
VEPU_swreg52	0x000d0	W	0x00000000	Base address for MV output
VEPU_swreg53	0x000d4	W	0x00000000	RGB to YUV conversion coefficien A and B
VEPU_swreg54	0x000d8	W	0x00000000	RGB to YUV conversion coefficien C and E
VEPU_swreg55	0x000dc	W	0x00000000	RGB mask MSB bit
VEPU_swreg56	0x000e0	W	0x00000000	intra area control register
VEPU_swreg57	0x000e4	W	0x00000000	CIR intra control reg
VEPU_swreg58	0x000e8	W	0x00000000	Intra slice bitmap for slices 0..31
VEPU_swreg59	0x000ec	W	0x00000000	Intra slice bitmap for slices32..63
VEPU_swreg60	0x000f0	W	0x00000000	1st ROI area register
VEPU_swreg61	0x000f4	W	0x00000000	Register0061 Abstract
VEPU_swreg62	0x000f8	W	0x00000000	MVC control reg
VEPU_swreg63	0x000fc	W	0x1f522780	Register0063 Abstract
VEPU_swreg64	0x00100	W	0x00000000	JPEG luma quantization 1
VEPU_swreg65	0x00104	W	0x00000000	JPEG luma quantization 2
VEPU_swreg66	0x00108	W	0x00000000	JPEG luma quantization 3
VEPU_swreg67	0x0010c	W	0x00000000	JPEG luma quantization 4
VEPU_swreg68	0x00110	W	0x00000000	JPEG luma quantization 5
VEPU_swreg69	0x00114	W	0x00000000	JPEG luma quantization 6
VEPU_swreg70	0x00118	W	0x00000000	JPEG luma quantization 7
VEPU_swreg71	0x0011c	W	0x00000000	JPEG luma quantization 8
VEPU_swreg72	0x00120	W	0x00000000	JPEG luma quantization 9
VEPU_swreg73	0x00124	W	0x00000000	Register0073 Abstract
VEPU_swreg74	0x00128	W	0x00000000	JPEG luma quantization 11
VEPU_swreg75	0x0012c	W	0x00000000	JPEG luma quantization 12
VEPU_swreg76	0x00130	W	0x00000000	JPEG luma quantization 13
VEPU_swreg77	0x00134	W	0x00000000	JPEG luma quantization 14
VEPU_swreg78	0x00138	W	0x00000000	JPEG luma quantization 15
VEPU_swreg79	0x0013c	W	0x00000000	JPEG luma quantization 16

Name	Offset	Size	Reset Value	Description
VEPU_swreg80	0x00140	W	0x00000000	JPEG chroma quantization 1
VEPU_swreg81	0x00144	W	0x00000000	JPEG chroma quantization 2
VEPU_swreg82	0x00148	W	0x00000000	JPEG chroma quantization 3
VEPU_swreg83	0x0014c	W	0x00000000	JPEG chroma quantization 4
VEPU_swreg84	0x00150	W	0x00000000	JPEG chroma quantization 5
VEPU_swreg85	0x00154	W	0x00000000	JPEG chroma quantization 6
VEPU_swreg86	0x00158	W	0x00000000	JPEG chroma quantization 7
VEPU_swreg87	0x0015c	W	0x00000000	JPEG chroma quantization 8
VEPU_swreg88	0x00160	W	0x00000000	JPEG chroma quantization 9
VEPU_swreg89	0x00164	W	0x00000000	JPEG chroma quantization 10
VEPU_swreg90	0x00168	W	0x00000000	JPEG chroma quantization 11
VEPU_swreg91	0x0016c	W	0x00000000	JPEG chroma quantization 12
VEPU_swreg92	0x00170	W	0x00000000	JPEG chroma quantization 13
VEPU_swreg93	0x00174	W	0x00000000	JPEG chroma quantization 14
VEPU_swreg94	0x00178	W	0x00000000	JPEG chroma quantization 15
VEPU_swreg95	0x0017c	W	0x00000000	JPEG chroma quantization 16
VEPU_swreg96	0x00180	W	0x00000000	DMV 4p/1p penalty values 0-3
VEPU_swreg97	0x00184	W	0x00000000	DMV 4p/1p penalty values 4-7
VEPU_swreg98	0x00188	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg99	0x0018c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg100	0x00190	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg101	0x00194	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg102	0x00198	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg103	0x0019c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg104	0x001a0	W	0x00000000	DMV 4p/1p penalty values

Name	Offset	Size	Reset Value	Description
VEPU_swreg105	0x001a4	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg106	0x001a8	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg107	0x001ac	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg108	0x001b0	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg109	0x001b4	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg110	0x001b8	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg111	0x001bc	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg112	0x001c0	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg113	0x001c4	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg114	0x001c8	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg115	0x001cc	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg116	0x001d0	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg117	0x001d4	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg118	0x001d8	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg119	0x001dc	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg120	0x001e0	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg121	0x001e4	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg122	0x001e8	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg123	0x001ec	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg124	0x001f0	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg125	0x001f4	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg126	0x001f8	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg127	0x001fc	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg128	0x00200	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg129	0x00204	W	0x00000000	DMV 4p/1p penalty values

Name	Offset	Size	Reset Value	Description
VEPU_swreg130	0x00208	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg131	0x0020c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg132	0x00210	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg133	0x00214	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg134	0x00218	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg135	0x0021c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg136	0x00220	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg137	0x00224	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg138	0x00228	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg139	0x0022c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg140	0x00230	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg141	0x00234	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg142	0x00238	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg143	0x0023c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg144	0x00240	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg145	0x00244	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg146	0x00248	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg147	0x0024c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg148	0x00250	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg149	0x00254	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg150	0x00258	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg151	0x0025c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg152	0x00260	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg153	0x00264	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg154	0x00268	W	0x00000000	DMV 4p/1p penalty values

Name	Offset	Size	Reset Value	Description
VEPU_swreg155	0x0026c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg156	0x00270	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg157	0x00274	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg158	0x00278	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg159	0x0027c	W	0x00000000	DMV 4p/1p penalty values
VEPU_swreg160	0x00280	W	0x00000000	vp8 control
VEPU_swreg161	0x00284	W	0x00000000	VP8 bit cost of golden ref frame
VEPU_swreg162	0x00288	W	0x00000000	vp8 loop filter delta registers
VEPU_swreg163	0x0028c	W	0x00000000	vp8 loop filter delta register

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

17.4.8 VEPU Detail Register Description

VEPU_swreg0

Address: Operational Base + offset (0x00000)
product ID

Bit	Attr	Reset Value	Description
31:16	RO	0x4831	prod_id Product ID
15:12	RW	0x0	major_num Major number
11:4	RW	0x00	minor_num Minor number
3:0	RW	0x0	synthesis

VEPU_swreg1

Address: Operational Base + offset (0x00004)
interrupt control and status

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	irq_slice_ready IRQ slice ready status bit
7	RO	0x0	reserved
6	RW	0x0	irq_timeout IRQ HW timeout status bit
5	RW	0x0	irq_buffer_full irq buffer full
4	RW	0x0	irq_reset irq SW reset

Bit	Attr	Reset Value	Description
3	RW	0x0	irq_bus_error
2	RW	0x0	irq_frame_rdy IRQ frame ready status bit. Encoder has finished a frame
1	RW	0x0	irq_dis IRQ disable. No interrupts from HW. SW must use polling
0	RW	0x0	enc_irq HINTenc interrupt from HW. SW resets at IRQ handler.

VEPU_swreg2

Address: Operational Base + offset (0x00008)
 axi control

Bit	Attr	Reset Value	Description
31:24	RW	0x00	axi_write_id axi write id axi write id
23:16	RW	0x00	axi_rd_id axi read id axi read id
15	RW	0x0	output_swap16 enable output swap 16-bits
14	RW	0x0	input_swap16 enable input swap 16-bits
13:8	RW	0x00	burst_len burst length
7	RW	0x0	disable_burst disable burst mode for AXI disable burst mode for AXI bus
6	RW	0x0	burst_incr burst increment burst increment. 1: INCR burst allowed 0: use SINGLE burst
5	RW	0x0	burst_discard enable burst data dicard enable burst data dicard. 2 or 3 long reads are using BURST4
4	RW	0x1	clk_gating_en enable clock gating enable clock gating
3	RW	0x0	output_swap32 enable output swap 32-bits enable output swap 32-bits
2	RW	0x0	input_swap32 enable input swap 32-bits enable input swap 32-bits

Bit	Attr	Reset Value	Description
1	RW	0x0	output_swap8 enable output swap 8-bits enable output swap 8-bits
0	RW	0x0	input_swap8 enable input swap 8-bits enable input swap 8-bits

VEPU_swreg3

Address: Operational Base + offset (0x0000c)

test control register

Bit	Attr	Reset Value	Description
31:28	RW	0x0	test_counter test counter
27:21	RO	0x0	reserved
20:3	RW	0x00000	test_len test data length
2	RW	0x0	test_memory test memory coherency
1	RW	0x0	test_reg test register coherency
0	RW	0x0	test_irq test irq

VEPU_swreg4

Address: Operational Base + offset (0x00010)

reserved

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	Field0000 Abstract Field0000 Description

VEPU_swreg5

Address: Operational Base + offset (0x00014)

addr_output_stream

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_output_stream base address for output stream

VEPU_swreg6

Address: Operational Base + offset (0x00018)

base address for output control

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_output_control base address for output control base address for output control

VEPU_swreg7

Address: Operational Base + offset (0x0001c)
 base address for reference luma

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_reflum base address for reference luma base address for reference luma

VEPU_swreg8

Address: Operational Base + offset (0x00020)
 base address for reference chroma

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_refchroma base address for reference chroma base address for reference chroma

VEPU_swreg9

Address: Operational Base + offset (0x00024)
 base address for reconstructed luma

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_recon_luma base address for reconstructed luma

VEPU_swreg10

Address: Operational Base + offset (0x00028)
 base address for reconstructed chroma

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_recon_chroma base address for reconstructed chroma

VEPU_swreg11

Address: Operational Base + offset (0x0002c)
 base addr for input luma

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_input_luma base addr for input luma

VEPU_swreg12

Address: Operational Base + offset (0x00030)
 base address for input cb

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_input_cb base address for input cb

VEPU_swreg13

Address: Operational Base + offset (0x00034)
 base address for input cr

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr_input_cr base address for input cr

VEPU_swreg14

Address: Operational Base + offset (0x00038)

enc control

Bit	Attr	Reset Value	Description
31	RW	0x0	int_timeout_en enable interrupt for timeout
30	RW	0x0	mv_sad_wren enable writing MV and SAD of each MB to BaseMvWrite
29	RW	0x0	nal_mode NAL size output to base control
28	RW	0x0	slice_rdyint_en enable interrupt for slice ready
27:19	RW	0x000	enc_width encoder width. lumwidth (macroblocks) H264: [9...255] JPEG: [6...511]
18:10	RW	0x000	enc_height encoderd height, lumHeight (macroblocks) H264: [6..255] JPEG: [6..511]
9:7	RO	0x0	reserved
6	RW	0x0	rocon_write_dis disable writing of reconstructed image. recWriteDisable
5	RO	0x0	reserved
4:3	RW	0x0	enc_pic_type encoder picture type. frame type 0: INTER 1: INTRA(IDR) 2: MVC-INTER
2:1	RW	0x0	enc_mode encoding mode. stream type , 1=VP8, 2=JPEG,3=H264
0	RW	0x0	enc_en encoder enable

VEPU_swreg15

Address: Operational Base + offset (0x0003c)

input image control

Bit	Attr	Reset Value	Description
31:29	RW	0x0	input_chroma_offset input chrominance offset (bytes)
28:26	RW	0x0	input_lum_offset input luminance offset(bytes)

Bit	Attr	Reset Value	Description
25:12	RW	0x0000	input_row_len input luminance row length
11:10	RW	0x0	overfill_right overfill pixels on right edge of image div4[0...3]
9:6	RW	0x0	overfill_bot overfill pixels on bottom edge of image. YFill [0..15]
5:2	RW	0x0	input_format input image format. YUV420P/YUV420SP/YUV422/UYVY422/RGB5 65/RGB444/RGB888/RGB101010
1:0	RW	0x0	imagein_rotmode input image rotation 0: disabled 1: 90 degrees right 2: 90 degrees left

VEPU_swreg16

Address: Operational Base + offset (0x00040)

encoder control register 0

Bit	Attr	Reset Value	Description
31:26	RW	0x00	init_qp H.264 pic init qp in PPS[0...51]
25:22	RW	0x0	slice_alpha H.264 slice filter alpha c0 offset div2 [-66]
21:18	RW	0x0	slice_beta h.264 slice filter beta offset div2 [-6 ...6]
17:13	RW	0x00	chroma_qp_offset H264 chroma qp index offset [-12...12]
12:5	RO	0x0	reserved
4:1	RW	0x0	idr_picid IDR pic ID
0	RW	0x0	constr_intra_pred constrained intra prediction

VEPU_swreg17

Address: Operational Base + offset (0x00044)

encoder control register 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	pic_parameter_set_id H.264 pic_parameter_set_id
23:16	RW	0x00	intra_pred_mode H.264 intra prediction previous 4x4 mode favor
15:0	RW	0x0000	frame_num H.264 frame number

VEPU_swreg18

Address: Operational Base + offset (0x00048)

encoder control register 2

Bit	Attr	Reset Value	Description
31:30	RW	0x0	deblocking_filter_mode Deblocking filter mode. 0=enabled. 1=disabled (vp8=simple). 2=disabled on slice
29:23	RW	0x00	h264_slice_size H.264 Slice size. mbRowPerSlice (mb rows) [0..127] 0=one slice per picture
22	RW	0x0	disable_quarter_pixmv H.264 Disable quarter pixel MVs. disableQuarterPixelMv
21	RW	0x0	transform8x8_mode_en H.264 Transform 8x8 enable. High Profile H.264. transform8x8Mode
20:19	RW	0x0	cabac_int_idc H.264 CABAC initial IDC. [0..2]
18	RW	0x0	entropy_coding_mode H.264 CABAC / VP8 boolenc enable. entropyCodingMode. 0=CAVLC (Baseline Profile H.264). 1=CABAC (Main Profile H.264)
17	RW	0x0	h264_inter4x4_mode H.264 Inter 4x4 mode restriction. restricted4x4Mode
16	RW	0x0	h264_stream_mode H.264 Stream mode. 0=NAL unit stream. 1=Byte stream
15:0	RW	0x0000	intra16x16_mode Intra prediction intra 16x16 mode favor

VEPU_swreg19

Address: Operational Base + offset (0x0004c)

encoder control register 3

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30	RW	0x0	mutimv_en Enable using more than 1 MV per macroblock.
29:20	RW	0x000	mv_penalty_1_4p Differential MV penalty for 1/4p ME. DMVPenaltyQp
19:10	RW	0x000	mv_penalty_4p Differential MV penalty for 4p ME. DMVPenalty4p
9:0	RW	0x000	mv_penalty_1p differential MV penalty for 1p

VEPU_swreg20

Address: Operational Base + offset (0x00050)

JPEG control register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:20	RW	0x000	mv_penalty_16x8_8x16 Penalty for using 16x8 or 8x16 MV.
19:10	RW	0x000	mv_penalty_8x8 Penalty for using 8x8 MV
9:0	RW	0x000	mv_penalty_8x4_4x8 Penalty for using 8x4 or 4x8 MV.

VEPU_swreg21

Address: Operational Base + offset (0x00054)

encoder control register 4

Bit	Attr	Reset Value	Description
31:24	RW	0x00	macroblock_penalty H.264 SKIP macroblock mode / VP8 zero/nearest/near mode penalty
23:16	RW	0x00	completed_slices H.264 amount of completed slices.
15:0	RW	0x0000	inter_mode inter MB mode favor in intra/inter selection

VEPU_swreg22

Address: Operational Base + offset (0x00058)

stream header remainder bits MSB

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	strm_hdr_rem stream header remainder bits MSB stream header remainder bits MSB

VEPU_swreg23

Address: Operational Base + offset (0x0005c)

stream header remainder bits LSB

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	strm_hdr_rem2 stream header remainder bits LSB

VEPU_swreg24

Address: Operational Base + offset (0x00060)

stream buffer limit

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	strm_buffer_limit stream buffer limit

VEPU_swreg25

Address: Operational Base + offset (0x00064)

MAD control register

Bit	Attr	Reset Value	Description
31:28	RW	0x0	QP_adjust MAD based QP adjustment. madQpChange [-8..7]
27:22	RW	0x00	MAD_thredhold MAD threshold div256
21	RO	0x0	reserved
20:0	RW	0x000000	qp_sum_div2 QP sum div2 output

VEPU_swreg27

Address: Operational Base + offset (0x0006c)

QP register

Bit	Attr	Reset Value	Description
31:26	RW	0x00	qp_lum H.264 Initial QP. qpLum [0..51]
25:20	RW	0x00	qp_max H.264 Minimum QP. qpMax [0..51]
19:14	RW	0x00	qp_min H.264 Minimum QP. qpMin [0..51]

Bit	Attr	Reset Value	Description
13	RO	0x0	reserved
12:0	RW	0x0000	checkpoint_distan checkpoint distance checkpoint distance

VEPU_swreg28

Address: Operational Base + offset (0x00070)

checkpoint 1 and 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	checkp_1 checkpoint 1 word target/usage
15:0	RW	0x0001	checkp_2 checkpoint 2 word target/usage

VEPU_swreg29

Address: Operational Base + offset (0x00074)

checkpoint 3 and 4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	checkp_3 checkpoint 3 word target/usage
15:0	RW	0x0000	checkp_4 checkpoint 4 word target/usage checkpoint 4 word target/usage

VEPU_swreg30

Address: Operational Base + offset (0x00078)

checkpoint 5 and 6

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	checkp_5 checkpoint 5 word target/usage
15:0	RW	0x0000	checkp_6 checkpoint 6 word target/usage checkpoint 6 word target/usage

VEPU_swreg31

Address: Operational Base + offset (0x0007c)

checkpoint 7 and 8

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	checkp_7 checkpoint 7 word target/usage
15:0	RW	0x0000	checkp_8 checkpoint 8 word target/usage checkpoint 8 word target/usage

VEPU_swreg32

Address: Operational Base + offset (0x00080)

checkpoint 9 and 10

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	checkp_9 checkpoint 9 word target/usage
15:0	RW	0x0000	checkp_10 checkpoint 10 word target/usage checkpoint 10 word target/usage

VEPU_swreg33

Address: Operational Base + offset (0x00084)
 checkpoint word error 1 and 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	checkp_error1 checkpoint word error 1
15:0	RW	0x0000	checkp_error2 checkpoint word error 2

VEPU_swreg34

Address: Operational Base + offset (0x00088)
 checkpoint word error 1 and 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	checkp_error3 checkpoint word error 3
15:0	RW	0x0000	checkp_error4 checkpoint word error 4

VEPU_swreg35

Address: Operational Base + offset (0x0008c)
 checkpoint word error 1 and 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	checkp_error5 checkpoint word error 5
15:0	RW	0x0000	checkp_error6 checkpoint word error 6

VEPU_swreg36

Address: Operational Base + offset (0x00090)
 checkpoint delta QP register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:24	RW	0x0	checkp_qp_1 checkpoint delta QP 1
23:20	RW	0x0	checkp_qp_2 checkpoint delta QP 2
19:16	RW	0x0	checkp_qp_3 checkpoint delta QP 3
15:12	RW	0x0	checkp_qp_4 checkpoint delta QP 4
11:8	RW	0x0	checkp_qp_5 checkpoint delta QP 5

Bit	Attr	Reset Value	Description
7:4	RW	0x0	checkp_qp_6 checkpoint delta QP 6
3:0	RW	0x0	checkp_qp_7 checkpoint delta QP 7

VEPU_swreg37

Address: Operational Base + offset (0x00094)

rlc control

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:23	RW	0x00	stream_st_offset Field0000 Description
22	RO	0x0	reserved
21:0	RW	0x0000000	rlc_sum rlc sum rlc sum

VEPU_swreg38

Address: Operational Base + offset (0x00098)

mb control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	macroblock_count Field0000 Description
15:0	RW	0x0000	mb_count_out Field0000 Description

VEPU_swreg39

Address: Operational Base + offset (0x0009c)

Base address for next pic

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	next_pic_addr Base address for next pic Base address for next pic luminance

VEPU_swreg40

Address: Operational Base + offset (0x000a0)

Stabilization minimum value

Bit	Attr	Reset Value	Description
31:30	RW	0x0	stab_mode Stabilization mode. 0=disabled. 1=stab only. 2=stab+encode
29:24	RO	0x0	reserved

Bit	Attr	Reset Value	Description
23:0	RW	0x000000	stab_min_value Stabilization minimum value output. max 253*253*255

VEPU_swreg41

Address: Operational Base + offset (0x000a4)

Stabilization motion sum

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	stab_motion_sum Stabilization motion sum div8 output. max 53*253*255*1089/8

VEPU_swreg42

Address: Operational Base + offset (0x000a8)

stab_matrix1 and stab_gmv_hor

Bit	Attr	Reset Value	Description
31:26	RW	0x00	stab_gmv_hor Stabilization GMV horizontal output [-16..16]
25:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix1 Stabilization matrix 1 (up-left position) output

VEPU_swreg43

Address: Operational Base + offset (0x000ac)

stab_matrix2 and stab_gmv_ver

Bit	Attr	Reset Value	Description
31:26	RW	0x00	stab_gmv_ver Stabilization GMV vertical output [-16..16]
25:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix2 Stabilization matrix 2 (up position) output

VEPU_swreg44

Address: Operational Base + offset (0x000b0)

stab_matrix3

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved

Bit	Attr	Reset Value	Description
23:0	RW	0x000000	stab_matrix3 Stabilization matrix 3 (up-right position) output

VEPU_swreg45

Address: Operational Base + offset (0x000b4)

stab_matrix4

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix4 Stabilization matrix 4 (left position) output

VEPU_swreg46

Address: Operational Base + offset (0x000b8)

stab_matrix5

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix5 Stabilization matrix 5 (GMV position) output

VEPU_swreg47

Address: Operational Base + offset (0x000bc)

stab_matrix6

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix6 Stabilization matrix 6 (right position) output

VEPU_swreg48

Address: Operational Base + offset (0x000c0)

stab_matrix7

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix7 Stabilization matrix 7 (down-left position) output

VEPU_swreg49

Address: Operational Base + offset (0x000c4)

stab_matrix8

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix8 Stabilization matrix 8 (down position) output

VEPU_swreg50

Address: Operational Base + offset (0x000c8)

stab_matrix9

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix9 Stabilization matrix 9 (down-right position) output

VEPU_swreg51

Address: Operational Base + offset (0x000cc)

cabac_table_addr

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cabac_table_addr Base address for cabac context tables (H264) or probability tables (VP8)

VEPU_swreg52

Address: Operational Base + offset (0x000d0)

Base address for MV output

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mv_out_addr Base address for MV output writing

VEPU_swreg53

Address: Operational Base + offset (0x000d4)

RGB to YUV conversion coefficient A and B

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	coeffB_RGB2YUV RGB to YUV conversion B RGB to YUV conversion coefficient B
15:0	RW	0x0000	coeffA_RGB2YUV RGB to YUV conversion coefficient A

VEPU_swreg54

Address: Operational Base + offset (0x000d8)

RGB to YUV conversion coefficien C and E

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	coeffE_RGB2YUV RGB to YUV conversion coefficient E
15:0	RW	0x0000	coeffC_RGB2YUV RGB to YUV conversion coefficient C

VEPU_swreg55

Address: Operational Base + offset (0x000dc)

RGB mask MSB bit

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:26	RW	0x00	Bmask_MST RGB B-component mask MSB bit position [0..31]
25:21	RW	0x00	Gmask_MST RGB G-component mask MSB bit position [0..31]
20:16	RW	0x00	Rmask_MST RGB R-component mask MSB bit position [0..31]
15:0	RW	0x0000	coeffF_RGB2YUV RGB to YUV conversion coefficien F

VEPU_swreg56

Address: Operational Base + offset (0x000e0)

intra area control register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	intra_left_mb Intra area left mb column (inside area) [0..255]
23:16	RW	0x00	intra_right_mb Intra area right mb column (inside area) [0..255]
15:8	RW	0x00	intra_top_mb Intra area top mb row (inside area) [0..255]
7:0	RW	0x00	intra_bot_mb Intra area bottom mb row (inside area) [0..255]

VEPU_swreg57

Address: Operational Base + offset (0x000e4)

CIR intra control reg

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CIR_first_intra CIR first intra mb. 0=disabled [0..65535]
15:0	RW	0x0000	CIR_intra_mbinterval CIR intra mb interval. 0=disabled [0..65535]

VEPU_swreg58

Address: Operational Base + offset (0x000e8)

Intra slice bitmap for slices 0..31

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	intra_slice_bitmap Intra slice bitmap for slices 0..31. LSB=slice0. MSB=slice31. 1=intra.

VEPU_swreg59

Address: Operational Base + offset (0x000ec)

Intra slice bitmap for slices32..63

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	intra_slice_bitmap Intra slice bitmap for slices32..63. LSB=slice32. MSB=slice63. 1=intra.

VEPU_swreg60

Address: Operational Base + offset (0x000f0)

1st ROI area register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ROI_1st_leftmb 1st ROI area left mb column (inside area) qp+=Roi1DeltaQp
23:16	RW	0x00	ROI_1st_rightmb 1st ROI area right mb column (outside area) qp-=Roi1DeltaQp
15:8	RW	0x00	ROI_1st_topmb 1st ROI area top mb row (inside area)
7:0	RW	0x00	ROI_1st_botmb 1st ROI area bottom mb row (outside area)

VEPU_swreg61

Address: Operational Base + offset (0x000f4)

Register0061 Abstract

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ROI_2st_leftmb 2st ROI area left mb column (inside area) qp+=Roi1DeltaQp
23:16	RW	0x00	ROI_2st_topmb 2st ROI area top mb row (inside area)
15:8	RW	0x00	ROI_2st_rightmb 2st ROI area right mb column (outside area) qp-=Roi1DeltaQp
7:0	RW	0x00	ROI_2st_botmb 2st ROI area bottom mb row (outside area)

VEPU_swreg62

Address: Operational Base + offset (0x000f8)

MVC control reg

Bit	Attr	Reset Value	Description
31:28	RW	0x0	mv16x16_favor Zero 16x16 MV favor div2.
27:19	RW	0x000	penalty_4x4mv Penalty for using 4x4 MV.
18:16	RW	0x0	mvc_priority_id MVC priority_id [0..7]
15:13	RW	0x0	mvc_view_id MVC view_id [0..7]
12:10	RW	0x0	mvc_temporal_id MVC temporal_id [0..7]
9	RW	0x0	mvc_anchor_pic_flag MVC anchor_pic_flag. Specifies that the picture is part of an anchor access unit.
8	RW	0x0	mvc_inter_view_flag MVC inter_view_flag. Specifies that the picture is used for inter- view prediction.
7:0	RO	0x0	reserved

VEPU_swreg63

Address: Operational Base + offset (0x000fc)

Register0063 Abstract

Bit	Attr	Reset Value	Description
31:12	RO	0x1f522	HW_CONFIG Field0000 Description
11:0	RO	0x780	MAX_VID_WIDTH Field0000 Description

VEPU_swreg64

Address: Operational Base + offset (0x00100)

JPEG luma quantization 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua1 JPEG luma quantization 1

VEPU_swreg65

Address: Operational Base + offset (0x00104)

JPEG luma quantization 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua2 JPEG luma quantization 2

VEPU_swreg66

Address: Operational Base + offset (0x00108)

JPEG luma quantization 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua3 JPEG luma quantization 3

VEPU_swreg67

Address: Operational Base + offset (0x0010c)

JPEG luma quantization 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua4 JPEG luma quantization 4

VEPU_swreg68

Address: Operational Base + offset (0x00110)

JPEG luma quantization 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua5 JPEG luma quantization 5

VEPU_swreg69

Address: Operational Base + offset (0x00114)

JPEG luma quantization 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua6 JPEG luma quantization 6 JPEG luma quantization 6

VEPU_swreg70

Address: Operational Base + offset (0x00118)

JPEG luma quantization 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua7 JPEG luma quantization 7

VEPU_swreg71

Address: Operational Base + offset (0x0011c)

JPEG luma quantization 8

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua8 JPEG luma quantization 8

VEPU_swreg72

Address: Operational Base + offset (0x00120)

JPEG luma quantization 9

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua9 JPEG luma quantization 9

VEPU_swreg73

Address: Operational Base + offset (0x00124)

Register0073 Abstract

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua10 JPEG luma quantization 10 JPEG luma quantization 10

VEPU_swreg74

Address: Operational Base + offset (0x00128)

JPEG luma quantization 11

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua11 JPEG luma quantization 11

VEPU_swreg75

Address: Operational Base + offset (0x0012c)

JPEG luma quantization 12

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	jpeg_lum_qua12 JPEG luma quantization 12

VEPU_swreg76

Address: Operational Base + offset (0x00130)

JPEG luma quantization 13

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua13 JPEG luma quantization 13

VEPU_swreg77

Address: Operational Base + offset (0x00134)

JPEG luma quantization 14

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua14 JPEG luma quantization 14

VEPU_swreg78

Address: Operational Base + offset (0x00138)

JPEG luma quantization 15

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua15 JPEG luma quantization 15

VEPU_swreg79

Address: Operational Base + offset (0x0013c)

JPEG luma quantization 16

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_lum_qua16 JPEG luma quantization 16

VEPU_swreg80

Address: Operational Base + offset (0x00140)

JPEG chroma quantization 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua1 JPEG chroma quantization 1

VEPU_swreg81

Address: Operational Base + offset (0x00144)

JPEG chroma quantization 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua2 JPEG chroma quantization 2

VEPU_swreg82

Address: Operational Base + offset (0x00148)

JPEG chroma quantization 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua3 JPEG chroma quantization 3

VEPU_swreg83

Address: Operational Base + offset (0x0014c)

JPEG chroma quantization 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua4 JPEG chroma quantization 4

VEPU_swreg84

Address: Operational Base + offset (0x00150)

JPEG chroma quantization 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua5 JPEG chroma quantization 5

VEPU_swreg85

Address: Operational Base + offset (0x00154)

JPEG chroma quantization 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua6 JPEG chroma quantization 6

VEPU_swreg86

Address: Operational Base + offset (0x00158)

JPEG chroma quantization 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua7 JPEG chroma quantization 7

VEPU_swreg87

Address: Operational Base + offset (0x0015c)

JPEG chroma quantization 8

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua8 JPEG chroma quantization 8

VEPU_swreg88

Address: Operational Base + offset (0x00160)

JPEG chroma quantization 9

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua9 JPEG chroma quantization 9

VEPU_swreg89

Address: Operational Base + offset (0x00164)

JPEG chroma quantization 10

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua10 JPEG chroma quantization 10

VEPU_swreg90

Address: Operational Base + offset (0x00168)

JPEG chroma quantization 11

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua11 JPEG chroma quantization 11

VEPU_swreg91

Address: Operational Base + offset (0x0016c)

JPEG chroma quantization 12

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua12 JPEG chroma quantization 12

VEPU_swreg92

Address: Operational Base + offset (0x00170)

JPEG chroma quantization 13

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua13 JPEG chroma quantization 13

VEPU_swreg93

Address: Operational Base + offset (0x00174)

JPEG chroma quantization 14

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua14 JPEG chroma quantization 14

VEPU_swreg94

Address: Operational Base + offset (0x00178)

JPEG chroma quantization 15

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua15 JPEG chroma quantization 15

VEPU_swreg95

Address: Operational Base + offset (0x0017c)

JPEG chroma quantization 16

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	jpeg_chrom_qua16 JPEG chroma quantization 16

VEPU_swreg96

Address: Operational Base + offset (0x00180)

DMV 4p/1p penalty values 0-3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values 0-3

VEPU_swreg97

Address: Operational Base + offset (0x00184)

DMV 4p/1p penalty values 4-7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values 4-7

VEPU_swreg98

Address: Operational Base + offset (0x00188)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg99

Address: Operational Base + offset (0x0018c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg100

Address: Operational Base + offset (0x00190)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg101

Address: Operational Base + offset (0x00194)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg102

Address: Operational Base + offset (0x00198)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg103

Address: Operational Base + offset (0x0019c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg104

Address: Operational Base + offset (0x001a0)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg105

Address: Operational Base + offset (0x001a4)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg106

Address: Operational Base + offset (0x001a8)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg107

Address: Operational Base + offset (0x001ac)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg108

Address: Operational Base + offset (0x001b0)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg109

Address: Operational Base + offset (0x001b4)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg110

Address: Operational Base + offset (0x001b8)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg111

Address: Operational Base + offset (0x001bc)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg112

Address: Operational Base + offset (0x001c0)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg113

Address: Operational Base + offset (0x001c4)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg114

Address: Operational Base + offset (0x001c8)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg115

Address: Operational Base + offset (0x001cc)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg116

Address: Operational Base + offset (0x001d0)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg117

Address: Operational Base + offset (0x001d4)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg118

Address: Operational Base + offset (0x001d8)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg119

Address: Operational Base + offset (0x001dc)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg120

Address: Operational Base + offset (0x001e0)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg121

Address: Operational Base + offset (0x001e4)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg122

Address: Operational Base + offset (0x001e8)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg123

Address: Operational Base + offset (0x001ec)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg124

Address: Operational Base + offset (0x001f0)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg125

Address: Operational Base + offset (0x001f4)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg126

Address: Operational Base + offset (0x001f8)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg127

Address: Operational Base + offset (0x001fc)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg128

Address: Operational Base + offset (0x00200)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg129

Address: Operational Base + offset (0x00204)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg130

Address: Operational Base + offset (0x00208)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg131

Address: Operational Base + offset (0x0020c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg132

Address: Operational Base + offset (0x00210)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg133

Address: Operational Base + offset (0x00214)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg134

Address: Operational Base + offset (0x00218)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg135

Address: Operational Base + offset (0x0021c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg136

Address: Operational Base + offset (0x00220)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg137

Address: Operational Base + offset (0x00224)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg138

Address: Operational Base + offset (0x00228)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg139

Address: Operational Base + offset (0x0022c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg140

Address: Operational Base + offset (0x00230)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg141

Address: Operational Base + offset (0x00234)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg142

Address: Operational Base + offset (0x00238)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg143

Address: Operational Base + offset (0x0023c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg144

Address: Operational Base + offset (0x00240)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg145

Address: Operational Base + offset (0x00244)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg146

Address: Operational Base + offset (0x00248)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg147

Address: Operational Base + offset (0x0024c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg148

Address: Operational Base + offset (0x00250)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg149

Address: Operational Base + offset (0x00254)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg150

Address: Operational Base + offset (0x00258)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg151

Address: Operational Base + offset (0x0025c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg152

Address: Operational Base + offset (0x00260)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg153

Address: Operational Base + offset (0x00264)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg154

Address: Operational Base + offset (0x00268)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg155

Address: Operational Base + offset (0x0026c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg156

Address: Operational Base + offset (0x00270)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg157

Address: Operational Base + offset (0x00274)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg158

Address: Operational Base + offset (0x00278)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values

VEPU_swreg159

Address: Operational Base + offset (0x0027c)

DMV 4p/1p penalty values

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMV_4p_1p_penalty DMV 4p/1p penalty values 124-127

VEPU_swreg160

Address: Operational Base + offset (0x00280)

vp8 control

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:12	RW	0x000	vp8_coeff_dmv_penalty VP8 coeff for dmv penalty for intra/inter selection
11:0	RW	0x000	vp8_inter_type VP8 bit cost of inter type

VEPU_swreg161

Address: Operational Base + offset (0x00284)

VP8 bit cost of golden ref frame

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	vp8_ref_frame VP8 bit cost of golden ref frame

VEPU_swreg162

Address: Operational Base + offset (0x00288)

vp8 loop filter delta registers

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:21	RW	0x00	vp8_loopfilter_altref VP8 loop filter delta for alt ref
20:14	RW	0x00	vp8_loopfilter_goldenref VP8 loop filter delta for golden ref
13:7	RW	0x00	vp8_loopfilter_lastref VP8 loop filter delta for last ref
6:0	RW	0x00	vp8_loopfilter_intra VP8 loop filter delta for intra mb

VEPU_swreg163

Address: Operational Base + offset (0x0028c)

vp8 loop filter delta register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:21	RW	0x00	vp8_loopfilter_splitmv VP8 loop filter delta for SPLITMV
20:14	RW	0x00	vp8_loopfilter_newmv VP8 loop filter delta for NEWMV

Bit	Attr	Reset Value	Description
13:7	RW	0x00	vp8_loopfilter_zeromv VP8 loop filter delta for ZEROMV
6:0	RW	0x00	vp8_loopfilter_bpred VP8 loop filter delta for BPRED

17.4.9 VDPU Register Summary

Name	Offset	Size	Reset Value	Description
VDPU_SWREG0	0x00000	W	0x67313688	ID register(read only)
VDPU_SWREG1	0x00004	W	0x00000000	interrupt register decoder
VDPU_SWREG2	0x00008	W	0x00000400	device configuration register decoder
VDPU_SWREG3	0x0000c	W	0x00000000	Device control register 0(deemode, picture type etc)
VDPU_SWREG4	0x00010	W	0x00000000	decoder control register 1(picture parameters)
VDPU_SWREG5	0x00014	W	0x00000000	decoder control register2 (stream decoding table selects)
VDPU_SWREG6	0x00018	W	0x00000000	decoder control register 3(stream buffer information)
VDPU_SWREG7	0x0001c	W	0x00000000	decoder control register 4(H264, VC-1,VP6 control)
VDPU_SWREG8	0x00020	W	0x00000000	decoder control register 5(H264, VC-1,VP6 and RV control)
VDPU_SWREG9	0x00024	W	0x00000000	decoder control register 6
VDPU_SREG10	0x00028	W	0x00000000	Base address for differential motion vector base address
VDPU_SWREG11	0x0002c	W	0x00000000	decoder control register 7
VDPU_SWREG12	0x00030	W	0x00000000	Base address for RLC data (RLC) / stream start address/decoded
VDPU_SWREG13	0x00034	W	0x00000000	Base address for decoded picture / base address for JPEG deco
VDPU_SWREG14	0x00038	W	0x00000000	Base address for reference picture index 0 / base address for J
VDPU_SWREG15	0x0003c	W	0x00000000	Base address for reference picture index 1 / JPEG control

Name	Offset	Size	Reset Value	Description
VDPU_SWREG15_JPEG_ROI	0x0003c	W	0x00000000	JPEG roi control
VDPU_SWREG16	0x00040	W	0x00000000	base address for reference picture index 2 / List of VLC code len
VDPU_SWREG17	0x00044	W	0x00000000	Base address for reference picture index 3 / List of VLC code len
VDPU_SWREG18	0x00048	W	0x00000000	Base address for reference picture index 4 / VC1 control / MPE
VDPU_SWREG19	0x0004c	W	0x00000000	Base address for reference picture index 5
VDPU_SWREG20	0x00050	W	0x00000000	Base address for reference picture index 6
VDPU_SWREG21	0x00054	W	0x00000000	Base address for reference picture index 7
VDPU_SWREG22	0x00058	W	0x00000000	Base address for reference picture index 8
VDPU_SWREG23	0x0005c	W	0x00000000	Base address for reference picture index 9
VDPU_SWREG24	0x00060	W	0x00000000	Base address for reference picture index 10
VDPU_SWREG25	0x00064	W	0x00000000	Base address for reference picture index 11
VDPU_SWREG26	0x00068	W	0x00000000	Base address for reference picture index 12
VDPU_SWREG27	0x0006c	W	0x00000000	Base address for reference picture index 13
VDPU_SWREG28	0x00070	W	0x00000000	Base address for reference picture index 14
VDPU_SWREG29	0x00074	W	0x00000000	Base address for reference picture index 15
VDPU_SWREG30	0x00078	W	0x00000000	Reference picture numbers for index 0 and 1 (H264 VLC)
VDPU_SWREG31	0x0007c	W	0x00000000	Reference picture numbers for index 2 and 3 (H264 VLC) /
VDPU_SWREG32	0x00080	W	0x00000000	Reference picture numbers for index 4 and 5 (H264 VLC)
VDPU_SWREG33	0x00084	W	0x00000000	Reference picture numbers for index 6 and 7 (H264 VLC)

Name	Offset	Size	Reset Value	Description
VDPU_SWREG34	0x00088	W	0x00000000	Reference picture numbers for index 8 and 9 (H264 VLC)
VDPU_SWREG35	0x0008c	W	0x00000000	Reference picture numbers for index 10 and 11 (H264 VLC)
VDPU_SWREG35_JPEG_ROI	0x0008c	W	0x00000000	JPEG roi offset/dc base address
VDPU_SWREG36	0x00090	W	0x00000000	Reference picture numbers for index 12 and 13 (H264 VLC)
VDPU_SWREG36_JPEG_ROI	0x00090	W	0x00000000	JPEG roi offset/dc length
VDPU_SWREG37	0x00094	W	0x00000000	Reference picture numbers for index 14 and 15 (H264 VLC)
VDPU_SWREG38	0x00098	W	0x00000000	Reference picture long term flags (H264 VLC) / VPx prediction filt
VDPU_SWREG39	0x0009c	W	0x00000000	Reference picture valid flags (H264 VLC) /VPx prediction filter ta
VDPU_SWREG40	0x000a0	W	0x00000000	Base address for standard dependent tables
VDPU_SWREG41	0x000a4	W	0x00000000	Base address for direct mode motion vectors
VDPU_SWREG42	0x000a8	W	0x00000000	bi_dir initial ref pic list register (0-2)/ VP6 prediction filter taps
VDPU_SWREG43	0x000ac	W	0x00000000	bi-dir initial ref pic list register (3-5)/ VP6 prediction filter taps
VDPU_SWREG44	0x000b0	W	0x00000000	bi-dir initial ref pic list register (6-8)/ VP6 prediction filter taps
VDPU_SWREG45	0x000b4	W	0x00000000	bi-dir initial ref pic list register (9- 11) / VP6 prediction filter taps
VDPU_SWREG46	0x000b8	W	0x00000000	bi-dir initial ref pic list register (12- 14) / VP7,VP8 quantization v
VDPU_SWREG47	0x000bc	W	0x00000000	bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,V
VDPU_SWREG48	0x000c0	W	0x00000000	Error concealment register
VDPU_SWREG49	0x000c4	W	0x00000000	Prediction filter tap register for H264, MPEG4, VC1, VP6

Name	Offset	Size	Reset Value	Description
VDPU_SWREG50	0x000c8	W	0xfb56f80	Synthesis configuration register decoder 0 (read only)
VDPU_SWREG51	0x000cc	W	0x00000000	Reference picture buffer control register
VDPU_SWREG52	0x000d0	W	0x00000000	Reference picture buffer information register 1 (read only)
VDPU_SWREG53	0x000d4	W	0x00000000	Reference picture buffer information register 2 (read only)
VDPU_SWREG54	0x000d8	W	0xe5da0000	Synthesis configuration register decoder 1 (read only)
VDPU_SWREG55	0x000dc	W	0x00000000	Reference picture buffer 2 / Advanced prefetch control register
VDPU_SWREG56	0x000e0	W	0x00000000	Reference buffer information register 3 (read only)
VDPU_SWREG57_INTRA_INTER	0x000e4	W	0x00000000	intra_dll3t,intra_dblspeed,inter_dblspeed,stream_len_hi
VDPU_SWREG58	0x000e8	W	0x00000000	Decoder debug register 0 (read only)
VDPU_SWREG59	0x000ec	W	0x00000000	H264 Chrominance 8 pixel interleaved data base
VDPU_SWREG60	0x000f0	W	0x00000000	Interrupt register post-processor
VDPU_SWREG61	0x000f4	W	0x00000000	Device configuration register post-processor
VDPU_SWREG62	0x000f8	W	0x00000000	Deinterlace control register
VDPU_SWREG63	0x000fc	W	0x00000000	base address for reading post-processing input picture uminan
VDPU_SWREG64	0x00100	W	0x00000000	Base address for reading post-processing input picture Cb/Ch
VDPU_SWREG65	0x00104	W	0x00000000	Base address for reading post-processing input picture Cr
VDPU_SWREG66	0x00108	W	0x00000000	Base address for writing post-processed picture luminance/RGB
VDPU_SWREG67	0x0010c	W	0x00000000	Base address for writing post-processed picture Ch
VDPU_SWREG68	0x00110	W	0x00000000	Register for contrast adjusting

Name	Offset	Size	Reset Value	Description
VDPU_SWREG69	0x00114	W	0x00000000	Register for colour conversion and contrast adjusting
VDPU_SWREG70	0x00118	W	0x00000000	Register for colour conversion 0
VDPU_SWREG71	0x0011c	W	0x00000000	Register for colour conversion 1 + rotation mode
VDPU_SWREG72	0x00120	W	0x00000000	PP input size and -cropping register
VDPU_SWREG73	0x00124	W	0x00000000	PP input picture base address for Y bottom field
VDPU_SWREG74	0x00128	W	0x00000000	PP input picture base for Ch bottom field
VDPU_SWREG79	0x0013c	W	0x00000000	Scaling ratio register 1 & padding for B
VDPU_SWREG80	0x00140	W	0x00000000	Scaling register 0 ratio & padding for R and G
VDPU_SWREG81	0x00144	W	0x00000000	Scaling ratio register 2
VDPU_SWREG82	0x00148	W	0x00000000	Rmask register
VDPU_SWREG83	0x0014c	W	0x00000000	Gmask register
VDPU_SWREG84	0x00150	W	0x00000000	Bmask register
VDPU_SWREG85	0x00154	W	0x00000000	Post-processor control register
VDPU_SWREG86	0x00158	W	0x00000000	Mask 1 start coordinate register
VDPU_SWREG87	0x0015c	W	0x00000000	Mask 2 start coordinate register
VDPU_SWREG88	0x00160	W	0x00000000	Mask 1 size and PP original width register
VDPU_SWREG89	0x00164	W	0x00000000	Mask 2 size register
VDPU_SWREG90	0x00168	W	0x00000000	PiP register 0
VDPU_SWREG91	0x0016c	W	0x00000000	PiP register 1 and dithering control
VDPU_SWREG92	0x00170	W	0x00000000	Display width and PP input size extension register
VDPU_SWREG93	0x00174	W	0x00000000	Display width and PP input size extension register
VDPU_SWREG94	0x00178	W	0x00000000	Base address for alpha blend 2 gui component
VDPU_SWREG95	0x0017c	W	0x00000000	Base address for alpha blend 2 gui component
VDPU_SWREG98	0x00188	W	0x00000000	PP output width/height extension
VDPU_SWREG99	0x0018c	W	0xe000f000	PP fuse register (read only)
VDPU_SWREG100	0x00190	W	0xff874000	Synthesis configuration register post-processor (read only)

Name	Offset	Size	Reset Value	Description
VDPU_SWREG101	0x00194	W	0x00000000	soft reset signals

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

17.4.10 VDPU Detail Register Description

VDPU_SWREG0

Address: Operational Base + offset (0x00000)

ID register(read only)

Bit	Attr	Reset Value	Description
31:16	RO	0x6731	pro_num product number
15:12	RO	0x3	major_version
11:4	RO	0x68	minor_version
3	RO	0x1	ID_ASCII_EN ASCII type product ID enable
2:0	RO	0x0	build_version

VDPU_SWREG1

Address: Operational Base + offset (0x00004)

interrupt register decoder

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	sw_dec_pic_inf B slice detected. This signal is driven high during picture ready interrupt if B-type slice is found. This bit does not launch interrupt but is used to inform SW about h264 tools. DIVX3: For DIVX3 this bit tells the value of extension header flag (flag called FLIPFLOP)
23:19	RO	0x0	reserved
18	RW	0x0	sw_dec_timeout Interrupt status bit decoder timeout. When high the decoder 0 has been idling for too long. HW will self reset. Possible only if timeout interrupt is enabled
17	RW	0x0	sw_dec_slice_int Interrupt status bit dec_slice_decoded. When high SW must set new base addresses for sw_dec_out_base and sw_jpg_ch_out_base before resetting this status bit. Used for JPEG and VP8 snapshot modes

Bit	Attr	Reset Value	Description
16	RW	0x0	sw_dec_error_int Interrupt status bit input stream error. When high, an error is found in input data stream decoding. HW will self reset.
15	RW	0x0	sw_dec_aso_int Interrupt status bit ASO (Arbitrary Slice Ordering) detected. When high, ASO detected in input data stream decoding. HW will self reset.
14	RW	0x0	sw_dec_buffer_int Interrupt status bit input buffer empty. When high, input stream buffer is empty but picture is not ready. HW will not self reset.
13	RW	0x0	sw_dec_bus_int Interrupt status bit bus. Error response from bus. HW will self reset
12	RW	0x0	sw_dec_rdy_int Interrupt status bit decoder. When this bit is high decoder has decoded a picture. HW will self reset.
11:9	RO	0x0	reserved
8	RW	0x0	sw_dec_irq Decoder IRQ. When high, decoder requests an interrupt. SW will reset this after interrupt is handled.
7:5	RO	0x0	reserved
4	RW	0x0	sw_dec_irq_dis Decoder IRQ disable. When high, there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt statuses.
3:1	RO	0x0	reserved
0	RW	0x0	sw_dec_en decoder enable. Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when picture is processed or ASO or stream error is detected or bus error or timeout interrupt is given.

VDPU_SWREG2

Address: Operational Base + offset (0x00008)
 device configuration register decoder

Bit	Attr	Reset Value	Description
31:24	RW	0x00	sw_dec_axi_rd_id Read ID used for decoder reading services in AXI bus (if connected to AXI)

Bit	Attr	Reset Value	Description
23	RW	0x0	sw_dec_timeout_e Timeout interrupt enable. If enabled HW may return timeout interrupt in case HW gets stucked while decoding picture.
22	RW	0x0	sw_dec_strswap32_e Decoder input 32bit data swap for stream data (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
21	RW	0x0	sw_dec_strendian_e Decoder input endian mode for stream data: 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)
20	RW	0x0	sw_dec_inswap32_e Decoder input 32bit data swap for other than stream data (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
19	RW	0x0	sw_dec_outswap32_e Decoder output 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
18	RW	0x0	sw_dec_data_disc_e Data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally.
17	RW	0x0	sw_tiled_mode_msb Tiled mode msb. Concatenated to Tiled mode lsb which form 2 bit tiled mode. Definition of tiledmode: 0 = Tiled mode not enabled 1 = Tiled mode enabled for 8x4 tile size 2,3 Reserved

Bit	Attr	Reset Value	Description
16:11	RW	0x00	<p>sw_dec_latency Decoder master interface additional latency. Can be used to slow down decoder HW between services in steps of 8 clock cycles: 0 = no latency 1 = minimum 8 cycles of IDLE between services 2 = minimum 16 cycles of IDLE between services ... 63 = minimum latency of 504 cycles of IDLE between services</p>
10	RW	0x1	<p>sw_dec_clk_gate_e Decoder dynamic clock gating enable: 0 = Clock is running for all structures 1 = Clock is gated for decoder structures that are not used Note: Clock gating value can be changed only when decoder is disabled</p>
9	RW	0x0	<p>sw_dec_in_endian Decoder input endian mode for other than stream data: 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)</p>
8	RW	0x0	<p>sw_dec_out_endian Decoder output endian mode: 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)</p>
7	RW	0x0	<p>sw_tiled_mode_lsb Tiled mode lsb. Concatenated to Tiled mode msb which form 2 bit tiled mode. Defined in tiled_mode_msb</p>
6	RW	0x0	<p>sw_dec_adv_pre_dis Advanced PREFETCH mode disable (advanced reference picture reading mode for video)</p>
5	RW	0x0	<p>sw_dec_scmd_dis AXI Single Command Multiple Data 0 disable. (where only the first addresses of the burst are given from address generator). This bit is used to disable the feature (possible SW workaround if something is not working correctly)</p>
4:0	RW	0x00	<p>sw_dec_max_burst Maximum burst length for decoder bus transactions. Valid values: AXI: 1-16</p>

VDPU_SWREG3

Address: Operational Base + offset (0x0000c)

Device control register 0(decmode, picture type etc)

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:28	RW	0x0	<p>sw_dec_mode Decoding mode: 0 = H.264, 1 = MPEG-4, 2 = H.263, 3 = JPEG, 4 = VC-1, 5 = MPEG-2, 6 = MPEG-1, 7 = VP6, 8 = RV, 9 = VP7, 10 = VP8, 11 = AVS, others = reserved</p>
27	RW	0x0	<p>sw_rlc_mode_e RLC mode enable: 1 = HW decodes video from RLC input data + side information (Differential MV's, separate DC coeffs, Intra 4x4 modes, MB control). Valid only for H.264 Baseline and MPEG-4 SP. 0 = HW decodes video from bit stream (VLC mode) + side information (bitplane data in VC-1)</p>
26	RW	0x0	<p>sw_skip_mode MPEG4: Valid if divx is enabled AVS: 0: special MB type code indicates skipped mbs 1 means that skipped mbs are indicated using skip_run -syntax element like in H264 VP8: 0 : HW decodes mb_coeff_skip -flag 1 : HW does not decode mb_coeff_skip -flag</p>
25	RW	0x0	<p>sw_divx3_e DIVX3 enable (possible if sw_dec_mode is MPEG4): 0 = disabled 1 = enabled</p>
24	RW	0x0	<p>sw_pjpeg_e Progressive JPEG enable: 0 = baseline JPEG 1 = progressive JPEG</p>
23	RW	0x0	<p>sw_pic_interlace_e Coding mode of the current picture: 0 = progressive 1 = interlaced</p>
22	RW	0x0	<p>sw_pic_fieldmode_e Structure of the current picture (residual structure) 0 = frame structure, this means MBAFF structured picture for interlaced sequence 1 = field structure</p>

Bit	Attr	Reset Value	Description
21	RW	0x0	sw_pic_b_e B picture enable for current picture: 0=picture type is I or P depending on sw_pic_inter_e 1=picture type is BI (vc1)/D (mpeg1) or B depending on sw_pic_inter_e (not valid for H264 since its slice based information)
20	RW	0x0	sw_pic_inter_e Picture type. 1= Inter type (P) 0= Intra type (I) See also sw_pic_b_e
19	RW	0x0	sw_pic_topfield_e If field structure is enabled this bit informs which one of the fields is being decoded: 0 = bottom field 1 = top field
18	RW	0x0	sw_fwd_interlace_e Coding mode of forward reference picture: 0 = progressive 1 = interlaced Note: for backward reference picture the coding mode is always same as for current picture.
17	RW	0x0	sw_sorenson_e Sorenson Sparc enable (possible if sw_dec_mode is MPEG- 4) 0 = disabled 1 = H.263 compatible stream with Sorenson escape coding
16	RW	0x0	sw_ref_topfield_e Indicates which field should be used as reference if sw_ref_frames = 0 : 0 = bottom field 1 = top field used only in VC-1 mode
15	RW	0x0	sw_dec_out_dis Disable decoder output picture writing: 0 = Decoder output picture is written to external memory 1 = Decoder output picture is not written to external memory
14	RW	0x0	sw_filtering_dis De-block filtering disable 1 = filtering is disabled for current picture 0 = filtering is enabled for current picture

Bit	Attr	Reset Value	Description
13	RW	0x0	sw_pic_fixed_quant sw_pic_fixed_quant (DEC mode is VC-1 and AVS) 0 = Quantization parameter can vary inside picture 1 = Quantization parameter is fixed (pquant) sw_mvc_e(DEC mode is H264) multi view coding enable. Possible for H264 only
12	RW	0x0	sw_write_mvs_e Direct mode motion vector write enable for current picture / MPEG2 and VP6 motion vector write enable for error concealment purposes: 0 = writing disabled for current picture 1 = the direct mode motion vectors are written to external memory. H264 direct mode motion vectors are written to DPB aside with the corresponding reference picture. Other decoding mode dir mode mvs are written to external memory starting from sw_dir_mv_base
11	RW	0x0	sw_reftopfirst_e Indicates which FWD reference field has been decoded first. 0 = FWD reference bottom field 1 = FWD reference top field
10	RW	0x0	sw_seq_mbaff_e Sequence includes MBAFF coded pictures
9	RW	0x0	sw_picord_count_e h264_high config: Picture order count table read enable. If enabled HW will read picture order counts from memory in the beginning of picture
8	RO	0x0	reserved
7:0	RW	0x00	sw_dec_axi_wr_id Write ID used for decoder writing services in AXI bus (if connected to AXI)

VDPU_SWREG4

Address: Operational Base + offset (0x00010)
 decoder control register 1(picture parameters)

Bit	Attr	Reset Value	Description
31:23	RW	0x000	sw_pic_mb_width Picture width in macroblocks = ((width in pixels + 15) /16)

Bit	Attr	Reset Value	Description
22:19	RW	0x0	sw_mb_width_off The amount of meaningfull horizontal pixels in last MB (width offset) 0 if exactly 16 pixels multiple picture and all the horizontal pixels in last MB are meaningfull
18:11	RW	0x00	sw_pic_mb_height_p Picture height in macroblocks =((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded
10:7	RW	0x0	sw_mb_height_off The amount of menaingfull vertical pixels in last MB (height offset 0 if exactly 16 pixels multiple picture and all the vertical pixels in last MB are meaningfull)
6	RW	0x0	sw_alt_scan_e indicates alternative vertical scan method used for interlaceedd frames
5:3	RW	0x0	sw_pic_mb_w_ext Picture mb width extension. If sw_pic_mb_width does not fit tod 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb)
2:0	RW	0x0	sw_pic_mb_h_ext Picture mb height extension. If sw_pic_mb_height_p does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb)

VDPU_SWREG5

Address: Operational Base + offset (0x00014)

decoder control register2 (stream decoding table selects)

Bit	Attr	Reset Value	Description
31:26	RW	0x00	sw_strm_start_bit Exact bit of stream start word where decoding can be started (assosiates with sw_rlc_vlc_base)
25	RW	0x0	sw_sync_marker_e Sync markers enable: '0' = synch markers are not used, '1' = synch markers are used. For progressive JPEG this indicates that there are restart markers in the stream after restart interval steps

Bit	Attr	Reset Value	Description
24	RW	0x0	sw_type1_quant_e MPEG4: Type 1 quantization enable '0' = type 2 inverse Q method '1' = type 1 inverse Q method (Q-tables used) H264 (h264_high config): scaling matrix enable: '0' = normal transform '1' = use scaling matrix for transform (read from external)
23:19	RW	0x00	sw_ch_qp_offset Chroma Qp filter offset. (For H.264 this offset concerns Cb only)
18:14	RW	0x00	sw_ch_qp_offset2 Chroma Qp filter offset for cr type
13:1	RO	0x0	reserved
0	RW	0x0	sw_fieldpic_flag_e Flag for streamd that field_pic_flag exists in stream

VDPU_SWREG6

Address: Operational Base + offset (0x00018)
decoder control register 3(stream buffer information)

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_start_code_e Bit for indicating stream start code existence: '0' = stream doesn't contain start codes '1' = stream contains start codes
30:25	RW	0x00	sw_init_qp Initial value for quantization parameter (picture quantizer).
24	RW	0x0	sw_ch_8pix_ileav_e Enable for additional chrominance data format writing where decoder writes chrominance in group of 8 pixels of Cb and then corresponding 8 pixels of Cr. Data is written to sw_dec_ch8pix_base. Cannot be used if tiled mode is enabled

Bit	Attr	Reset Value	Description
23:0	RW	0x000000	<p>sw_stream_len Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (associates with sw_rlc_vlc_base).</p> <p>For VC-1/VP6 the buffer must include data for one picture/slice of the picture</p> <p>For H264/MPEG4/H263/MPEG2/MPEG1 the buffer must include at least data for one slice/VP of the picture</p> <p>For JPEG the buffer size must be a multiple of 256 bytes or the amount of data for one picture.</p>

VDPU_SWREG7

Address: Operational Base + offset (0x0001c)
 decoder control register 4(H264, VC-1, VP6 control)

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_cabac_e CABAC enable
30	RW	0x0	sw_blackwhite_e '0' = 4:2:0 sampling format '1' = 4:0:0 sampling format (H264 monochroma)
29	RW	0x0	sw_dir_8x8_infer_e Specifies the method to use to derive luma motion vectors in B_skip, B_Direct_16x16 and B_direct_8x8_inference_flag (see direct_8x8_inference_flag)
28	RW	0x0	sw_weight_pred_e Weighted prediction enable for P slices
27:26	RW	0x0	sw_weight_bipr_idc weighted prediction specification for B slices: "00" = default weighted prediction is applied to B slices "01" = explicit weighted prediction shall be applied to B slices "10" = implicit weighted prediction shall be applied to B slices
25:21	RO	0x0	reserved
20:16	RW	0x00	sw_framenum_len H.264: Bit length of frame_num in data stream RV: frame size length. Informs how many bits in stream are used for frame size (HW discards these bits)

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	sw_framenum current frame_num, used to identify short-term reference frames. Used in reference picture reordering

VDPU_SWREG8

Address: Operational Base + offset (0x00020)

decoder control register 5(H264, VC-1, VP6 and RV control)

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_const_intra_e constrained_intra_pred_flag equal to 1 specifies that intra prediction uses only neighbouring intra macroblocks in prediction. When equal to 0 also neighbouring inter macroblocks are used in intra prediction process.
30	RW	0x0	sw_filt_ctrl_pres deblocking_filter_control_present_flag indicates whether extra variables controlling characteristics of the deblocking filter are present in the slice header.
29	RW	0x0	sw_rdpic_cnt_pres redundant_pic_cnt_present_flag specifies whether redundant_pic_cnt syntax elements
28	RW	0x0	sw_8x8trans_flag_e 8x8 transform flag enable for stream decoding
27:17	RW	0x000	sw_refpic_mk_len Length of decoded reference picture marking bits
16	RW	0x0	sw_idr_pic_e IDR (instantaneous decoding refresh) picture flag.
15:0	RW	0x0000	sw_idr_pic_id idr_pic_id, identifies IDR (instantaneous decoding refresh) picture

VDPU_SWREG9

Address: Operational Base + offset (0x00024)

decoder control register 6

Bit	Attr	Reset Value	Description
31:24	RW	0x00	sw_pps_id pic_parameter_set_id, identifies the picture parameter set that is referred to in the slice header.
23:19	RW	0x00	sw_refidx1_active specifies the maximum reference index that can be used while decoding inter predicted macro blocks.

Bit	Attr	Reset Value	Description
18:14	RW	0x00	sw_refidx0_active Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. This is same as in previous decoders (width increased with q bit)
13:8	RO	0x0	reserved
7:0	RW	0x00	sw_poc_length Length of picture order count field in stream

VDPU_SREG10

Address: Operational Base + offset (0x00028)

Base address for differential motion vector base address

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_diff_mv_base for H264 and MPEG4, RLC mode: Differential motion vector base address.
1:0	RO	0x0	reserved

VDPU_SWREG11

Address: Operational Base + offset (0x0002c)

decoder control register 7

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_i4x4_or_dc_base RLC mode: H.264: Intra prediction 4x4 mode base address. RLC mode: MPEG-4: DC component base address.
29:25	RW	0x00	sw_pinit_rlist_f15 Initial reference picture list for P forward picid 15
24:20	RW	0x00	sw_pinit_rlist_f14 Initial reference picture list for P forward picid 14
19:15	RW	0x00	sw_pinit_rlist_f13 Initial reference picture list for P forward picid 13
14:10	RW	0x00	sw_pint_rlist_f12 Initial reference picture list for P forward picid 12
9:5	RW	0x00	sw_pint_rlist_f11 Initial reference picture list for P forward picid 11
4:0	RW	0x00	sw_pint_rlist_f10 Initial reference picture list for P forward picid 10

VDPU_SWREG12

Address: Operational Base + offset (0x00030)

Base address for RLC data (RLC) / stream start address/decoded

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_rlc_vlc_base RLC mode: Base address for RLC data (swreg3.sw_rlc_mode_e = 1). VLC mode: Stream start address / end addr+I288ess with byte precision (swreg4.rlc_mode_en = 0), start bit number in swreg5.stream_start_bit. When sw_dec_buffer_int is high or sw_dec_e is low this register contains HW return value of last_byte_address (not valid for jpeg) where stream has been read (and used) in accuracy of byte. For debug purposes the last_byte_address is also written when stream error/ASO is detected even though it may not be accurate. VP7/VP8: This base address is used as sw_dct strm0_base including DCT stream for MB rows 0,2n
1:0	RO	0x0	reserved

VDPU_SWREG13

Address: Operational Base + offset (0x00034)

Base address for decoded picture / base address for JPEG deco

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_dec_out_base Video: Base address for decoder output picture. Points directly to start of decoder output picture or field. JPEG/VP8 snapshot: Base address for decoder output luminance picture
1:0	RO	0x0	reserved

VDPU_SWREG14

Address: Operational Base + offset (0x00038)

Base address for reference picture index 0 / base address for J

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer0_base Base address for reference picture index 0. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer0_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer0_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG15

Address: Operational Base + offset (0x0003c)

Base address for reference picture index 1 / JPEG control

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer1_base Base address for reference picture index 1. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer1_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer1_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG15_JPEG_ROI

Address: Operational Base + offset (0x0003c)

JPEG roi control

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19	RW	0x0	sw_jpegroi_in_endian jpeg offset input endian sw_jpegroi_in_endian 0 = big endian (0-1-2-3 order) 1 = little endian (3-2-1-0 order)
18	RW	0x0	sw_jpegroi_in_swap32 jpeg offset input 32-bit swap sw_jpegroi_in_swap32 0: no swapping of 32 bit words 1: 32bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1 byte order (also little endian should be enabled))
17:16	RW	0x0	sw_roi_sample_size ROI MB num sample each time ROI MB num sample each time 00 01 10□ 11
15:12	RW	0x0	sw_roi_distance roi distance The distance between the sample MB and ROI start MB

Bit	Attr	Reset Value	Description
11:10	RW	0x0	sw_roi_out_sel roi output selection ROI output selection 00: output offset/dc 01: output picture 10: output offset/dc and picture 11: output offset/dc
9	RW	0x0	sw_roi_decode roi decode JPEG ROI decode 0: build offset/dc table 1: ROI decode
8	RW	0x0	sw_roi_en roi enable JPEG roi mode enable 0: normal jpeg decode mode 1: JPEG roi mode
7:0	RO	0x0	reserved

VDPU_SWREG16

Address: Operational Base + offset (0x00040)

base address for reference picture index 2 / List of VLC code len

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer2_base Base address for reference picture index 2. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer2_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer2_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG17

Address: Operational Base + offset (0x00044)

Base address for reference picture index 3 / List of VLC code len

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer3_base Base address for reference picture index 3. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer3_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields

Bit	Attr	Reset Value	Description
0	RW	0x0	sw_refer3_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG18

Address: Operational Base + offset (0x00048)

Base address for reference picture index 4 / VC1 control / MPE

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer4_base Base address for reference picture index 4. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer4_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer4_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG19

Address: Operational Base + offset (0x0004c)

Base address for reference picture index 5

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer5_base Base address for reference picture index 5. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer5_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer5_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG20

Address: Operational Base + offset (0x00050)

Base address for reference picture index 6

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer6_base Base address for reference picture index 6. See picture index definition from toplevel_sp

Bit	Attr	Reset Value	Description
1	RW	0x0	sw_refer6_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer6_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG21

Address: Operational Base + offset (0x00054)

Base address for reference picture index 7

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer7_base Base address for reference picture index 7. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer7_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer7_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG22

Address: Operational Base + offset (0x00058)

Base address for reference picture index 8

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer8_base Base address for reference picture index 8. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer8_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer8_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG23

Address: Operational Base + offset (0x0005c)

Base address for reference picture index 9

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer9_base Base address for reference picture index 9. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer9_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer9_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG24

Address: Operational Base + offset (0x00060)

Base address for reference picture index 10

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer10_base Base address for reference picture index 10. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer10_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer10_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG25

Address: Operational Base + offset (0x00064)

Base address for reference picture index 11

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	sw_refer11_base Base address for reference picture index 11. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer11_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer11_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG26

Address: Operational Base + offset (0x00068)

Base address for reference picture index 12

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer12_base Base address for reference picture index 12. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer12_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer12_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG27

Address: Operational Base + offset (0x0006c)

Base address for reference picture index 13

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer13_base Base address for reference picture index 13. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer13_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer13_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG28

Address: Operational Base + offset (0x00070)

Base address for reference picture index 14

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer14_base Base address for reference picture index 14. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer14_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer14_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG29

Address: Operational Base + offset (0x00074)

Base address for reference picture index15

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer15_base Base address for reference picture index 15. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer15_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer15_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

VDPU_SWREG30

Address: Operational Base + offset (0x00078)

Reference picture numbers for index 0 and 1 (H264 VLC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer1_nbr Number for reference picture index 1
15:0	RW	0x0000	sw_refer0_nbr Number for reference picture index 0

VDPU_SWREG31

Address: Operational Base + offset (0x0007c)

Reference picture numbers for index 2 and 3 (H264 VLC) /

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer3_nbr Number for reference picture index 3
15:0	RW	0x0000	sw_refer2_nbr Number for reference picture index 2

VDPU_SWREG32

Address: Operational Base + offset (0x00080)

Reference picture numbers for index 4 and 5 (H264 VLC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer5_nbr Number for reference picture index 5
15:0	RW	0x0000	sw_refer4_nbr Number for reference picture index 4

VDPU_SWREG33

Address: Operational Base + offset (0x00084)

Reference picture numbers for index 6 and 7 (H264 VLC)

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer7_nbr Number for reference picture index 7
15:0	RW	0x0000	sw_refer6_nbr Number for reference picture index 6

VDPU_SWREG34

Address: Operational Base + offset (0x00088)

Reference picture numbers for index 8 and 9 (H264 VLC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer9_nbr Number for reference picture index 9
15:0	RW	0x0000	sw_refer8_nbr Number for reference picture index 8

VDPU_SWREG35

Address: Operational Base + offset (0x0008c)

Reference picture numbers for index 10 and 11 (H264 VLC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer11_nbr Number for reference picture index 11
15:0	RW	0x0000	sw_refer10_nbr Number for reference picture index 10

VDPU_SWREG35_JPEG_ROI

Address: Operational Base + offset (0x0008c)

JPEG roi offset/dc base address

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_jpegdcoff_base JPEG roi offset/dc base address JPEG roi offset/dc base address
1:0	RO	0x0	reserved

VDPU_SWREG36

Address: Operational Base + offset (0x00090)

Reference picture numbers for index 12 and 13 (H264 VLC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer13_nbr Number for reference picture index 13
15:0	RW	0x0000	sw_refer12_nbr Number for reference picture index 12

VDPU_SWREG36_JPEG_ROI

Address: Operational Base + offset (0x00090)

JPEG roi offset/dc length

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16:0	RW	0x00000	sw_jpegdcoff_len sw_jpegdcoff_len The number of 64bit jpegdcoff, it can be used both when sw_roi_decode is 1'b0 or 1'b1

VDPU_SWREG37

Address: Operational Base + offset (0x00094)

Reference picture numbers for index 14 and 15 (H264 VLC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer15_nbr Number for reference picture index 15
15:0	RW	0x0000	sw_refer14_nbr Number for reference picture index 14

VDPU_SWREG38

Address: Operational Base + offset (0x00098)

Reference picture long term flags (H264 VLC) / VPx prediction filt

Bit	Attr	Reset Value	Description
31:22	RW	0x000	sw_pred_bc_tap_3_3 Prediction filter set 3, tap 3
21:12	RW	0x000	sw_pred_bc_tap_4_0 Prediction filter set 4, tap 0
11:2	RW	0x000	sw_perd_bc_tap_4_1 Prediction filter set 4, tap 1
1:0	RO	0x0	reserved

VDPU_SWREG39

Address: Operational Base + offset (0x0009c)

Reference picture valid flags (H264 VLC) / VPx prediction filter ta

Bit	Attr	Reset Value	Description
31:22	RW	0x000	sw_pred_bc_tap_4_2 Prediction filter set 4, tap 2
21:12	RW	0x000	sw_pred_bc_tap_4_3 Prediction filter set 4, tap 3
11:2	RW	0x000	sw_pred_bc_tap_5_0 Prediction filter set 5, tap 0
1:0	RO	0x0	reserved

VDPU_SWREG40

Address: Operational Base + offset (0x000a0)

Base address for standard dependent tables

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_qtable_base Base address for standard dependent tables: JPEG= AC,DC, QP tables MPEG4=QP table base address if type 1 quantization is used MPEG2=QP table base address H.264=base address for various tables VP6,VP7,VP8=base address for stream decoding tables RV=base address for picture slice sizes
1:0	RO	0x0	reserved

VDPU_SWREG41

Address: Operational Base + offset (0x000a4)

Base address for direct mode motion vectors

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_dir_mv_base Direct mode motion vector write/read base address. For H264 this is used only for direct mode motion vector write base. Progressive JPEG: ACDC coefficient read/write base address. If current round is for DC components this base address is pointing to luminance (separate base addresses for chrominances), for AC component rounds this base is used for current type
1:0	RO	0x0	reserved

VDPU_SWREG42

Address: Operational Base + offset (0x000a8)

bi_dir initial ref pic list register (0-2)/ VP6 prediction filter taps

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b2 Initial reference picture list for bi- direct backward picid 2
24:20	RW	0x00	sw_binit_rlist_f2 Initial reference picture list for bi- direct forward picid 2
19:15	RW	0x00	sw_binit_rlist_b1 Initial reference picture list for bi- direct backward picid 1
14:10	RW	0x00	sw_binit_rlist_f1 Initial reference picture list for bi- direct forward picid 1
9:5	RW	0x00	sw_binit_rlist_b0 Initial reference picture list for bi- direct backward picid 0

Bit	Attr	Reset Value	Description
4:0	RW	0x00	sw_binit_rlist_f0 Initial reference picture list for bi- direct forward picid 0

VDPU_SWREG43

Address: Operational Base + offset (0x000ac)

bi-dir initial ref pic list register (3-5)/ VP6 prediction filter taps

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b5 Initial reference picture list for bi- direct backward picid 5
24:20	RW	0x00	sw_binit_rlist_f5 Initial reference picture list for bi- direct forward picid 5
19:15	RW	0x00	sw_binit_rlist_b4 Initial reference picture list for bi- direct backward picid 4
14:10	RW	0x00	sw_binit_rlist_f4 Initial reference picture list for bi- direct forward picid 4
9:5	RW	0x00	sw_binit_rlist_b3 Initial reference picture list for bi- direct backward picid 3
4:0	RW	0x00	sw_binit_rlist_f3 Initial reference picture list for bi- direct forward picid 3

VDPU_SWREG44

Address: Operational Base + offset (0x000b0)

bi-dir initial ref pic list register (6-8)/ VP6 prediction filter taps

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b8 Initial reference picture list for bi- direct backward picid 8
24:20	RW	0x00	sw_binit_rlist_f8 Initial reference picture list for bi- direct forward picid 8
19:15	RW	0x00	sw_binit_rlist_b7 Initial reference picture list for bi- direct backward picid 7
14:10	RW	0x00	sw_binit_rlist_f7 Initial reference picture list for bi- direct forward picid 7
9:5	RW	0x00	sw_binit_rlist_b6 Initial reference picture list for bi- direct backward picid 6

Bit	Attr	Reset Value	Description
4:0	RW	0x00	sw_binit_rlist_f6 Initial reference picture list for bi- direct forward picid 6

VDPU_SWREG45

Address: Operational Base + offset (0x000b4)

bi-dir initial ref pic list register (9- 11) / VP6 prediction filter taps

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b11 Initial reference picture list for bi- direct backward picid 11
24:20	RW	0x00	sw_binit_rlist_f11 Initial reference picture list for bi- direct forward picid 11
19:15	RW	0x00	sw_binit_rlist_b10 Initial reference picture list for bi- direct backward picid 10
14:10	RW	0x00	sw_binit_rlist_f10 Initial reference picture list for bi- direct forward picid 10
9:5	RW	0x00	sw_binit_rlist_b9 Initial reference picture list for bi- direct backward picid 9
4:0	RW	0x00	sw_binit_rlist_f9 Initial reference picture list for bi- direct forward picid 9

VDPU_SWREG46

Address: Operational Base + offset (0x000b8)

bi-dir initial ref pic list register (12- 14) / VP7,VP8 quantization v

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b14 Initial reference picture list for bi- direct backward picid 14
24:20	RW	0x00	sw_binit_rlist_f14 Initial reference picture list for bi- direct forward picid 14
19:15	RW	0x00	sw_binit_rlist_b13 Initial reference picture list for bi- direct backward picid 13
14:10	RW	0x00	sw_binit_rlist_f13 Initial reference picture list for bi- direct forward picid 13
9:5	RW	0x00	sw_binit_rlist_b12 Initial reference picture list for bi- direct backward picid 12

Bit	Attr	Reset Value	Description
4:0	RW	0x00	sw_binit_rlist_f12 Initial reference picture list for bi- direct forward picid 12

VDPU_SWREG47

Address: Operational Base + offset (0x000bc)

bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,V

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_pinit_rlist_f3 Initial reference picture list for P forward picid 3
24:20	RW	0x00	sw_pinit_rlist_f2 Initial reference picture list for P forward picid 2
19:15	RW	0x00	sw_pinit_rlist_f1 Initial reference picture list for P forward picid 1
14:10	RW	0x00	sw_pinit_rlist_f0 Initial reference picture list for P forward picid 0
9:5	RW	0x00	sw_binit_rlist_b15 Initial reference picture list for bi- direct backward picid 15
4:0	RW	0x00	sw_binit_rlist_f15 Initial reference picture list for bi- direct forward picid 15

VDPU_SWREG48

Address: Operational Base + offset (0x000c0)

Error concealment register

Bit	Attr	Reset Value	Description
31:23	RW	0x000	sw_startmb_x Start MB from SW for X dimension. Used in error concealment case
22:15	RW	0x00	sw_startmb_y Start MB from SW for Y dimension. Used in error concealment case
14:0	RO	0x0	reserved

VDPU_SWREG49

Address: Operational Base + offset (0x000c4)

Prediction filter tap register for H264, MPEG4, VC1, VP6

Bit	Attr	Reset Value	Description
31:22	RW	0x000	sw_pred_bc_tap_0_0 Prediction filter set 0, tap 0
21:12	RW	0x000	sw_pred_bc_tap_0_1 Prediction filter set 0, tap 1

Bit	Attr	Reset Value	Description
11:2	RW	0x000	sw_pred_bc_tap_0_2 Prediction filter set 0, tap 2
1:0	RO	0x0	reserved

VDPU_SWREG50

Address: Operational Base + offset (0x000c8)

Synthesis configuration register decoder 0 (read only)

Bit	Attr	Reset Value	Description
31	RO	0x1	SW_DEC_MPEG2_PROF Decoding format support, MPEG-2 / MPEG-1 '0' = not supported '1' = supported
30:29	RO	0x3	SW_DEC_VC1_PROF Decoding format support, VC-1 0 = not supported 1 = supported up to simple profile 2 = supported up to main profile 3 = supported up to advanced profile
28	RO	0x1	SW_DEC_JPEG_PROF Decoding format support, JPEG 0 = not supported 1 = supported
27:26	RO	0x2	SW_DEC_MPEG4_PROF Decoding format support, MPEG-4 / H.263 0 = not supported 1 = supported up to simple profile 2 = supported up to advanced simple profile
25:24	RO	0x3	SW_DEC_H264_PROF Decoding format support, H.264 0 = not supported 1 = supported up to baseline profile 2 = supported up to high profile labeled stream with restricted high profile tools
23	RO	0x1	SW_DEC_VP6_PROF Decoding format support, VP6 0 = not supported 1 = supported
22	RO	0x0	SW_DEC_PJPEG_EXIT Progressive JPEG support: '0' = Not supported '1' = supported
21	RO	0x1	SW_DEC_OBUFF_LEVEL Decoder output buffer level: '0' = 1 MB buffering is used '1' = 4 MB buffering is used
20	RO	0x1	SW_REF_BUFF_EXIST Field0000 Description
19:16	RO	0x5	SW_DEC_BUS_STRD Field0000 Description
15:14	RO	0x1	SW_DEC_SYNTH_LAN

Bit	Attr	Reset Value	Description
13:12	RO	0x2	SW_DEC_BUS_WIDTH 0 = error 1 = 32 bit bus 2 = 64 bit bus 3 = 128 bit bus
11	RO	0x1	SW_DEC_SOREN_PRO Decoding format support, Sorenson '0' = not supported '1' = supported
10:0	RO	0x780	SW_DEC_MAX_OWIDTH Max configured decoder video resolution that can be decoded. Informed as width of the picture in pixels

VDPU_SWREG51

Address: Operational Base + offset (0x000cc)

Reference picture buffer control register

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_refbu_e Refer picture buffer enable: '0' = refer picture buffer disabled '1' = refer picture buffer enabled. Valid if picture size is QVGA or more
30:19	RW	0x000	sw_refbu_thr Reference buffer disable threshold value (cache miss amount). Used to buffer shut down (if more misses than allowed)
18:14	RW	0x00	sw_refbu_picid The used reference picture ID for reference buffer usage
13	RW	0x0	sw_refbu_eval_e Enable for HW internal reference ID calculation. If given threshold level is reached by any picture_id after first MB row, that picture_id is used for reference buffer fill for rest of the picture
12	RW	0x0	sw_refbu_fparmod_e Field parity mode enable. Used in rebufferd evaluation mode '0' = use the result field of the evaluation '1' = use the parity mode field
11:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:0	RW	0x000	sw_refbu_y_offset Y offset for refbufferd. This coordinate is used to compensate the global motion of the video for better buffer hit rate

VDPU_SWREG52

Address: Operational Base + offset (0x000d0)

Reference picture buffer information register 1 (read only)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refbu_hit_sum The sum of the refbufferd hits of the picture. Determined for each 8x8 luminance partition of the picture. The proceeding of the HW calculation can be read during HW decoding
15:0	RW	0x0000	sw_refbu_intra_sum The sum of the luminance 8x8 intra partitions of the picture. The proceeding of the HW calculation can be read during HW decoding

VDPU_SWREG53

Address: Operational Base + offset (0x000d4)

Reference picture buffer information register 2 (read only)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:0	RW	0x000000	sw_refbu_y_mv_sum The sum of the decoded motion vector y-components of the picture. The first luminance motion vector of each MB is used in calculation. Other motion vectors of the MB are discarded. Each motion vector is saturated between -256 - 255 before calculation. The proceeding of the HW calculation can be read during HW decoding

VDPU_SWREG54

Address: Operational Base + offset (0x000d8)

Synthesis configuration register decoder 1 (read only)

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	RO	0x1	SW_DEC_JPEG_EXTENS JPEG sampling support extension for 411 and 444 samplings and support for bigger max resolution than 16 Mpix (up to 67Mpixels): '0' = not supported '1' = supported
30	RO	0x1	SW_DEC_REFBU_ILACE Refbufferd support for interlaced content: '0' = not supported '1' = supported
29	RO	0x1	SW_DEC_DIVX_PROF DIVX Support : '0' = not supported '1' = supported
28	RO	0x0	SW_REF_BUFF2_EXIST Reference picture buffer 2 usage: '0' = not supported '1' = reference buffer 2 is used
27:26	RO	0x1	SW_DEC_RV_PROF Decoding format support, RV 0 = not supported 1 = supported up to 2 = NA
25	RO	0x0	SW_DECRTL_ROM ROM implementation type (If design includes ROMs) '0': ROMs are implemented from actual ROM units '1': ROMs are impelemted from RTL
24	RO	0x1	SW_DEC_VP7_PROF Decoding format support, VP7 0 = not supported 1 = supported
23	RO	0x1	SW_DEC_VP8_PROF Decoding format support, VP8 0 = not supported 1 = supported
22	RO	0x1	SW_DEC_AVSPROF Decoding format support, AVS 0 = not supported 1 = supported
21:20	RO	0x1	SW_DEC_MVC_PROF Decoding format support, MVC 0 = not supported 1 = supported

Bit	Attr	Reset Value	Description
19	RO	0x1	SW_DEC_VP8SNAP_E Decoding format support, VP8 snapshot 0 = not supported bigger than 1080p resolution 1 = supported upto 16kx16k pixel resolution (defined max)
18:17	RO	0x1	SW_DEC_TILED_L Tiled mode support level 0 = not supported 1 = supported with 8x4 tile size 2,3 = reserved
16:0	RO	0x0	reserved

VDPU_SWREG55

Address: Operational Base + offset (0x000dc)

Reference picture buffer 2 / Advanced prefetch control register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:0	RW	0x0000	sw_apf_threshold Advanced prefetch threshold value. If current MB exceeds the threshold the advanced mode is not used. Value 0 disables threshold usage and advanced prefetch usage is restricted by internal memory limitation only

VDPU_SWREG56

Address: Operational Base + offset (0x000e0)

Reference buffer information register 3 (read only)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refbu_top_sum The sum of the top partitions of the picture
15:0	RW	0x0000	sw_refbu_bot_sum The sum of the bottom partitions of the picture

VDPU_SWREG57_INTRA_INTER

Address: Operational Base + offset (0x000e4)

intra_dll3t,intra_dblspeed,inter_dblspeed,stream_len_hi

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:8	RW	0x00	debug_service debug_service signals service_wr[2:0], service_rd[3:0]

Bit	Attr	Reset Value	Description
7	RW	0x0	sw_cache_en cache enable 1'b1: cache enable 1'b0: cache disable when sw_cache_en is 1'b1, sw_pref_sigchan should also be 1'b1
6	RW	0x0	sw_pref_sigchan prefetch single channel enable 1'b1: prefetch single channel enable
5	RW	0x0	sw_axiwr_sel axi write master select 1'b0: auto sel encoder axi signals and decoder axi signals 1'b1: sel decoder axi signals (it only use to set bu_dec_e to 1'b0 in the middle of a frame)
4	RW	0x0	sw_paral_bus paral_bus enable when it is set to 1'b1, the axi support read and write service parallel; when it is set to 1'b0, the axi only support read and write serial
3	RW	0x0	sw_intra dbl3t sw_intra dbl3t In chroma dc intra prediction, when this bit is enable, there will 3 cycle enhance for every block
2	RW	0x0	sw_intra dblspeed intra double speed enable Intra double speed enable
1	RW	0x0	sw_inter dblspeed inter double speed enable Inter double speed enable
0	RW	0x0	sw_stream_len_hi stream length high bit The extension bit of sw_stream_len

VDPU_SWREG58

Address: Operational Base + offset (0x000e8)

Decoder debug register 0 (read only)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	debug_mv_req mvst_mv_req signal value
29	RO	0x0	debug_rlc_req prtr_res_y_req signal value
28	RO	0x0	debug_res_y_req prtr_res_y_req signal value
27	RO	0x0	debug_res_c_req prtr_res_c_req signal value
26	RO	0x0	debug_strm_da_e strm_da_e signal value

Bit	Attr	Reset Value	Description
25	RO	0x0	debug_framerdy dfbu_framerdy signal value
24	RO	0x0	debug_filter_req dfbu_req_e signal value
23	RO	0x0	debug_referreq0 prbu_referreq0 signal value
22	RO	0x0	debug_referreq1 prbu_referreq1 signal value
21	RO	0x0	reserved
20:0	RO	0x0000000	debug_dec_mb_count HW internal MB counter value

VDPU_SWREG59

Address: Operational Base + offset (0x000ec)

H264 Chrominance 8 pixel interleaved data base

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_dec_ch8pix_base Base address for additional chrominance data format where chrominance is interleaved in group of 8 pixels. The usage is enabled by sw_ch_8pix_ileav_e
1:0	RO	0x0	reserved

VDPU_SWREG60

Address: Operational Base + offset (0x000f0)

Interrupt register post-processor

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	sw_pp_bus_int Interrupt status bit bus. Error response from bus. In pipeline mode this bit is not used
12	RW	0x0	sw_pp_rdy_int Interrupt status bit pp. When this bit is high post processor has processed a picture in external mode. In pipeline mode this bit is not used.
11:9	RO	0x0	reserved
8	RW	0x0	sw_pp_irq Post-processor IRQ. SW will reset this after interrupt is handled. HINTpp is not used for pp if IRQ disable pp is high (sw_pp_irq_n_e = 1). In pipeline mode this bit is not used
7:5	RO	0x0	reserved
4	RW	0x0	sw_pp_irq_dis Post-processor IRQ disable. When high, there are no interrupts from HW concerning post processing. Polling must be used to see the interrupt
3:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	sw_pp_pipeline_e Decoder –post-processing pipeline enable: 0 = Post-processing is processing different picture than decoder or is disabled 1 = Post-processing is performed in pipeline with decoder
0	RW	0x0	sw_pp_e External mode post-processing enable. This bit will start the post-processing operation. Not to be used if PP is in pipeline with decoder (sw_pp_pipeline_e = 1). HW will reset this when picture is post-processed.

VDPU_SWREG61

Address: Operational Base + offset (0x000f4)

Device configuration register post-processor

Bit	Attr	Reset Value	Description
31:24	RW	0x00	sw_pp_axi_rd_id Read ID used for AXI PP read services (if connected to AXI)
23:16	RW	0x00	sw_pp_axi_wr_id Write ID used for AXI PP write services (if connected to AXI)
15	RO	0x0	reserved
14	RW	0x0	sw_pp_scmd_dis AXI Single Command Multiple Data disable.
13	RW	0x0	sw_pp_in_a2_endsel Endian/swap select for Alpha blend input source 2: '0' = Use PP in endian/swap definitions (sw_pp_in_endian, sw_pp_in_swap) '1' = Use Ablend source 1 endian/swap definitions
12	RW	0x0	sw_pp_in_a1_swap32 Alpha blend source 1 input 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
11	RW	0x0	sw_pp_in_a1_endian Alpha blend source 1 input data byte endian mode. 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)

Bit	Attr	Reset Value	Description
10	RW	0x0	sw_pp_in_swap32_e PP input 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
9	RW	0x0	sw_pp_data_disc_e PP data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally.
8	RW	0x0	sw_pp_clkgate_e PP dynamic clock gating enable: 1 = Clock is gated from PP structures that are not used 0 = Clock is running for all PP structures Note: Clock gating value can be changed only when PP is not enabled
7	RW	0x0	sw_pp_in_endian PP input picture byte endian mode. Used only if PP is in standalone mode. If PP is running pipelined with the decoder, this bit has no effect. 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)
6	RW	0x0	sw_pp_out_endian PP output picture endian mode for YCbCr data or for any data if config value SW_PP_OEN_VERSION=1 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order) (NOTE: For SW_PP_OEN_VERSION=0 16 bit RGB data this bit works as pixel swapping bit. For 32 bit RGB this bit has no meaning)
5	RW	0x0	sw_pp_out_swap32_e PP output data word swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order (also little endian should be enabled))
4:0	RW	0x00	sw_pp_max_burst Maximum burst length for PP bus transactions. 1-16

VDPU_SWREG62

Address: Operational Base + offset (0x000f8)

Deinterlace control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_deint_e De-interlace enable. Input data is in interlaced format and deinterlacing needs to be performed
30	RO	0x0	reserved
29:16	RW	0x0000	sw_deint_threshold Threshold value used in deinterlacing
15	RW	0x0	sw_deint_blend_e Blend enable for de-interlacing
14:0	RW	0x0000	sw_deint_edge_det Edge detect value used for deinterlacing

VDPU_SWREG63

Address: Operational Base + offset (0x000fc)

base address for reading post-processing input picture lumina

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_pp_in_lu_base Base address for post-processing input luminance picture. If PP input picture is fetched from fields this base address is used to point to topfield of the picture. Used in external mode only.
1:0	RO	0x0	reserved

VDPU_SWREG64

Address: Operational Base + offset (0x00100)

Base address for reading post-processing input picture Cb/Ch

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_pp_in_cb_base Base address for post-processing input Cb picture or for both chrominance pictures (if chrominances interleaved). If PP input picture is fetched from fields this base address is used to point to topfield of the picture. Used in external mode only
1:0	RO	0x0	reserved

VDPU_SWREG65

Address: Operational Base + offset (0x00104)

Base address for reading post-processing input picture Cr

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_pp_in_cr_base Base address for post-processing input cr picture. Used in external mode only
1:0	RO	0x0	reserved

VDPU_SWREG66

Address: Operational Base + offset (0x00108)

Base address for writing post-processed picture luminance/RGB

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_pp_out_lu_base Base address for post-processing output picture (luminance/YUYV/RGB).

VDPU_SWREG67

Address: Operational Base + offset (0x0010c)

Base address for writing post-processed picture Ch

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_pp_out_ch_base Base address for post-processing output chrominance picture (interleaved chrominance).

VDPU_SWREG68

Address: Operational Base + offset (0x00110)

Register for contrast adjusting

Bit	Attr	Reset Value	Description
31:24	RW	0x00	sw_contrast_thr1 Threshold value 1, used with contrast adjusting
23:20	RO	0x0	reserved
19:10	RW	0x000	sw_contrast_off2 Offset value 2, used with contrast adjusting
9:0	RW	0x000	sw_contrast_off1 Offset value 1, used with contrast adjusting

VDPU_SWREG69

Address: Operational Base + offset (0x00114)

Register for colour conversion and contrast adjusting

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_pp_in_start_ch For YUYV 422 input format. Enable for start_with_chrominance. '0' =the order is Y0CbY0Cr or Y0CrY0Cb '1'= the order is CbY0CrY0 or CrY0CbY0
30	RW	0x0	sw_pp_in_cr_first For YUYV 422 input format and YCbCr 420 semiplanar format. Enable for Cr first (before Cb) '0' =the order is Y0CbY0Cr or CbY0CrY0 (if 420 semiplanar chrominance: CbCrCbCr) '1'= the order is Y0CrY0Cb or CrY0CbY0 (if 420 semiplanar chrominance: CrCbCrCb)
29	RW	0x0	sw_pp_out_start_ch For YUYV 422 output format. Enable for start_with_chrominance. '0' =the order is Y0CbY0Cr or Y0CrY0Cb '1'= the order is CbY0CrY0 or CrY0CbY0

Bit	Attr	Reset Value	Description
28	RW	0x0	sw_pp_out_cr_first For YUYV 422 output format. Enable for Cr first (beforeCb) '0' =the order is Y0CbY0Cr or CbY0CrY0 '1'= the order is Y0CrY0Cb or CrY0CbY0
27:18	RW	0x000	sw_color_coeffa2 Coefficient a2, used with Y pixel to calculate all color components
17:8	RW	0x000	sw_color_coeffa1 Coefficient a1, used with Y pixel to calculate all color components
7:0	RW	0x00	sw_contrast_thr2 Threshold value 2, used with contrast adjusting

VDPU_SWREG70

Address: Operational Base + offset (0x00118)

Register for colour conversion 0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:20	RW	0x000	sw_color_coeffd Coefficient d, used with Cb to calculate green component value
19:10	RW	0x000	sw_color_coeffc Coefficient c, used with Cr to calculate green component value
9:0	RW	0x000	sw_color_coeffb Coefficient b, used with Cr to calculate red component value

VDPU_SWREG71

Address: Operational Base + offset (0x0011c)

Register for colour conversion 1 + rotation mode

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:21	RW	0x000	sw_crop_startx Start coordinate x for the cropped area in macroblocks.
20:18	RW	0x0	sw_rotation_mode Rotation mode: 000 = rotation disabled 001 = rotate + 90 010 = rotate -90 011 = horizontal flip (mirror) 100 = vertical flip 101 = rotate 180
17:10	RW	0x00	sw_color_coefff Coefficient f, used with Y to adjust brightness

Bit	Attr	Reset Value	Description
9:0	RW	0x000	sw_color_coeffe Coefficient e, used with Cb to calculate blue component value

VDPU_SWREG72

Address: Operational Base + offset (0x00120)

PP input size and -cropping register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	sw_crop_starty Start coordinate y for the cropped area in macroblocks.
23	RO	0x0	reserved
22:18	RW	0x00	sw_rangemap_coef_y Range map value for Y component (RANGE_MAPY+9 in VC-1 standard)
17	RO	0x0	reserved
16:9	RW	0x00	sw_pp_in_height PP input picture height in MBs. Can be cropped from a bigger input picture in external mode
8:0	RW	0x000	sw_pp_in_width PP input picture width in MBs. Can be cropped from a bigger input picture in external mode

VDPU_SWREG73

Address: Operational Base + offset (0x00124)

PP input picture base address for Y bottom field

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_pp_bot_yin_base PP input Y base for bottom field
1:0	RO	0x0	reserved

VDPU_SWREG74

Address: Operational Base + offset (0x00128)

PP input picture base for Ch bottom field

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_pp_bot_cin_base PP input C base for bottom field (mixed chrominance)
1:0	RO	0x0	reserved

VDPU_SWREG79

Address: Operational Base + offset (0x0013c)

Scaling ratio register 1 & padding for B

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_rangemap_y_e Range map enable for Y component (RANGE_MAPY_FLAG in VC-1 standard). For VC1 main profile this bit is used as range expansion enable
30	RW	0x0	sw_rangemap_c_e Range map enable for chrominance component RANGE_MAPUV_FLAG in VC-1 standard)
29	RW	0x0	sw_ycbcr_range Defines the YCbCr range in RGB conversion: 0 = 16µ for Y, 16µ for Chrominance 1 = 0µ for all components
28	RW	0x0	sw_rgb_pix_in32 RGB pixel amount/ 32 bit word 0 = 1 RGB pixel/32 bit 1 = 2 RGB pixels/32 bit
27:23	RW	0x00	sw_rgb_r_padd Amount of ones that will be padded in front of the R- component
22:18	RW	0x00	sw_rgb_g_padd Amount of ones that will be padded in front of the G- component
17:0	RW	0x00000	sw_scale_wratio Scaling ratio for width (outputw-1/inputw-1)

VDPU_SWREG80

Address: Operational Base + offset (0x00140)

Scaling register 0 ratio & padding for R and G

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30	RW	0x0	sw_pp_fast_scale_e 0 = fast downscaling is not enabled 1 = fast downscaling is enabled. The quality of the picture is decreased but performance is improved
29:27	RW	0x0	sw_pp_in_struct PP input data picture structure: 0 = Top field / progressive frame structure: Read input data from top field base address /frame base address and read every line 1 = Bottom field structure: Read input data from bottom field base address and read every line. 2 = Interlaced field structure: Read input data from both top and bottom field base address and take every line from each field. 3 = Interlaced frame structure: Read input data from both top and bottom field base address and take every second line from each field. 4 = Ripped top field structure: Read input data from top field base address and read every second line. 5 = Ripped bottom field structure: Read input data from bottom field base address and read every second line
26:25	RW	0x0	sw_hor_scale_mode Horizontal scaling mode: 00 = Off 01 = Upscale 10 = Downscale
24:23	RW	0x0	sw_ver_scale_mode Vertical scaling mode: 00 = Off 01 = Upscale 10 = Downscale
22:18	RW	0x00	sw_rgb_b_padd Amount of ones that will be padded in front of the B-component

Bit	Attr	Reset Value	Description
17:0	RW	0x00000	sw_scale_hratio Scaling ratio for height (outpuh-1/inpuh-1)

VDPU_SWREG81

Address: Operational Base + offset (0x00144)

Scaling ratio register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_wscale_invra Inverse scaling ratio for width, or ch (inputw-1 / outputw-1)
15:0	RW	0x0000	sw_hscale_invra Inverse scaling ratio for height or cv (inpuh-1 / outpuh-1)

VDPU_SWREG82

Address: Operational Base + offset (0x00148)

Rmask register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_r_mask Bit mask for R component (and alpha channel)

VDPU_SWREG83

Address: Operational Base + offset (0x0014c)

Gmask register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_g_mask Bit mask for G component (and alpha channel)

VDPU_SWREG84

Address: Operational Base + offset (0x00150)

Bmask register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_b_mask Bit mask for B component (and alpha channel)

VDPU_SWREG85

Address: Operational Base + offset (0x00154)

Post-processor control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:29	RW	0x0	<p>sw_pp_in_format PP input picture data format 0 = YUYV 4:2:2 interleaved (supported only in external mode) 1 = YCbCr 4:2:0 Semi-planar in linear raster-scan format 2 = YCbCr 4:2:0 planar (supported only in external mode) 3 = YCbCr 4:0:0 (supported only in pipelined mode) 4 = YCbCr 4:2:2 Semi-planar (supported only in pipelined mode) 5 = YCbCr 4:2:0 Semi-planar in tiled format (supported only in external mode (8170 decoder only)) 6 = YCbCr 4:4:0 Semi-planar (supported only in pipelined mode, possible for jpeg only) 7 = Escape pp input data format. Defined in swreg86</p>
28:26	RW	0x0	<p>sw_pp_out_format PP output picture data format: 0 = RGB 1 = YCbCr 4:2:0 planar (Not supported) 2 = YCbCr 4:2:2 planar (Not supported) 3 = YUYV 4:2:2 interleaved 4 = YCbCr 4:4:4 planar (Not supported) 5 = YCh 4:2:0 chrominance interleaved 6 = YCh 4:2:2 (Not supported) 7 = YCh 4:4:4 (Not supported)</p>
25:15	RW	0x000	<p>sw_pp_out_height Scaled picture height in pixels (Must be dividable by 2 or by any if Pixel Accurate PP output configuration is enabled) Max scaled picture height is 1920 pixels or maximum three times the input source height minus 8 pixels</p>
14:4	RW	0x000	<p>sw_pp_out_width Scaled picture width in pixels. Must be dividable by 8 or by any if Pixel Accurate PP output configuration is enabled. Max scaled picture width is 1920 pixels or maximum three times the input source width minus 8 pixels</p>

Bit	Attr	Reset Value	Description
3	RW	0x0	sw_pp_out_tiled_e Tiled mode enable for PP output. Can be used only for YCbYCr 422 output format. Can be used only if correpongind configuration supports this feature. Tile size is 4x4 pixels.
2	RW	0x0	sw_pp_out_swap16_e PP output swap 16 swaps 16 bit halves inside of 32 bit word. Can be used for 16 bit RGB to change pixel orders but is valid also for any output format NOTE: requires that configuration of SW_PPD_OEN_VERSION=1
1	RW	0x0	sw_pp_crop8_r_e PP input picture width is not 16 pixels multiple. Only 8 pixels of the most right MB of the unrotated input picture is used for PP input.
0	RW	0x0	sw_pp_crop8_d_e PP input picture height is not 16 pixels multiple. Only 8 pixel rows of the most down MB of the unrotated input picture is used for PP input.

VDPU_SWREG86

Address: Operational Base + offset (0x00158)

Mask 1 start coordinate register

Bit	Attr	Reset Value	Description
31:29	RW	0x0	sw_pp_in_format_es Escape PP in format. Used if sw_pp_in_format is defined to 7: 0 0 = YCbCr 4:4:4 1 = YCbCr 4:1:1
28	RO	0x0	reserved
27:23	RW	0x00	sw_rangemap_coef_c Range map value for chrominance component (RANGE_MAPUV+9 in VC-1 standard)

Bit	Attr	Reset Value	Description
22	RW	0x0	sw_mask1_ablend_e Mask 1 alpha blending enable. Instead of masking the output picture the alpha blending is performed. Alpha blending source can be found from alpha blend 1 base address. Alpha blending can be enabled only for RGB/ YUYV 422 data.
21:11	RW	0x000	sw_mask1_starty Vertical start pixel for mask area 1. Defines the y coordinate. Coordinate 0,0 means the up-left corner in PP output luminance picture. See Table 47 for restrictions
10:0	RW	0x000	sw_mask1_startix Horizontal start pixel for mask area 1. Defines the x coordinate. Coordinate 0,0 means the up-left corner in PP output luminance picture. See Table 47 for restrictions

VDPU_SWREG87

Address: Operational Base + offset (0x0015c)

Mask 2 start coordinate register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_mask2_ablend_e Mask 2 alpha blending enable. Instead of masking the output picture the alpha blending is performed. Alpha blending source can be found from alpha blend 2 base address. Alpha blending can be enabled only for RGB/YUYV 422 data.
21:11	RW	0x000	sw_mask_starty Vertical start pixel for mask area 2. Defines the y coordinate. Coordinate 0,0 means the up-left corner in PP output Y picture. See Table 47 for restrictions

Bit	Attr	Reset Value	Description
10:0	RW	0x000	sw_mask2_startx Horizontal start pixel for mask area 2. Defines the x coordinate. Coordinate 0,0 means the up-left corner in PP output Y picture. See Table 47 for restrictions

VDPU_SWREG88

Address: Operational Base + offset (0x00160)

Mask 1 size and PP original width register

Bit	Attr	Reset Value	Description
31:23	RW	0x000	sw_ext_orig_width PP input picture original width in macro blocks.
22	RW	0x0	sw_mask1_e Mask 1 enable. If mask 1 is used this bit is high
21:11	RW	0x000	sw_mask1_endy Mask 1 end coordinate y in pixels (inside of PPD output picture). Range must be between [Mask1StartCoordinateY, ScaledHeight].
10:0	RW	0x000	sw_mask1_endx Mask 1 end coordinate x in pixels (inside of PPD output picture). Range must be between [Mask1StartCoordinateX, ScaledWidth]

VDPU_SWREG89

Address: Operational Base + offset (0x00164)

Mask 2 size register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_mask2_e Mask 2 enable. If mask 1 is used this bit is high

Bit	Attr	Reset Value	Description
21:11	RW	0x000	sw_mask2_endy Mask 2 end coordinate y in pixels (inside of PP output picture). Range must be between [Mask2StartCoordinateY, ScaledHeight].
10:0	RW	0x000	sw_mask2_endx Mask 2 end coordinate x in pixels (inside of PP output picture). Range must be between [Mask2StartCoordinateX, ScaledWidth].

VDPU_SWREG90

Address: Operational Base + offset (0x00168)

PiP register 0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	sw_right_cross_e Right side overcross enable. 0 = No right side overcross, 1 = right side overcross
28	RW	0x0	sw_left_cross_e Left side overcross enable. 0 = No left side overcross, 1 = left side overcross
27	RW	0x0	sw_up_cross_e Upward overcross enable. 0 = No upward overcross, 1 = upward overcross
26	RW	0x0	sw_down_cross_e Downward overcross enable. 0 = No downward overcross, 1 = downward overcross
25:15	RW	0x000	sw_up_cross Amount of upward overcross (vertical pixels outside of display from the upper side). Range must be between [0, ScaledHeight].
14:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:0	RW	0x000	sw_down_cross Amount of downward overcross (vertical pixels outside of display from the down side). Range must be between [0, ScaledHeight].

VDPU_SWREG91

Address: Operational Base + offset (0x0016c)

PiP register 1 and dithering control

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_dither_select_r Dithering control for R channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix
29:28	RW	0x0	sw_dither_select_g Dithering control for G channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix
27:26	RW	0x0	sw_dither_select_b Dithering control for B channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix
25:24	RO	0x0	reserved
23:22	RW	0x0	sw_pp_tiled_mode Input data is in tiled mode (at the moment valid only for YCbCr 420 data, pipeline or external mode): 0 = Tiled mode not used 1 = Tiled mode enabled for 8x4 sized tiles 2,3 = reserved
21:11	RW	0x000	sw_right_cross Amount of right side overcross (Horizontal pixels outside of display from the right side). Range must be between [0, ScaledWidth].

Bit	Attr	Reset Value	Description
10:0	RW	0x000	sw_left_cross Amount of left side overcross (Horizontal pixels outside of display from the left side). Range must be between [0, ScaledWidth].

VDPU_SWREG92

Address: Operational Base + offset (0x00170)

Display width and PP input size extension register

Bit	Attr	Reset Value	Description
31:29	RW	0x0	sw_pp_in_h_ext Extended PP input height. Used with JPEG
28:26	RW	0x0	sw_pp_in_w_ext Extended PP input width. Used with JPEG
25:23	RW	0x0	sw_crop_starty_ext Extended PP input crop start coordinate x. Used with JPEG
22:20	RW	0x0	sw_crop_start_y_ext Extended PP input crop start coordinate y. Used with JPEG
19:12	RO	0x0	reserved
11:0	RW	0x000	sw_display_width Width of the display in pixels. Max HDTV (1920)

VDPU_SWREG93

Address: Operational Base + offset (0x00174)

Display width and PP input size extension register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_abledn1_base Base address for alpha blending input 1 (if mask1 is used in alpha blending mode). Format of data is 24 bit RGB/ YCbCr and endian/swap -mode is as in PP input. Amount of data is informed with mask 1 size or with ablend1_scanline if ablend cropping is supported in configuration.

VDPU_SWREG94

Address: Operational Base + offset (0x00178)

Base address for alpha blend 2 gui component

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_ablend2_base Base address for alpha blending input 2 (if mask2 is used in alpha blending mode). Format of data is 24 bit RGB/ YCbCr and endian/swap -mode is as in PP input. Amount of data is informed with mask 2 size or with ablend2_scanline if ablend cropping is supported in configuration.

VDPU_SWREG95

Address: Operational Base + offset (0x0017c)

Base address for alpha blend 2 gui component

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:13	RW	0x0000	sw_ablend2_scan Scanline width in pixels for Ablend 2. Usage enabled if corresponding configuration bit is enabled
12:0	RW	0x0000	sw_ablend1_scan Scanline width in pixels for Ablend 1. Usage enabled if corresponding configuration bit is enabled

VDPU_SWREG98

Address: Operational Base + offset (0x00188)

PP output width/height extension

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	sw_pp_out_h_ext sw_pp_out_h_ext PP output heightextension
0	RW	0x0	sw_pp_out_w_ext sw_pp_out_w_ext PP output widthextension

VDPU_SWREG99

Address: Operational Base + offset (0x0018c)

PP fuse register (read only)

Bit	Attr	Reset Value	Description
31	RO	0x1	fuse_pp_pp 1 = PP enabled

Bit	Attr	Reset Value	Description
30	RO	0x1	fuse_pp_deint 1 = Deinterlacing enabled
29	RO	0x1	fuse_pp_ablend 1 = Alpha Blending enabled
28:16	RO	0x0	reserved
15	RO	0x1	fuse_pp_maxw_1920 1 = Max PP output width up to 1920 pixels enabled. Priority coded with priority 1
14	RO	0x1	fuse_pp_maxw_1280 1 = Max PP output width up to 1280 pixels enabled. Priority coded with priority 2
13	RO	0x1	fuse_pp_maxw_720 1 = Max PP output width up to 720 pixels enabled. Priority coded with priority 3
12	RO	0x1	fuse_pp_maxw_352 1 = Max PP output width up to 352 pixels enabled. Priority coded with priority 4
11:0	RO	0x0	reserved

VDPU_SWREG100

Address: Operational Base + offset (0x00190)

Synthesis configuration register post-processor (read only)

Bit	Attr	Reset Value	Description
31	RO	0x1	SW_ABLEND_CROP_E Alpha blending support for input cropping: '0': Not supported. External memory must include the exact image of the area being alpha blended '1' Supported. External memory can include a picture from blended area can be cropped. Requires usage of swreg95
30	RO	0x1	SW_PPD_PIXAC_E Pixel Accurate PP output mode exists: '0'= PIP, Scaling and masks can be adjusted by steps of 8 pixels (width) or 2 pixels (height) '1' = PIP, Scaling and masks can be adjusted by steps of 1 pixel for RGB and 2 pixels for subsampled chroma formats (by using bus specific write strobe functionality)

Bit	Attr	Reset Value	Description
29	RO	0x1	SW_PPD_TILED_EXIST PP output YCbYCr 422 tiled support (4x4 pixel tiles) '0'=Not supported '1'=Supported
28	RO	0x1	SW_PPD_DITH_EXIST Dithering exists: '0' = no '1' = yes
27:26	RO	0x3	SW_PPD_SCALE_LEVEL Scaling support: 00 = No scaling 01 = Scaling with lo perfomance architecture 10 = Scaling with high performance architecture 11 = Scaling with high performance architecture + fast
25	RO	0x1	SW_PPD_DEINT_EXIST De-interlacing exists: '0' = no '1' = yes
24	RO	0x1	SW_PPD_BLEND_EXIST Alpha blending exists: '0' = no '1' = yes
23	RO	0x1	SW_PPD_IBUFF_LEVEL PP input buffering level: '0' = 1 MB input buffering is used '1' = 4 MB input buffering is used
22:19	RO	0x0	reserved
18	RO	0x1	SW_PPD_OEN_VERSION PP output endian version: '0' = Endian mode supported for other than RGB '1' = Endian mode supported for any output format
17	RO	0x1	SW_PPD_OBUFF_LEVEL PP output buffering level: '0' = 1 unit output buffering is used '1' = 4 unit output buffering is used
16	RO	0x1	SW_PPD_PP_EXIST PPD exists: '0'=no '1'=yes
15:14	RO	0x1	SW_PPD_IN_TILED_L PPD input tiled mode support level 0 = not supported 1 = 8x4 tile size supported
13:11	RO	0x0	reserved
10:0	RO	0x000	SW_PPD_MAX_OWIDTH Max supported PP output width in pixels

VDPU_SWREG101

Address: Operational Base + offset (0x00194)
soft reset signals

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	sw_soft_reset softreset pulse signal softreset pulse signal write to 1'b1, valid; write to 1'b0, invalid;

17.4.11 VPU MMU Register Summary

Name	Offset	Size	Reset Value	Description
VCODEC_MMU_DTE_ADDR	0x00000	W	0x00000000	MMU current page Table address
VCODEC_MMU_STATUS	0x00004	W	0x00000000	MMU status register
VCODEC_MMU_COMMAND	0x00008	W	0x00000000	MMU command register
VCODEC_MMU_PAGE_FAULT_ADDR	0x0000c	W	0x00000000	MMU logical address of last page fault
VCODEC_MMU_ZAP_ONE_LINE	0x00010	W	0x00000000	MMU Zap cache line register
VCODEC_MMU_INT_RAWSTAT	0x00014	W	0x00000000	MMU raw interrupt status register
VCODEC_MMU_INT_CLEAR	0x00018	W	0x00000000	MMU raw interrupt status register
VCODEC_MMU_INT_MASK	0x0001c	W	0x00000000	MMU raw interrupt status register
VCODEC_MMU_INT_STATUS	0x00020	W	0x00000000	MMU raw interrupt status register
VCODEC_MMU_AUTO_GATING	0x00024	W	0x00000001	mmu auto gating

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

17.4.12 VPU MMU Detail Register Description**VCODEC_MMU_DTE_ADDR**

Address: Operational Base + offset (0x00000)
MMU current page Table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MMU_DTE_ADDR MMU_DTE_ADDR MMU current page Table address

VCODEC_MMU_STATUS

Address: Operational Base + offset (0x00004)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGE_FAULT_BUS_ID page fault bus id Index of master responsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE page fault access The direction of access for last page fault: 0 = Read 1 = Write
4	RO	0x0	REPLAY_BUFFER_EMPTY replay buffer empty The MMU replay buffer is empty
3	RO	0x0	MMU_IDLE mmu idle The MMU is idle when accesses are being translated and there are no unfinished translated accesses.
2	RO	0x0	STAIL_ACTIVE stall active MMU stall mode currently enabled. The mode is enabled by command
1	RO	0x0	PAGE_FAULT_ACTIVE page fault active MMU page fault mode currently enabled . The mode is enabled by command.
0	RO	0x0	PAGING_ENABLED Paging is enabled Paging is enabled

V_CODEC_MMU_COMMAND

Address: Operational Base + offset (0x00008)

MMU command register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	WO	0x0	MMU_CMD mmu cmd MMU_CMD. This can be: 0: MMU_ENABLE_PAGING 1: MMU_DISABLE_PAGING 2: MMU_ENABLE_STALL 3: MMU_DISABLE_STALL 4: MMU_ZAP_CACHE 5: MMU_PAGEFAULT_DONE 6: MMU_FORCE_RESET

V_CODEC_MMU_PAGE_FAULT_ADDR

Address: Operational Base + offset (0x0000c)

MMU logical address of last page fault

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	PAGE_FAULT_ADDR page fault addr address of last page fault

VCODEC_MMU_ZAP_ONE_LINE

Address: Operational Base + offset (0x00010)

MMU Zap cache line register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE zap one line address to be invalidated from the page table cache

VCODEC_MMU_INT_RAWSTAT

Address: Operational Base + offset (0x00014)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error read bus error status
0	RW	0x0	PAGE_FAULT page fault page fault status

VCODEC_MMU_INT_CLEAR

Address: Operational Base + offset (0x00018)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	WO	0x0	READ_BUS_ERROR read bus error write 1 to clear read bus error
0	WO	0x0	PAGE_FAULT page fault clear write 1 to page fault clear

VCODEC_MMU_INT_MASK

Address: Operational Base + offset (0x0001c)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error enable the read bus interrupt source when this bit is set to 1'b1

Bit	Attr	Reset Value	Description
0	RW	0x0	PAGE_FAULT page fault mask enable the page fault interrupt source when this bit is set to 1'b1

V_CODEC_MMU_INT_STATUS

Address: Operational Base + offset (0x00020)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR read bus error read bus error status
0	RO	0x0	PAGE_FAULT page fault page fault status

V_CODEC_MMU_AUTO_GATING

Address: Operational Base + offset (0x00024)

mmu auto gating

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_clkgating mmu_auto_clkgating when it is 1'b1, the mmu will auto gating it self

17.4.13 VDPU Pref_cacheRegister Summary

Name	Offset	Size	Reset Value	Description
pref_cache_VERSIO N	0x00000	W	0xcac20101	VERSION register
pref_cache_SIZE	0x00004	W	0x06170206	L2 cache SIZE
pref_cache_STATUS	0x00008	W	0x00000000	Status register
pref_cache_COMMAN D	0x00010	W	0x00000000	Command setting register
pref_cache_CLEAR_P AGE	0x00014	W	0x00000000	clear page register
pref_cache_MAX_RE ADS	0x00018	W	0x00000001c	maximum read register
pref_cache_PERFCNT _SRC0	0x00020	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT _VAL0	0x00024	W	0x00000000	performance counter 0 value register
pref_cache_PERFCNT _SRC1	0x00028	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT _VAL1	0x0002c	W	0x00000000	performance counter 1 value register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

17.4.14 VDPU_Pref_cache Detail Register Description

pref_cache_VERSION

Address: Operational Base + offset (0x00000)

VERSION register

Bit	Attr	Reset Value	Description
31:16	RO	0xcac2	PRODUCT_ID Field0000 Abstract Field0000 Description
15:8	RO	0x01	VERSION_MAJOR Field0000 Abstract Field0000 Description
7:0	RO	0x01	VERSION_MINOR Field0000 Abstract Field0000 Description

pref_cache_SIZE

Address: Operational Base + offset (0x00004)

L2 cache SIZE

Bit	Attr	Reset Value	Description
31:24	RO	0x06	External_bus_width Field0000 Abstract Log2 external bus width in bits
23:16	RO	0x17	CACHE_SIZE Field0000 Abstract Log2 cache size in bytes
15:8	RO	0x02	ASSOCIATIVITY Field0000 Abstract Log2 associativity
7:0	RO	0x06	LINE_SIZE Field0000 Abstract Log2 line size in bytes

pref_cache_STATUS

Address: Operational Base + offset (0x00008)

Status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	DATA_BUSY Field0000 Abstract set when the cache is busy handling data
0	RW	0x0	CMD_BUSY Field0000 Abstract set when the cache is busy handling commands

pref_cache_COMMAND

Address: Operational Base + offset (0x00010)

Command setting register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:4	RW	0x0	sw_addrb_sel Field0000 Abstract 2'b00: to sel b[14:6] 2'b01: to sel b[15:9], b[7:6] 2'b10: to sel b[16:10], b[7:6] 2'b11: to sel b[17:11], b[7:6]
3	RO	0x0	reserved
2:0	RW	0x0	COMMAND Field0000 Abstract The possible command is 1 = Clear entire cache

pref_cache_CLEAR_PAGE

Address: Operational Base + offset (0x00014)

clear page register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	CLEAR_PAGE Field0000 Abstract writing an address, invalidates all lines in that page from the cache

pref_cache_MAX_READS

Address: Operational Base + offset (0x00018)

maximum read register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x1c	MAX_READS Field0000 Abstract Limit the number of outstanding read transactions to this amount

pref_cache_PERFCNT_SRC0

Address: Operational Base + offset (0x00020)

performance counter 0 source register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	<p>PERFCNT_SRC0 Field0000 Abstract</p> <p>This register holds all the possible source values for Performance Counter 0</p> <p>0: total clock cycles 1: active clock cycles 2: read transactions, master 3: word reads, master 4: read transactions, slave 5: word reads, slave 6: read hit, slave 7: read misses, slave 8: read invalidates, slave 9: cacheable read transactions, slave 10: bad hit nmber, slave</p>

pref_cache_PERFCNT_VAL0

Address: Operational Base + offset (0x00024)

performance counter 0 value register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERFCNT_VAL0 Field0000 Abstract</p> <p>Performance counter 0 value</p>

pref_cache_PERFCNT_SRC1

Address: Operational Base + offset (0x00028)

performance counter 0 source register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>PERFCNT_SRC1 Field0000 Abstract</p> <p>This register holds all the possible source values for Performance Counter 1</p> <p>0: total clock cycles 1: active clock cycles 2: read transactions, master 3: word reads, master 4: read transactions, slave 5: word reads, slave 6: read hit, slave 7: read misses, slave 8: read invalidates, slave 9: cacheable read transactions, slave 10: bad hit nmber, slave</p>

pref_cache_PERFCNT_VAL1

Address: Operational Base + offset (0x0002c)

performance counter 1 value register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERFCNT_VAL1 Field0000 Abstract Performance counter 1 value

17.5 Application Notes

Rockchip Confidential

Chapter 18 Video Output Processor (VOP)

18.1 Overview

Video Output Processor is a video process engine and a display interface from memory frame buffer to display device(HDMI,LCDpanel,MIPI,LVDS, or CVBS). VOP is connected to an AHB bus through an AHB slave and AXI bus through an AXI master. The register setting is configured through the AHB slave interface and the display frame data is read through the AXI master interface.

18.1.1 Features

- **Display interface**

- Parallel RGB LCD Interface: 24-bit(RGB888), 18-bit(RGB666), 15-bit(RGB565)
- TV Interface
 - ◆ RGB2YCbCr, 8bit
 - ◆ TV encoder 10bit out for DAC
 - ◆ RGB888+1080i for HDMI
 - ◆ Parallel RGB HDMI Interface: 24-bit(RGB888 YCbCr444)
- Max output resolution
 - ◆ 1920x1080 for HDMI
 - ◆ 480i/576i for CVBS
 - ◆ Support displaying the same source on HDMI and PAD simultaneously(not support HDMI+vertical PAD)

- **Display process**

- Background layer
 - ◆ programmable 24-bit color
- Win0 layer
 - ◆ RGB888, ARGB888, RGB565, YCbCr422, YCbCr420, YCbCr444
 - ◆ 1/8 to 8 scaling-down and scaling-up engine
 - ◆ Support virtual display
 - ◆ 256 level alpha blending (pre-multiplied alpha support)
 - ◆ Transparency color key
 - ◆ De-flicker support for interlace output
 - ◆ Direct path support
 - ◆ YCbCr2RGB(rec601-mpeg/ rec601-jpeg/rec709)
 - ◆ RGB2YCbCr(BT601/BT709)
- Win1 layer
 - ◆ RGB888, ARGB888, RGB565
 - ◆ Support virtual display

- ◆ 256 level alpha blending (pre-multiplied alpha support)
- ◆ Transparency color key
- ◆ Direct path support
- ◆ RGB2YCbCr(BT601/BT709)
- Hardware cursor:
 - ◆ 8BPP(ARGB888 LUT)
 - ◆ Support two size: 32x32 and 64x64
 - ◆ 256 level alpha blending
 - ◆ Support hwc over panel at right and below side
- Scaler:
 - Display interface
 - ◆ output for LVDS/RGB
 - max output resolution: 1280x800
 - support display input directly bypass
 - ◆ Asynchronous output pixel clock (PLL required)
 - ◆ output dclk from PLL integer or fraction output
 - Image scale
 - ◆ Max input resolution: 1920x1080p
 - ◆ Min input resolution: 720x480p
 - ◆ Max output resolution: 1280x800
 - ◆ Min output resolution: 800x600
 - ◆ support 1:1 bypass using line buffer
 - ◆ scale down
 - Arbitrary non-integer scaling ratio
 - Max 1/4 scaling ratio
 - ◆ scale up
 - Arbitrary non-integer scaling ratio
 - Max 4 scaling ratio
- **Others**
 - Win0 layer and Win1 layer overlay exchangeable
 - BCSH(Brightness,Contrast,Saturation,Hue adjustment)
 - BCSH:YCbCr2RGB(rec601-mpeg/ rec601-jpeg/rec709)
 - BCSH:RGB2YCbCr(BT601/BT709)
 - Support Gamma adjust for PAD
 - Support dither down allegro RGB888to666 RGB888to565 & dither down frc (configurable) RGB888to666
 - Blank and black display

- Standby mode
- Support MMU
- Support QoS request for higher bus priority for win1/HWC
- Support NOC hurry for higher bus priority for win0
- Support DMA stop mode

18.2 Block Diagram

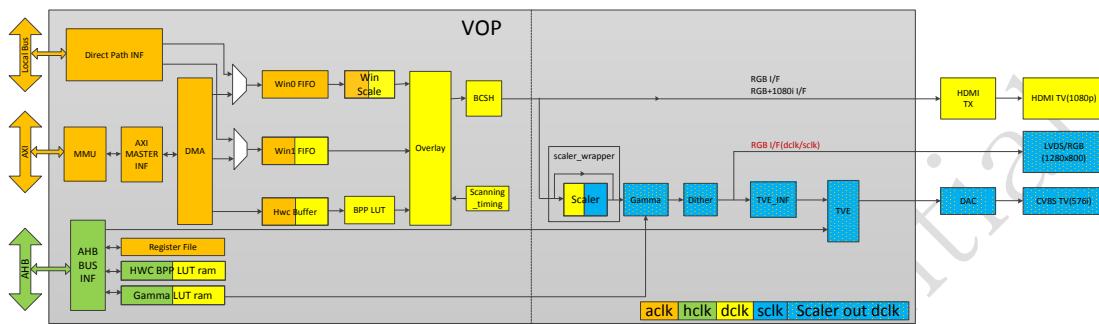


Fig. 18-1 VOP Block Diagram

18.3 Function description

18.3.1 Data Format

VOP master read the frame data from the frame buffer in the system memory. There are total 7 formats supported in three layers.

- Win0: RGB888, ARGB888, RGB565, YCbCr422_SP, YCbCr420_SP, YCbCr444_SP
- Win1: RGB888, ARGB888, RGB565
- Hwc: 8bpp

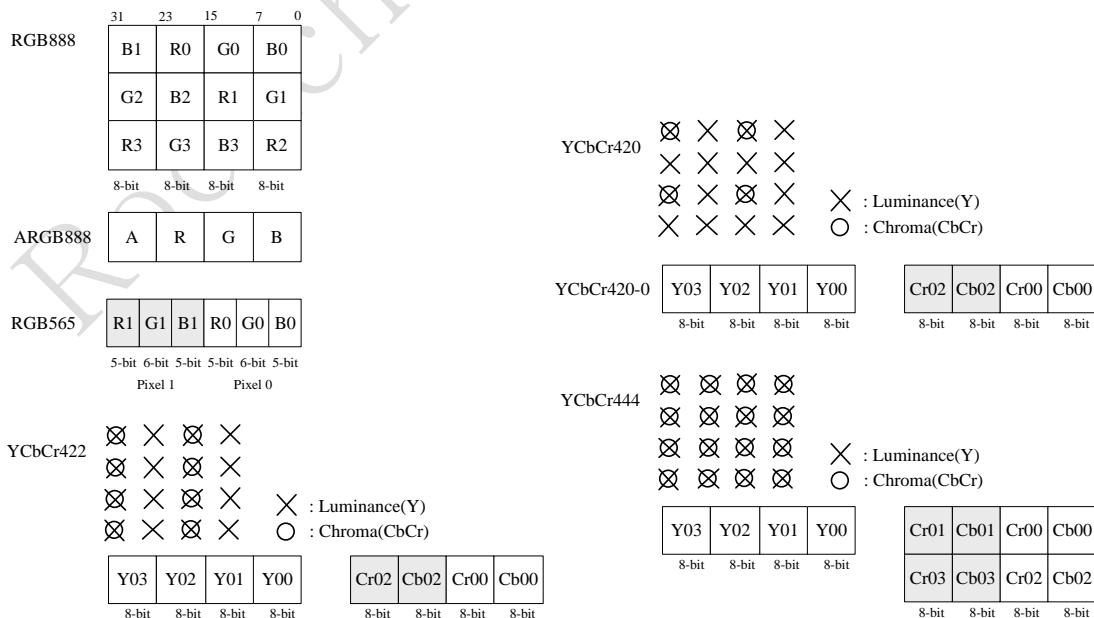


Fig. 18-2 VOP Frame Buffer Data Format

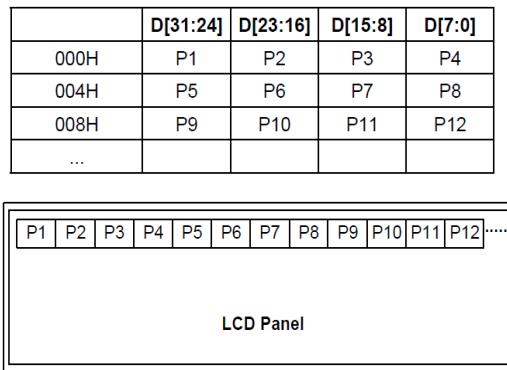


Fig. 18- 3VOP Win1 Palette (8bpp)

Data SWAP function

There are several swap options for different frame data formats. The register is VOP_SYS_CTRL[21:15].

All the data swap types are in the following table.

Table 18- 1VOP Data Swap of Win0 and Win1

Data-swap	<i>RB swap</i>	<i>Alpha swap</i>	<i>Y-M8 swap</i>	<i>CbCr swap</i>
Win0	yes	yes	yes	yes
Win1	yes	yes	No	No

18.3.2 Data path

There are two data input path for VOP to get display layers' pixel data. One is internal DMA; the other is direction path interface.

1. Internal DMA

Internal DMA can fetch the pixel data through AXI bus from system memory (DDR) for all the display layers. Data fetching is driven by display output requirement.

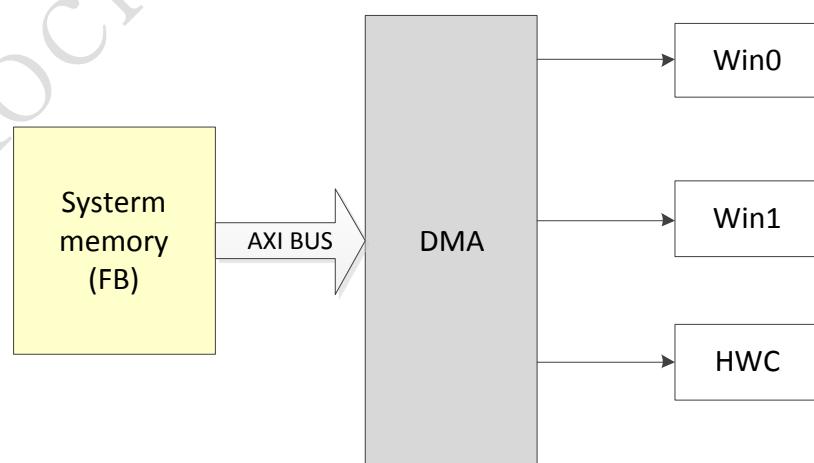


Fig. 18- 4 VOP Internal DMA

2. Direct Path Interface

Direct path interface (DPI) is used for direct image display of external image processing IP. There is a local bus between VOP and external image processing IP for the data transfer.

DPI is connected to WIN0/WIN1 but can only be configured for One layer use (Win0 or Win1) in each frame.

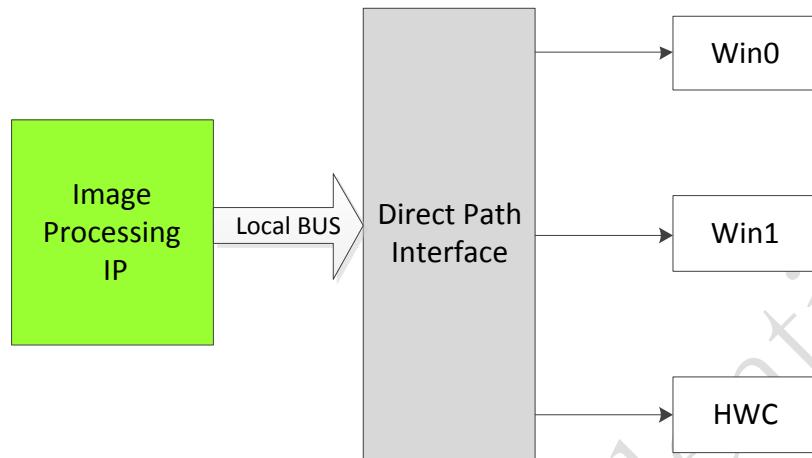


Fig. 18- 5 VOP Direct Path Interface

18.3.3 Virtual display

Virtual display is supported in Win0 and Win1. The active image is part of the virtual (original) image in frame buffer memory. The virtual width is indicated by setting WIN0/WIN1_VIR_STRIDE for different data format.

The virtual stride should be multiples of word (32-bit). That means dummy bytes in the end of virtual line if the real pixels are not 32-bit aligned.

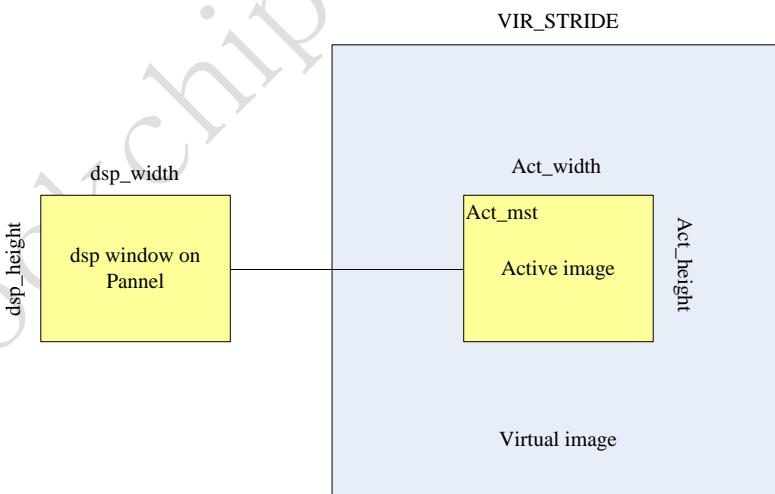


Fig. 18- 6VOP Virtual Display Mode

18.3.4 Scaling

The scaling operation is the imageresizing processof data transfer from the frame buffer memory toLCD panel or TV set.

Horizontal and vertical scaling factor should be set according the window scaling ratio.

1. Scaling factor

Because the Chroma data may have different sampling rate with Luma data in the memory format of YCbCr422/YCbCr420. The scaling factor of Win0 has two couples of factor registers:

VOP_WIN0_SCL_FACTOR_Y/VOP_WIN0_SCL_FACTOR_CBR

Software calculates the scaling factor value using the following equations:

$$y_rgb_vertical_factor = \left(\frac{VOP_WIN0_ACT_INFO[31:16]}{VOP_WIN0_DSP_INFO[31:16]} \right) \times 2^{12}$$

$$y_rgb_horizontal_factor = \left(\frac{VOP_WIN0_ACT_INFO[15:0]}{VOP_WIN0_DSP_INFO[15:0]} \right) \times 2^{12}$$

$$yuv422_yuv444_Cbr_vertical_factor = \left(\frac{VOP_WIN0_ACT_INFO[31:16]}{VOP_WIN0_DSP_INFO[31:16]} \right) \times 2^{12}$$

$$yuv420_Cbr_vertical_factor = \left(\frac{VOP_WIN0_ACT_INFO[31:16]/2}{VOP_WIN0_DSP_INFO[31:16]} \right) \times 2^{12}$$

$$yuv444_Cbr_horizontal_factor = \left(\frac{VOP_WIN0_ACT_INFO[15:0]}{VOP_WIN0_DSP_INFO[15:0]} \right) \times 2^{12}$$

$$yuv422_yuv420_Cbr_horizontal_factor = \left(\frac{VOP_WIN0_ACT_INFO[15:0]/2}{VOP_WIN0_DSP_INFO[15:0]} \right) \times 2^{12}$$

2. Scaling start point offset

The x and y start point of the generated pixels can be adjusted, the offset value is in the range of 0 to 0.99.

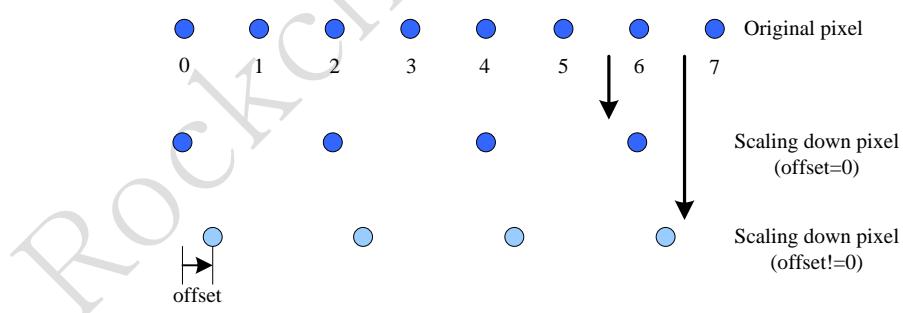


Fig. 18- 7VOP Scaling Down Offset

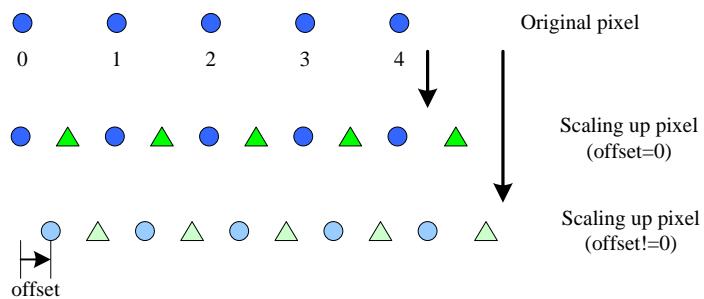


Fig. 18- 8VOP Scaling Up Offset

Table 18- 2VOP Scaling Start Point Offset Registers

scaling down/up start point offset	Offset variable	Register
Win0 YRGB vertical scaling offset	Win0_YRGB_vscl_offset	Win0_SCL_OFFSET [32:24]
Win0 YRGB horizontal scaling offset	Win0_YRGB_hscl_offset	Win0_SCL_OFFSET [23:16]
Win0 Cbr vertical scaling offset	Win0_CBR_vscl_offset	Win0_SCL_OFFSET [15:8]
Win0 Cbr horizontal scaling offset	Win0_CBR_hscl_offset	Win0_SCL_OFFSET [7:0]

18.3.5 De-flicker (Interlace vertical filtering)

It is necessary to display a non-interlaced video signal on an interlaced display (such as TV set). Thus some form of "non-interlaced-to-interlaced conversion" may be required.

The easiest approach is to throw away every other active scan line in each non-interlaced frame. Although the cost is minimal, there are problems with this approach. If there is a sharp vertical transition of color or intensity. It will flicker at one-half the refresh rate.

A better solution is to use two lines of non-interlaced data to generate one line of interlace data. Fast vertical transitions are smoothed out over several interlace lines.

The vertical filtering of two non-interlaced lines can be done by enabling the vertical scaling offset dynamic change in different field (even/odd). The dynamic change value of scaling offset is half of the scaling factor. You should enable the scaling down vertical offset in scaling down mode; enable the scaling up vertical offset in scaling up mode, or one of it in no-scaling mode.

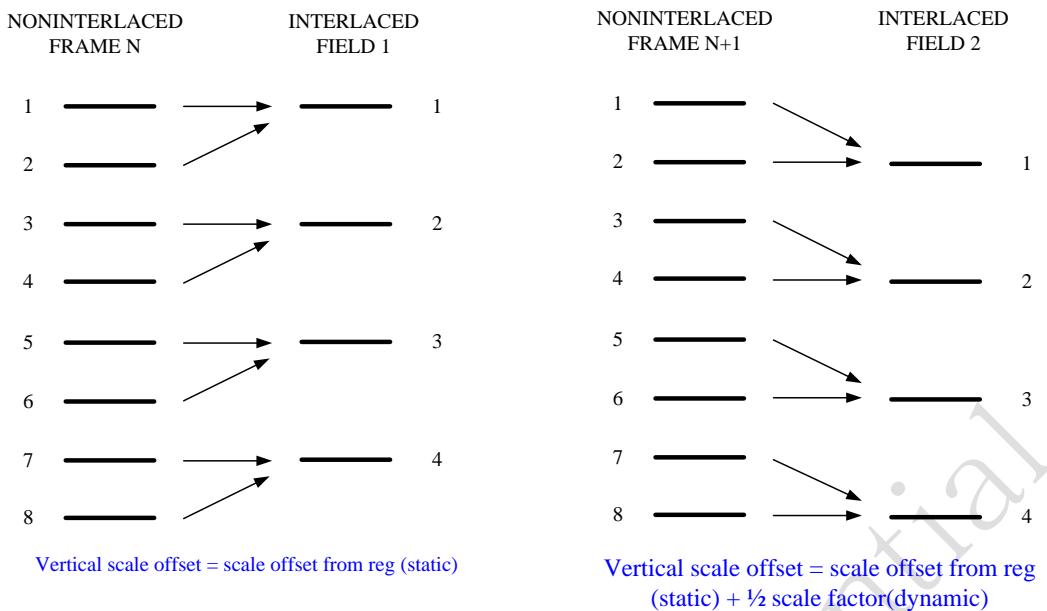


Fig. 18- 9VOP Interlace Vertical Filtering

18.3.6 Overlay

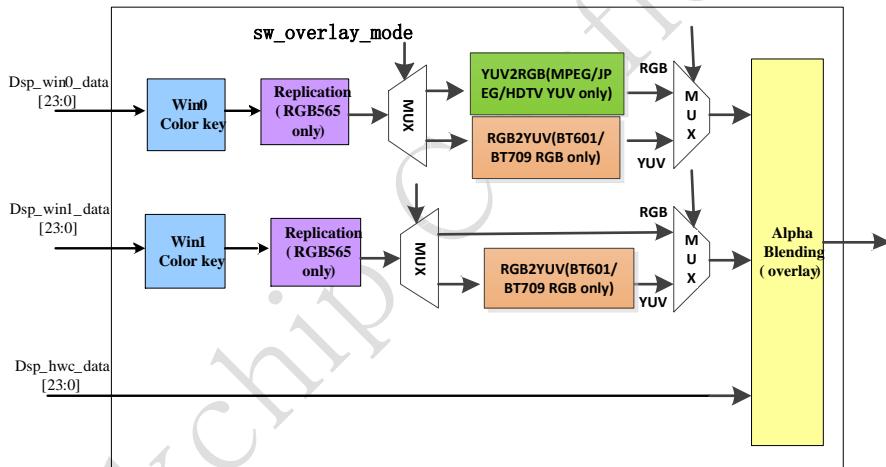


Fig. 18- 10VOP Overlay Block Diagram

1. Overlay display

There are totally 4 layers for overlay display: Background, Win0, Win1 and Hwc.

The background is a programmable solid color layer, which is always the bottom of the display screen.

Hwc is a 32x32 or 64x64 8BPP color palette layer, which is the top layer of the display screen. `dsp_hwc_data[23:0]` come from the HWC_LUT, which input is a 8BPP data format and output is 24 bits `dsp_hwc_data`.

The two middle layers are Win0 and Win1. Win1 is on the top of Win0 in default setting, setting `VOP_DSP_CTRL0[8]` to '1' can let Win0 be on the top of Win1.

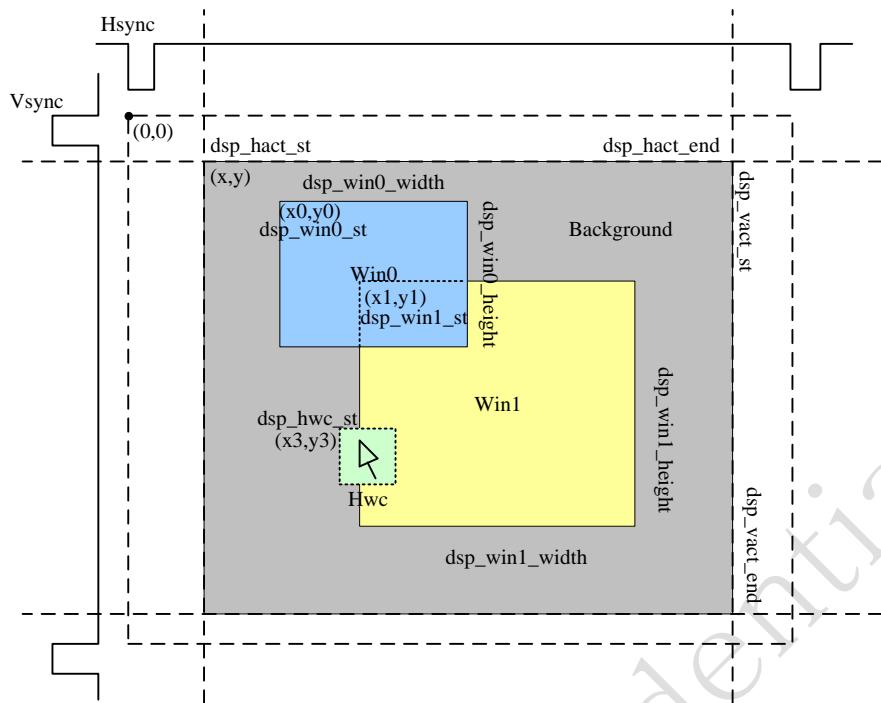


Fig. 18- 11VOP Overlay Display

2. Transparency color key

There are specific registers(VOP_WIN0_COLOR_KEY,VOP_WIN1_COLOR_KEY) for Win0 and Win1 layer to configure the colorkey value. The two transparency color key can be active at the same time.

The pixel color value is compared to the transparency color key before final display. The transparency color key value defines the pixel data considered as the transparent pixel. The pixel values with the source color key value are pixels not visible on the screen, and the under layer pixel values or solid background color are visible.

Transparency color key is done after the scaling module . So transparency color key can only be used in non-scaling mode.

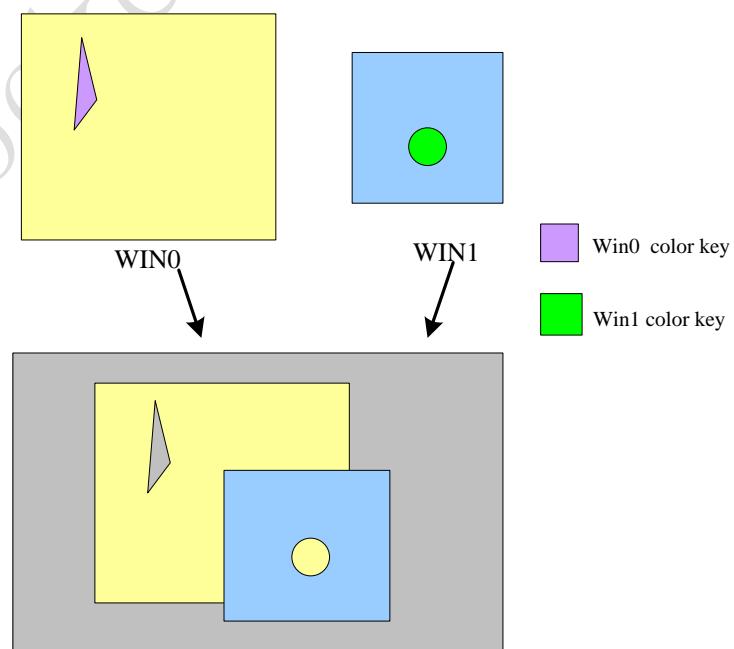


Fig. 18- 12VOP Transparency Color Key

3. Alpha Blending

There are three alpha values for blending between four overlay layers: alpha_win0[7:0], alpha_win1[7:0], alpha_hwc[7:0].

Two blending modes are supported. One is per-pixel (ARGB) mode; the other is user-specified mode. In ARGB mode, the alpha value is in the ARGB data (Win0 and Win1 normal mode only). In user-specified mode, the alpha value comes from the register (VOP_ALPHA_CTRL[27: 4]).

Pre-multiplied alpha are supported for per-pixel alpha in Win0 and Win 1, for Pre-multiplied alpha, the SRC data has already been multiplied with alpha value.

For win1 layer, it supports alpha scale, and it use “win1_aa_pre_mul” to pre_multiply RGB by alpha in hardware if RGB is not pre_multiplied by software. This is used before scaling alpha channel.

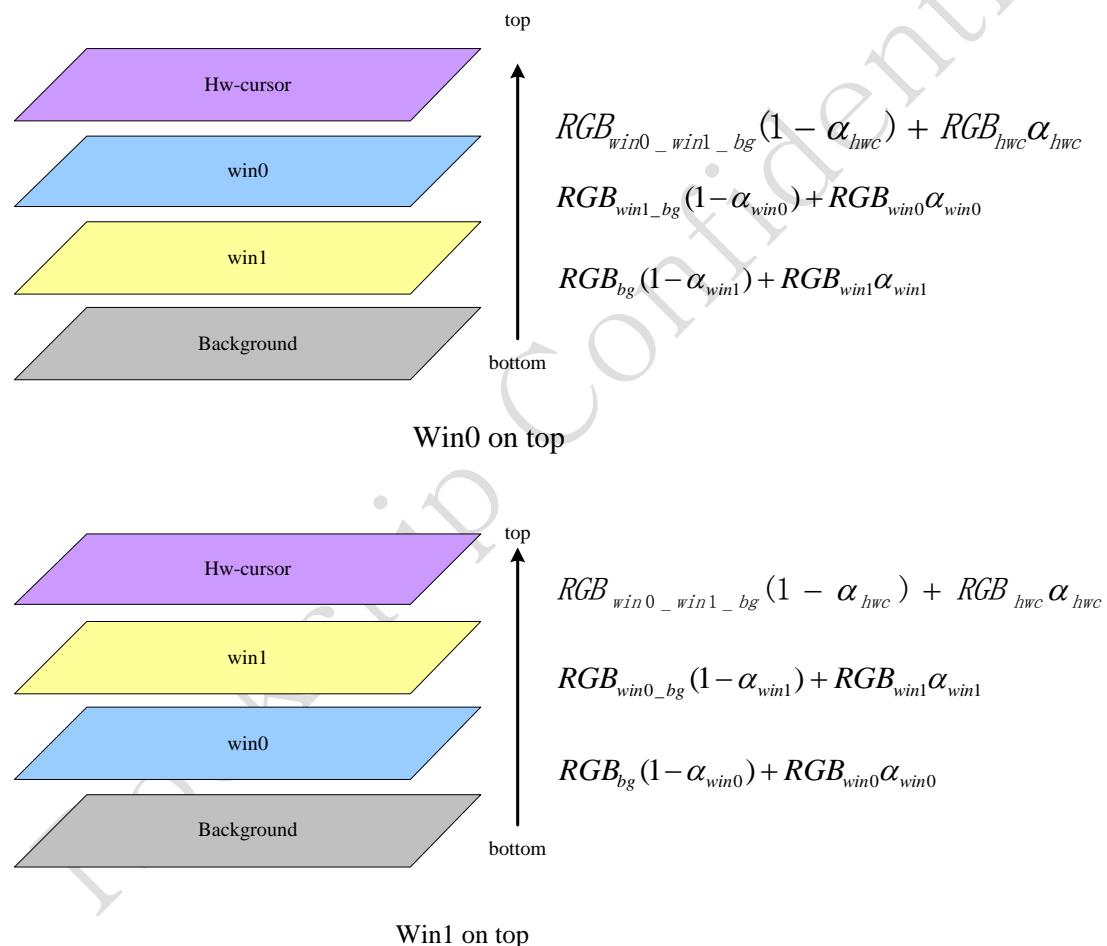


Fig. 18- 13VOP Alpha blending

4 RGB OVERLAY and YCbCr OVERLAY

VOP has a signal named sw_overlay_mode at DSP_CTRL0[31] to decide overlay in RGB or YUV color space.

RGB OVERLAY

All layers overlay in RGB color space. If win0 is YUV420/422/444, it can be converted to RGB888 by YUV2RGB use MPEG,JPEG and HDTV formula.

YCbCr OVERLAY

All layers overlay in YCbCr color space. It is designed for HDMI/TVE output. When win0/1 input picture format is RGB, it can be converted to YUV444 using BT601 or BT709 formula.

Since HWC supports BPP8, we can directly use AYUV444 LUT instead of ARGB888 LUT.

The output format of overlay module is YUV. If BCSH is enable, RGB2YUV shoule be disabled, because BCSH works at YCbCr color space. If TVE is enable, YUV2RGB should be disabled in order to make sure thar the input format of TVE is YUV444 and sw_uv_offset_en should be enabled.

18.3.7 Scaler

Scaler is used to support dual panel display for HDMI + PAD.

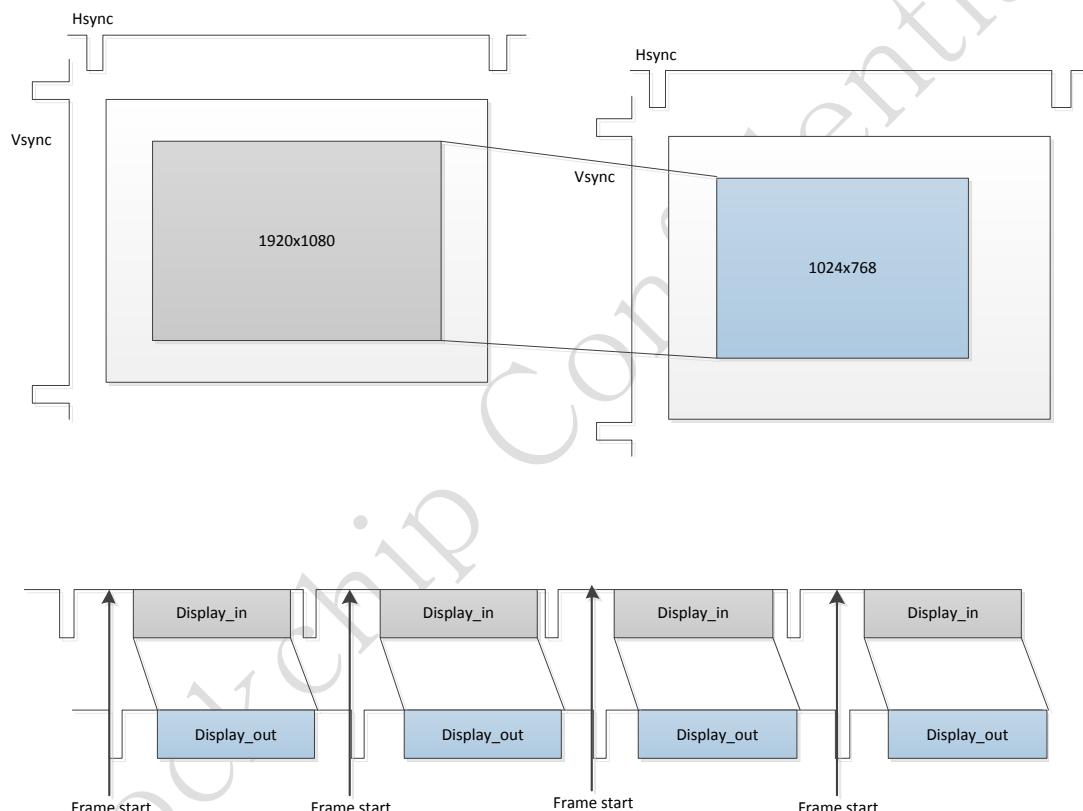


Fig. 18- 14Scaler Frame Sync

The horizontal and vertical scale ratio ofscaler is 0.25~4.

It allows border(black display on the pannel edge) for PAD output,as the figure showed below.

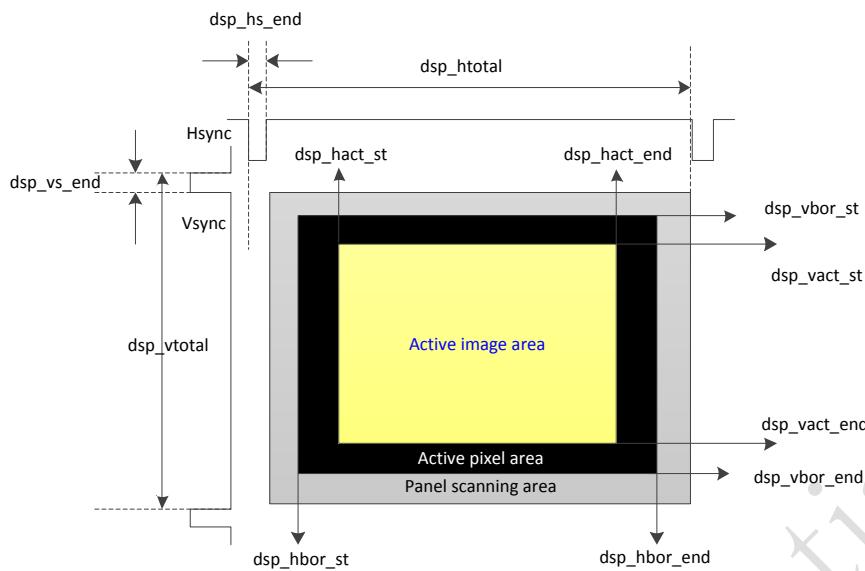


Fig. 18- 15Scaler Output Timing setting

18.3.8 Gamma Correction

Gamma Correction is necessary because most monitors don't have a linear relationship between the voltage and the brightness, which results in your scene looking like it has too much contrast and the light falling off from the source outward, happens too quickly. The result can also be problematic if you are going into a composition program.

You can correct this by "Gamma Correction", which allows you to display the images and textures on your computer in an accurate manner.

Your screen is not linear, in that it displays the brightness unevenly. As a result, the image looks to be more high contrast than it should, you end up adding more lights or turning up the intensity, or you don't use the lighting in a realistic way that matches well with live action scenes. It also creates problems for you if you use compositing software.

It consumes 256x8bit LUT for each channel. You can write gamma correction LUT through register "DSP_LUT_ADDR" one by one after set `dsp_lut_en = 0`.

18.3.9 Replication and dither down

1 Replication(dither up)

If the interface data bus is wider than the pixel format size, by programming the pixel components replication active/inactive, the MSB is replicated to the LSB of the interface data bus or the LSB is filled with 0s.

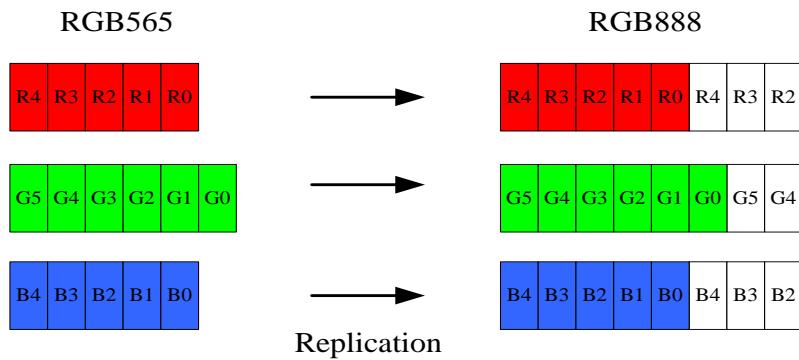


Fig. 18- 16VOP Replication

Dithering is used to improve the quality of display the pixel data in a lower color depth on the LCD panel. The Dithering algorithm is based on the (x,y) pixel position, the value of removed bits and frame counter.

2 Dither down

Here we support three kinds of dithering arithmetic. RGB888 to RGB666 has two ways, one is allegro, the other is FRC. RGB888 to RGB565 has only one arithmetic, which is allegro.

When Dithering is not enabled, the MSBs of the pixel color components are output on the interface data bus if the interface data bus is smaller than the pixel format size.

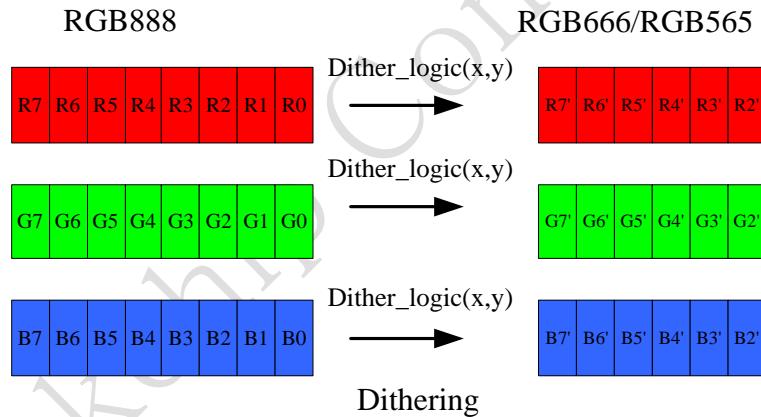


Fig. 18- 17vop dithering

The pattern of dither frc was configured by vopregfile vop_base+0x0e0~0x0f4.

The following figure is the recommended pattern picture in vop, you can config different value of reg 0x1e0~0x1f4, to change the pattern picture.(default value is all 0s here, can give recommended values when initial)

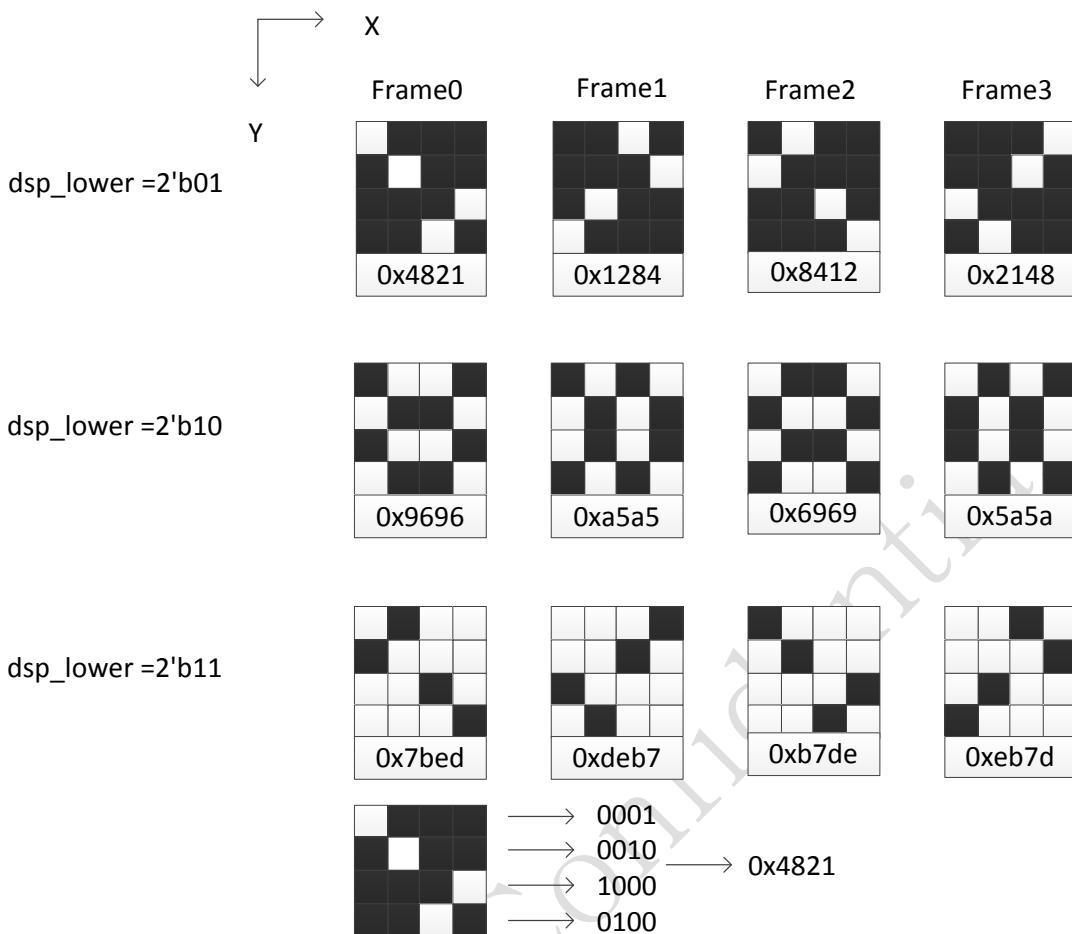


Fig. 18- 18frc pattern diagram

Recommended pattern:

0x0e0 : 0x12844821

0x0e4 : 0x21488412

0x0e8 : 0x55aaaa55

0x0ec : 0x55aaaa55

0x0f0 : 0xdeb77bed

0x0f4 : 0xed7bb7de

18.3.10 TV ENCODE

The TV ENCODE core feature :

- Single 10-bit output to on-chip VDAC
- PAL/NTSC encoding
- Programmable luma filter coefficient
- Programmable luma/chroma delay
- Programmable brightness/contrast

There is a special register to config In TV mode.

tve_mode -> dsp_ctrl0[25]

0 : NTSC mode

1 : PAL mode

Other registers reference to resigister description.

There list NTSC and PAL typical register config as follow:

```
ntsc_config[][][2] ={  
    {0x00 , 0x000a0000} ,  
    {0x04 , 0x00C07a81} ,  
    {0x08 , 0x169800FC} ,  
    {0x0C , 0x96B40000} ,  
    {0x10 , 0x21F07BD7} ,  
    {0x14 , 0x02ff0000} ,  
    {0x18 , 0xF40202fd} ,  
    {0x1C , 0xF332d919} ,  
    {0x34 , 0x001500D6} ,  
    {0x38 , 0x0100888C} ,  
    {0x3C , 0x00000000} ,  
    {0x50 , 0x00000000} ,  
    {0x68 , 0x00000000} ,  
    {0x78 , 0x0052543C} ,  
    {0x8C , 0x00000002} ,  
    {0x90 , 0x00008300}};
```

```
pal_config[][][2] ={  
    {0x00 , 0x010a0000} ,  
    {0x04 , 0x00C28381} ,  
    {0x08 , 0x2694011D} ,  
    {0x0C , 0xB6C00880} ,  
    {0x10 , 0x2A098ACB} ,  
    {0x14 , 0x02ff0000} ,  
    {0x18 , 0xF40202fd} ,  
    {0x1C , 0xF332d919} ,  
    {0x34 , 0x001500F6} ,  
    {0x38 , 0x4100088A} ,  
    {0x3C , 0x00000000} ,
```

```
{0x50 , 0x00000000} ,
{0x68 , 0x00000000} ,
{0x78 , 0x002e553C} ,
{0x8C , 0x00000022} ,
{0x90 , 0x00008900}};
```

When working in TV mode ,the dclk is 27Mhz,the sw_core_clk_sel should enable.

18.3.11 VDAC

If vop working in pal/ntscmode ,Tve_dac_dclk_en should be configured to 1, the vdac clk invert can be configured at the same time.

Tve_dac_dclk_en ->vop_base_addr + 0x20[20];

Tve_dac_dclk_inv->vop_base_addr + 0x20[21];

There is a 10 bit on-chip VDAC for the TVE ENCODE.

You should configure the VDAC digital signals. GRF_TVE_CON is the VDAC global register, the high 16bit is the write enable signals map to low 16bits.

VDAC GRF signals as follow:

GRF register	function name	function
GRF_TVE_CON[0]	enextref	Enable control pin to allow internally generated bandgap to be probed in the analog pin 'vbg'. Under normal operation and envbg=1 1'b1: Internal bandgap is passed to vbg pin; 1'b0: vbg pins is at high impedance.
GRF_TVE_CON[1]	envbg	Enable control pin to power up internal bandgap. 1'b1: Internal bandgap is ON; 1'b0: Internal bandgap is OFF.
GRF_TVE_CON[2]	endac	Enable control pin to power up the VDAC
GRF_TVE_CON[3]	ensc	Sense comparator enable for cable connection detection of DAC
GRF_TVE_CON[6:4]	enctr[2:0]	Enable control pins for analog biasing Test. In normal mode set

		enctr[2:0]=3'b000
GRF_TVE_CON[12:7]	dacgc[5:0]	Gain setting digital input word for VDAC

The detail description of VDAC reference to VDAC chapter.

18.4 Register description

18.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
VOP_SYS_CTRL	0x0000	W	0x00000000	System control register
VOP_DSP_CTRL0	0x0004	W	0x00000000	Display control register0
VOP_DSP_CTRL1	0x0008	W	0x01000000	Display control register1
VOP_INT_SCALER	0x000c	W	0x00000000	interrupt register about scaler
VOP_INT_STATUS	0x0010	W	0x00000000	Interrupt status register
VOP_ALPHA_CTRL	0x0014	W	0x00000000	Blending control register
VOP_WIN0_COLOR_KEY	0x0018	W	0x00000000	Win0 color key register
VOP_WIN1_COLOR_KEY	0x001c	W	0x00000000	Win1 color key register
VOP_WIN0_YRGB_MST0	0x0020	W	0x00000000	Win0 YRGB memory start address 0
VOP_WIN0_CBR_MST0	0x0024	W	0x00000000	Win0 Cbr memory start address 0
VOP_WIN1_VIR	0x0028	W	0x00000000	Win1 virtual stride
VOP_AXI_BUS_CTRL	0x002c	W	0x15500000	Axi Bus interface control register
VOP_WIN0_VIR	0x0030	W	0x01400140	Win0 virtual stride
VOP_WIN0_ACT_INFO	0x0034	W	0x00ef013f	Win0 active window width/height
VOP_WIN0_DSP_INFO	0x0038	W	0x00ef013f	Win0 display width/height on panel

VOP_WIN0_DSP_ST	0x003c	W	0x000a000a	Win0 display start point on panel
VOP_WIN0_SCL_FACTOR_YRGB	0x0040	W	0x10001000	Win0 YRGB scaling factor
VOP_WIN0_SCL_FACTOR_CBR	0x0044	W	0x10001000	Win0 CBR scaling factor
VOP_WIN0_SCL_OFFSET	0x0048	W	0x00000000	Win0 scaling start point offset
VOP_WIN1_MST	0x004c	W	0x00000000	Win1 memory start address
VOP_WIN1_DSP_INFO	0x0050	W	0x00ef013f	Win1 display width/height on panel
VOP_WIN1_DSP_ST	0x0054	W	0x000a000a	Win1 display start point on panel
VOP_HWC_MST	0x0058	W	0x00000000	Hwc memory start address
VOP_HWC_DSP_ST	0x005c	W	0x000a000a	Hwc display start point on panel
VOP_DSP_HTOTAL_HS_END	0x006c	W	0x014a000a	Panel scanning horizontal width and hsync pulse end point
VOP_DSP_HACT_ST_END	0x0070	W	0x000a014a	Panel active horizontal scanning start point and end point
VOP_DSP_VTOTAL_VS_END	0x0074	W	0x00fa000a	Panel scanning vertical height and vsync pulse end point
VOP_DSP_VACT_ST_END	0x0078	W	0x000a00fa	Panel active vertical scanning start point and end point
VOP_DSP_VS_ST_END_F1	0x007c	W	0x00000000	Vertical scanning start point and vsync pulse end point of even field in interlace mode
VOP_DSP_VACT_ST_END_F1	0x0080	W	0x00000000	Vertical scanning active start point and end point of even field in interlace mode
VOP_GATHER_TRANSFER	0x0084	W	0x00008220	AXI read transfer gather setting
VOP_REG_CFG_DONE	0x0090	W	0x00000000	Register config done flag

VOP_VERSION	0x0094	W	0x00000000	version for vop
VOP_SCALER_CTRL	0x00a0	W	0x00000000	scaler ctrl register
VOP_SCALER_FACTOR	0x00a4	W	0x00000000	scaler module scaling factor
VOP_SCALER_FRAME_ST	0x00a8	W	0x00000000	scaler output scanning start setting register
VOP_SCALER_DSP_H_OR_TIMING	0x00ac	W	0x00000000	scaler output scanning horizontal timing
VOP_SCALER_DSP_H_ACT_TIMING	0x00b0	W	0x00000000	scaler output horizontal active window
VOP_SCALER_DSP_V_ER_TIMING	0x00b4	W	0x00000000	scaler output scanning vertical timing
VOP_SCALER_DSP_V_ACT_TIMING	0x00b8	W	0x00000000	scaler output vertical active window
VOP_SCALER_DSP_H_BOR_TIMING	0x00bc	W	0x00000000	scaler output horizontal border window
VOP_SCALER_DSP_V_BOR_TIMING	0x00c0	W	0x00000000	scaler output vertical border window
VOP_BCSH_CTRL	0x00d0	W	0x00000000	Brightness/Contrast enhancement/Saturation/Hue contrl
VOP_BCSH_COL_BAR	0x00d4	W	0x00000000	Colorbar YUV value
VOP_BCSH_BCS	0x00d8	W	0x00000000	Brightness/Contrast enhancement/Saturation
VOP_BCSH_H	0x00dc	W	0x00000000	Hue
VOP_FRC_LOWER01_0	0x00e0	W	0x00000000	Frc algorithm configuration register 1
VOP_FRC_LOWER01_1	0x00e4	W	0x00000000	Frc algorithm configuration register 2
VOP_FRC_LOWER10_0	0x00e8	W	0x00000000	Frc algorithm configuration register 3
VOP_FRC_LOWER10_1	0x00ec	W	0x00000000	Frc algorithm configuration register 4
VOP_FRC_LOWER11	0x00f0	W	0x00000000	Frc algorithm configuration

_0				register 5
VOP_FRC_LOWER11_1	0x00f4	W	0xed7bb7de	Frc algorithm configuration register 6
VOP_TV_CTRL	0x0200	W	0x00000000	tv mode control register
VOP_TV_SYNC_TIMING	0x0204	W	0x00000000	sync timing ctrl register
VOP_TV_ACT_TIMING	0x0208	W	0x00000000	act timing ctrl register
VOP_TV_ADJ_TIMING	0x020c	W	0x00000000	adjust timing register
VOP_TV_FRQ_SC	0x0210	W	0x00000000	Sub-carrier Frequency
VOP_TV_FILTER0	0x0214	W	0x00000000	Filter 0
VOP_TV_FILTER1	0x0218	W	0x00000000	Filter 1
VOP_TV_FILTER2	0x021c	W	0x00000000	Filter 2
VOP_TV_ACT_ST	0x0234	W	0x00000000	picture start offset
VOP_TV_ROUTING	0x0238	W	0x00000000	Routing
VOP_TV_SYNC_ADJUST	0x0250	W	0x00000000	Sync Adjust
VOP_TV_STATUS	0x0254	W	0x00000000	TV Status register
VOP_TV_RST	0x0268	W	0x00000000	tv reset Control
VOP_TV_SATURATION	0x0278	W	0x00000000	Colour Burst and Saturation
VOP_TV_BANDWIDTH_CTRL	0x028c	W	0x00000000	Chroma bandwidth
VOP_TV_BRIGHTNESS_CONTRAST	0x0290	W	0x00000000	Brightness and Contrast
VOP_MMU_DTE_ADDR	0x0300	W	0x00000000	MMU current page Table address
VOP_MMU_STATUS	0x0304	W	0x00000000	MMU status register
VOP_MMU_COMMAND	0x0308	W	0x00000000	MMU command register
VOP_MMU_PAGE_FAULT_ADDR	0x030c	W	0x00000000	MMU logical address of last page fault

VOP_MMU_ZAP_ONE_LINE	0x0310	W	0x00000000	MMU Zap cache line register
VOP_MMU_INT_RAW_STAT	0x0314	W	0x00000000	MMU raw interrupt status register
VOP_MMU_INT_CLEA_R	0x0318	W	0x00000000	MMU raw interrupt status register
VOP_MMU_INT_MAS_K	0x031c	W	0x00000000	MMU raw interrupt status register
VOP_MMU_INT_STA_TUS	0x0320	W	0x00000000	MMU raw interrupt status register
VOP_MMU_AUTO_GATING	0x0324	W	0x00000000	mmu auto gating
VOP_HWC_LUT_ADD_R	0x0800	W	0x00000000	Access entry for HWC LUT memory(size is word only)
VOP_DSP_LUT_ADD_R	0x0c00	W	0x00000000	Access entry for DSP LUT memory(size is word only)

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

18.4.2 Detail Register Description

VOP_DSP_CTRL0

Address: Operational Base + offset (0x0004)

Display control register0

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_overlay_mode 1'b0: overlay in rgb domain 1'b1: overlay in yuv domain
30	RW	0x0	alpha_mode_sel1 alpha mode select 1 1'b0 : alpha value no change 1'b1 : alpha = alpha + alpha[7]
29	RW	0x0	alpha_mode_sel0 alpha mode select 0 1'b0 : Non-premultiplied alpha

			1'b1 : Premultiplied alpha
28	RW	0x0	hwc_alpha_mode hwc alpha mode select 1'b0: user-defined alpha 1'b1: per-pixel alpha
27	RW	0x0	dither_down_sel dither down mode select 1'b0: allegro 1'b1: FRC
26	RW	0x0	sw_uv_offset_en 1'b0:uv no offset 1'b1:uv value minus 128
25	RW	0x0	tve_mode tve mode select 1'b0 : NTSC mode 1'b1 : PAL mode
24	RO	0x0	reserved
23	RW	0x0	yuv_clip YCrCb clip 1'b0: disable, YCbCr no clip 1'b1: enable, YCbCr clip before YCbCr2RGB *Y clip: 16~235, CBCR clip: 16~239
22	RO	0x0	reserved
21:20	RW	0x0	win0_csc_mode Win0 YUV2RGB Color space conversion: 2'b00: mpeg 2'b01: jpeg 2'b10: hd 2'b11: reserved reused by win0 r2y color space conversion: 2'bX0: BT601 2'bX1: BT709

19	RW	0x0	win1_alpha_mode Win1 alpha mode 1'b0: user-defined alpha 1'b1: per-pixel alpha
18	RW	0x0	win0_alpha_mode Win0 alpha mode 1'b0: user-defined alpha 1'b1: per-pixel alpha
17	RW	0x0	win0_cbr_deflick Win0 Cbr deflick mode 1'b0: disable 1'b1: enable
16	RW	0x0	win0_yrgb_deflick Win0 YRGB deflick mode 1'b0: disable 1'b1: enable
15	RW	0x0	win1_interlace_read Win1 interlace read mode 1'b0: disable 1'b1: enable
14	RW	0x0	win0_interlace_read Win0 interlace read mode 1'b0: disable 1'b1: enable
13	RW	0x0	interlace_field_pol Interlace field polarity 1'b0: normal 1'b1: invert
12	RW	0x0	dsp_interlace Interlace display enable 1'b0: disable 1'b1: enable *This mode is related to the TVE output, the display timing of odd field must be set

			correctly. (vop_dsp_vs_st_end_f1/vop_dsp_vact_end_f1)
11	RW	0x0	dither_down Dither-down enable 1'b0: disable 1'b1: enable
10	RW	0x0	dither_down_mode Dither-down mode 1'b0: RGB888 to RGB565 1'b1: RGB888 to RGB666
9	RW	0x0	dither_up dither up RGB565 to RGB888 enable 1'b0: disable 1'b1: enable
8	RW	0x0	dsp_win0_top Win0 and Win1 position swap 1'b0: win1 on the top of win0 1'b1: win0 on the top of win1
7	RW	0x0	dsp_dclk_pol DCLK invert enable 1'b0: normal 1'b1: invert
6	RW	0x0	dsp_den_pol DEN polarity 1'b0: positive 1'b1: negative
5	RW	0x0	dsp_vsync_pol VSYNC polarity 1'b0: negative 1'b1: positive
4	RW	0x0	dsp_hsync_pol HSYNC polarity 1'b0: negative

			1'b1: positive
3:0	RW	0x0	<p>dsp_out_mode Display output format 4'b0000: Parallel 24-bit RGB888 output R[7:0],G[7:0],B[7:0] 4'b0001: Parallel 18-bit RGB666 output 6'b0,R[5:0],G[5:0],B[5:0] 4'b0010: Parallel 16-bit RGB565 output 8'b0,R[4:0],G[5:0],B[4:0] Others: Reserved.</p>

VOP_DSP_CTRL1

Address: Operational Base + offset (0x0008)

Display control register1

Bit	Attr	Reset Value	Description
31	RW	0x0	<p>dsp_out_zero DEN、HSYNC、VSYNC output software ctrl 1'b0: normal output 1'b1: all output '0' means:hsync,vsync,den =000</p>
30:29	RO	0x0	reserved
28	RW	0x0	<p>dsp_rg_swap Display output red and green swap enable 1'b0: RGB 1'b1: GRB</p>
27	RW	0x0	<p>dsp_rb_swap Display output red and blue swap enable 1'b0: RGB 1'b1: BGR</p>
26	RW	0x0	<p>dsp_bg_swap Display output blue and green swap enable 1'b0: RGB 1'b1: RBG</p>
25	RW	0x0	dsp_black_en

			Black display mode When this bit enable, the pixel data output is all black (0x000000)
24	RW	0x1	dsp_blank_en Blank display mode When this bit enable, the hsync/vsync/den output is blank. means:hsync,vsync,den =110
23:16	RW	0x00	dsp_bg_red Background Red color Background Red color
15:8	RW	0x00	dsp_bg_green Background Green color Background Green color
7:0	RW	0x00	dsp_bg_blue Background Blue color Background Blue color

VOP_INT_SCALER

Address: Operational Base + offset (0x000c)

interrupt register about scaler

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	hdmi_den_pol DEN polarity 1'b0: positive 1'b1: negative
5	RW	0x0	hdmi_vsync_pol VSYNC polarity 1'b0: negative 1'b1: positive
4	RW	0x0	hdmi_hsync_pol

			Hsync polarity 1'b0: negative 1'b1: positive
3	RW	0x0	fs_addr_mask_en frame start mask bit 1'b0: disable 1'b1: enable
2	RO	0x0	scaler_empty_intr_status scaler empty interrupt status
1	W1C	0x0	scaler_empty_intr_clr scaler empty interrupt clear (Auto clear)
0	RW	0x0	scaler_empty_intr_en scaler empty interrupt enable 1'b0: disable 1'b1: enable

VOP_INT_STATUS

Address: Operational Base + offset (0x0010)

Interrupt status register

Bit	Attr	Reset Value	Description
31	RO	0x0	line_flag_raw_status Line flag raw Interrupt status
30	RO	0x0	fs_raw_status Frame start raw interrupt status
29	RO	0x0	win1_empty_intr_status win1 data empty interrupt status
28	RO	0x0	win0_empty_intr_status

			win0 data empty interrupt status
27	W1C	0x0	win1_empty_intr_clr win1 data empty interrupt clear(auto clear)
26	W1C	0x0	win0_empty_intr_clr win0 data empty interrupt clear(auto clear)
25	RW	0x0	win1_empty_intr_en win1 data empty interrupt enable signal 1'b0:disable 1'b1:enable
24	RW	0x0	win0_empty_intr_en win0 data empty interrupt enable signal 1'b0:disable 1'b1:enable
23:12	RW	0x000	dsp_line_frag_num Line number of the Line flag interrupt The display line number when the flag interrupt occur, the range is (0~DSP_VTOTAL-1).
11	W1C	0x0	bus_error_intr_clr Bus error Interrupt clear (Auto clear)
10	W1C	0x0	line_frag_intr_clr Line flag Interrupt clear (Auto clear)
9	W1C	0x0	fs_intr_clr Frame start interrupt clear (Auto clear)
8	W1C	0x0	dsp_hold_valid_intr_clr display hold valid interrupt clear (Auto clear)
7	RW	0x0	bus_error_intr_en

			Bus error interrupt enable 1'b0: disable 1'b1: enable
6	RW	0x0	line_frag_intr_en Line flag Interrupt enable 1'b0: disable 1'b1: enable
5	RW	0x0	fs_intr_en Frame start interrupt enable 1'b0: disable 1'b1: enable
4	RW	0x0	dsp_hold_valid_intr_en display hold valid interrupt enable 1'b0: disable 1'b1: enable
3	RO	0x0	bus_error_intr Bus error Interrupt status
2	RO	0x0	line_frag_intr Line flag Interrupt status
1	RO	0x0	fs_intr Frame start interrupt status
0	RO	0x0	dsp_hold_valid_intr display hold valid interrupt status

VOP_ALPHA_CTRL

Address: Operational Base + offset (0x0014)

Blending control register

Bit	Attr	Reset Value	Description

31:28	RO	0x0	reserved
27:20	RW	0x00	hwc_alpha_value HWC alpha blending value
19:12	RW	0x00	win1_alpha_value Win1 alpha blending value
11:4	RW	0x00	win0_alpha_value Win0 alpha blending value
3	RO	0x0	reserved
2	RW	0x0	hwc_alpha_en HWC alpha blending enable 1'b0: disable 1'b1: enable
1	RW	0x0	win1_alpha_en Win1 alpha blending enable 1'b0: disable 1'b1: enable
0	RW	0x0	win0_alpha_en Win0 alpha blending enable 1'b0: disable 1'b1: enable

VOP_WIN0_COLOR_KEY

Address: Operational Base + offset (0x0018)

Win0 color key register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	win0_key_en Win0 transparency color key enable

			1'b0: disable 1'b1: enable
23:0	RW	0x000000	win0_key_color Win0 key color

VOP_WIN1_COLOR_KEY

Address: Operational Base + offset (0x001c)

Win1 color key register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	win1_key_en Win1 transparency color key enable 1'b0: disable 1'b1: enable
23:0	RW	0x000000	win1_key_color Win1 key color

VOP_WIN0_YRGB_MST0

Address: Operational Base + offset (0x0020)

Win0 YRGB memory start address 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win0_yrgb0_mst win0 YRGB frame buffer memory start address 0

VOP_WIN0_CBR_MST0

Address: Operational Base + offset (0x0024)

Win0 Cbr memory start address 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win0_cbr0_mst win0 CBR frame buffer memory start address 0

VOP_WIN1_VIR

Address: Operational Base + offset (0x0028)

Win1 virtual stride

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	win1_vir_stride Win1 virtual stride Number of words of Win1 Virtual width ARGB888 : win1_vir_width RGB888 : (win1_vir_width*3/4) + (win1_vir_width%3) RGB565 : ceil(win1_vir_width/2)

VOP_AXI_BUS_CTRL

Address: Operational Base + offset (0x002c)

Axi Bus interface control register

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_io_pad_clk_sel IO pad clk select bit 1'b0: normal dclk out 1'b1: gating dclk out
30	RW	0x0	sw_core_clk_sel 1'b0: select dclk sclk 1'b1: select dclk_divsclk_div
29	RW	0x0	mipi_dclk_inv

			mipi dclk invert select bit 1'b0:mipi dclk inv disable 1'b1:mipi dclk inv enable
28	RW	0x1	mipi_dclk_en mipi dclk enable bit 1'b0:mipi dclk disable 1'b1:mipi dclk enable
27	RW	0x0	lvds_dclk_inv lvds dclk invert select bit 1'b0:lvds dclk inv disable 1'b1:lvds dclk inv enable
26	RW	0x1	lvds_dclk_en lvds dclk enable bit 1'b0:lvds dclk disable 1'b1:lvds dclk enable
25	RW	0x0	rgb_dclk_inv rgb dclk invert select bit 1'b0:rgb dclk inv disable 1'b1:rgb dclk inv enable
24	RW	0x1	rgb_dclk_en rgb dclk enable bit 1'b0:rgb dclk disable 1'b1:rgb dclk enable
23	RW	0x0	hdmi_dclk_inv hdmi dclk invert select bit 1'b0:hdmi dclk inv disable 1'b1:hdmi dclk inv enable
22	RW	0x1	hdmi_dclk_en hdmi dclk enable bit 1'b0:hdmi dclk disable 1'b1:hdmi dclk enable
21	RW	0x0	tve_dac_dclk_inv tve dac dclk invert select bit 1'b0:tve dac dclk inv disable

			1'b1:tve dac dclk inv enable
20	RW	0x1	tve_dac_dclk_en tvedac dclk enable bit 1'b0:tve dac dclk disable 1'b1:tve dac dclk enable
19	RW	0x0	sw_hdmi_clk_i_sel hdmiio clk select bit 1'b0: select dclk to hdmiio 1'b1: select dclk_core to hdmiio
18:17	RO	0x0	reserved
16:12	RW	0x00	sw_axi_max_outstand_num Max number of AXI read outstanding
11	RW	0x0	sw_axi_max_outstand_en AXI read outstanding limited enable bit 1'b0: disable 1'b1: enable
10	RW	0x0	sw_vop_mmu_en Vop MMU enable bit 1'b0: disable 1'b1: enable
9:6	RW	0x0	sw_noc_hurry_threshold Noc hurry threshold
5:4	RW	0x0	sw_noc_hurry_value Noc hurry level
3	RW	0x0	sw_noc_hurry_en Noc hurry mode enable bit 1'b0: disable 1'b1: enable
2:1	RW	0x0	sw_noc_qos_value NocQoS level

0	RW	0x0	<p>sw_noc_qos_en NocQoS mode enable bit 1'b0: disable 1'b1: enable</p>

VOP_WIN0_VIR

Address: Operational Base + offset (0x0030)

Win0 virtual stride

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x0140	<p>win0_cbr_vir_stride Number of words of Win0 cbr Virtual width UV420 : ceil(win0_vir_width/4) UV422 : ceil(win0_vir_width/4) UV444 : ceil(win0_vir_width/2)</p>
15:13	RO	0x0	reserved
12:0	RW	0x0140	<p>win0_vir_stride Win0 Virtual stride Number of words of Win0 Virtual width ARGB888 : win0_vir_width RGB888 : (win0_vir_width*3/4) + (win0_vir_width%3) RGB565 : ceil(win0_vir_width/2) YUV : ceil(win0_vir_width/4)</p>

VOP_WIN0_ACT_INFO

Address: Operational Base + offset (0x0034)

Win0 active window width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved

28:16	RW	0x00ef	win0_act_height Win0 active(original) window height win_act_height = (win0 vertical size -1)
15:13	RO	0x0	reserved
12:0	RW	0x013f	win0_act_width Win0 active(original) window width win_act_width = (win0 horizontal size -1)

VOP_WIN0_DSP_INFO

Address: Operational Base + offset (0x0038)

Win0 display width/height on panel

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x0ef	dsp_win0_height Win0 display window height win0_dsp_height = (win0 vertical size -1)
15:11	RO	0x0	reserved
10:0	RW	0x13f	dsp_win0_width Win0 display window width win0_dsp_width = (win0 horizontal size -1)

VOP_WIN0_DSP_ST

Address: Operational Base + offset (0x003c)

Win0 display start point on panel

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x00a	dsp_win0_yst Win0 vertical start point(y) of the Panel scanning

15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_win0_xst Win0 horizontal start point(x) of the Panel scanning

VOP_WIN0_SCL_FACTOR_YRGB

Address: Operational Base + offset (0x0040)

Win0 YRGB scaling factor

Bit	Attr	Reset Value	Description
31:16	RW	0x1000	win0_vs_factor_yrgb Win0 YRGB vertical scaling factor: $\text{factor} = ((\text{VOP_WIN0_ACT_INFO}[31:16]) / (\text{VOP_WIN0_DSP_INFO}[31:16])) * 2^{12}$
15:0	RW	0x1000	win0_hs_factor_yrgb Win0 YRGB horizontal scaling factor: $\text{factor} = ((\text{VOP_WIN0_ACT_INFO}[15:0]) / (\text{VOP_WIN0_DSP_INFO}[15:0])) * 2^{12}$

VOP_WIN0_SCL_FACTOR_CBR

Address: Operational Base + offset (0x0044)

Win0 CBR scaling factor

Bit	Attr	Reset Value	Description
31:16	RW	0x1000	win0_vs_factor_cbr Win0 CBR vertical scaling factor: YCbCr420: $\text{factor} = ((\text{VOP_WIN0_ACT_INFO}[31:16] / 2) / (\text{VOP_WIN0_DSP_INFO}[31:16])) * 2^{12}$ YCbCr422, YCbCr444: $\text{factor} = ((\text{VOP_WIN0_ACT_INFO}[31:16]) / (\text{VOP_WIN0_DSP_INFO}[31:16])) * 2^{12}$
15:0	RW	0x1000	win0_hs_factor_cbr Win0 CBR horizontal scaling factor:

			YCbCr422,YCbCr420: factor=((VOP_WIN0_ACT_INFO[15:0]/2)/(VOP_WIN0_DSP_INFO[15:0]))*2^12 YCbCr444: factor=((VOP_WIN0_ACT_INFO[15:0])/(VOP_WIN0_DSP_INFO[15:0]))*2^12
--	--	--	---

VOP_WIN0_SCL_OFFSET

Address: Operational Base + offset (0x0048)

Win0 scaling start point offset

Bit	Attr	Reset Value	Description
31:24	RW	0x00	win0_vs_offset_cbr Cbr Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
23:16	RW	0x00	win0_vs_offset_yrgb Y Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
15:8	RW	0x00	win0_hs_offset_cbr Cbr Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99
7:0	RW	0x00	win0_hs_offset_yrgb Y Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99

VOP_WIN1_MST

Address: Operational Base + offset (0x004c)

Win1 memory start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win1_mst Win1 frame buffer memory start address

VOP_WIN1_DSP_INFO

Address: Operational Base + offset (0x0050)

Win1 display width/height on panel

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x0ef	dsp_win1_height Win1 display window height win1_dsp_height = (win1 dsp vertical size -1)
15:11	RO	0x0	reserved
10:0	RW	0x13f	dsp_win1_width Win1 display window width win1_dsp_width = (win1 dsphorizontal size -1)

VOP_WIN1_DSP_ST

Address: Operational Base + offset (0x0054)

Win1 display start point on panel

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x00a	dsp_win1_yst Win1 vertical start point(y) of the Panel scanning
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_win1_xst Win1 horizontal start point(x) of the Panel scanning

VOP_HWC_MST

Address: Operational Base + offset (0x0058)

Hwc memory start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	hwc_mst HWC data memory start address

VOP_HWC_DSP_ST

Address: Operational Base + offset (0x005c)

Hwc display start point on panel

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x00a	dsp_hwc_yst HWC vertical start point(y) of the Panel scanning
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_hwc_xst HWC horizontal start point(x) of the Panel scanning

VOP_DSP_HTOTAL_HS_END

Address: Operational Base + offset (0x006c)

Panel scanning horizontal width and hsync pulse end point

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x14a	dsp_htotal Panel display scanning horizontal period
15:12	RO	0x0	reserved

11:0	RW	0x00a	dsp_hs_end Panel display scanning hsync pulse width
------	----	-------	--

VOP_DSP_HACT_ST_END

Address: Operational Base + offset (0x0070)

Panel active horizontal scanning start point and end point

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x00a	dsp_hact_st Panel display scanning horizontal active start point
15:12	RO	0x0	reserved
11:0	RW	0x14a	dsp_hact_end Panel display scanning horizontal active end point

VOP_DSP_VTOTAL_VS_END

Address: Operational Base + offset (0x0074)

Panel scanning vertical height and vsync pulse end point

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x0fa	dsp_vtotal Panel display scanning vertical period.
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_vs_end Panel display scanning vsync pulse width

VOP_DSP_VACT_ST_END

Address: Operational Base + offset (0x0078)

Panel active vertical scanning start point and end point

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x00a	dsp_vact_st Panel display scanning vertical active start point
15:12	RO	0x0	reserved
11:0	RW	0x0fa	dsp_vact_end Panel display scanning vertical active end point

VOP_DSP_VS_ST_END_F1

Address: Operational Base + offset (0x007c)

Vertical scanning start point and vsync pulse end point of even field in interlace mode

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_vs_st_f1 Panel display scanning vertical vsync start point of 2nd field (interlace display mode)
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_vs_end_f1 Panel display scanning vertical vsync end point of 2nd field(interlace display mode)

VOP_DSP_VACT_ST_END_F1

Address: Operational Base + offset (0x0080)

Vertical scanning active start point and end point of even field in interlace mode

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_vact_st_f1 Panel display scanning vertical active start point of 2nd field (interlace display mode)
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_vact_end_f1 Panel display scanning vertical active end point of 2nd field (interlace display mode)

VOP_GATHER_TRANSFER

Address: Operational Base + offset (0x0084)

AXI read transfer gather setting

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:12	RW	0x8	win1_axi_gather_num
11	RO	0x0	reserved
10:8	RW	0x2	win0_cbr_axi_gather_num
7:4	RW	0x2	win0_yrgb_axi_gather_num
3	RO	0x0	reserved
2	RW	0x0	win1_axi_gather_en win1 dma channel AXI read transfer gather 1'b0: disable 1'b1: enable
1	RW	0x0	win0_cbr_axi_gather_en win0 cb/crdma channel AXI read transfer

			gather 1'b0: disable 1'b1: enable
0	WO	0x0	win0_yrgb_axi_gather_en win0 yrgbdma channel AXI read transfer gather 1'b0: disable 1'b1: enable

VOP_REG_CFG_DONE

Address: Operational Base + offset (0x0090)

Register config done flag

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	reg_load_en vop register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.

VOP_VERSION

Address: Operational Base + offset (0x0094)

version for vop

Bit	Attr	Reset Value	Description
31:24	RO	0x00	major IP major version used for IP structure
23:16	RO	0x00	minor minor version

			big feature change under same structure
15:0	RO	0x0000	build rtl current svn number

VOP_SCALER_CTRL

Address: Operational Base + offset (0x00a0)

scaler ctrl register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	scaler_in_vsync_vst 0~255,scaler input line num after vsync negedge
7:6	RW	0x0	scaler_in_vsync_mode scl_in_vsync_mode 2'b00:scaler input vsync negedge 2'b01:scaler input trigger for output 2'b10:configurable scaler input frame start 2'b11:reserved
5	RW	0x0	scaler_out_sel 1'b0: scaler output disable 1'b1: scaler output enable
4	RW	0x0	scaler_out_zero DEN、HSYNC、VSYNC output software ctrl 1'b0: normal output 1'b1: all output '0' means:hsync,vsync,den =000
3	RW	0x0	scaler_den_inv 1'b0: positive 1'b1: negative
2	RW	0x0	scaler_sync_inv hsync vsync polarity 1'b0: negative 1'b1: positive

1	RO	0x0	reserved
0	RW	0x0	scaler_en scaler enable bit 1'b0: scaler disable 1'b1: scaler enable

VOP_SCALER_FACTOR

Address: Operational Base + offset (0x00a4)

scaler module scaling factor

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	scaler_v_factor (1)scaling up: $(Hin_act-1)/(Hout_act-1)*2^{12};$ (2)scaling down(bilinear): $(Hin_act-1)/(Hout_act-1)*2^{12};$
15:14	RO	0x0	reserved
13:0	RW	0x0000	scaler_h_factor (1)scaling up: $(Hin_act-1)/(Hout_act-1)*2^{12};$ (2)scaling down(bilinear): $(Hin_act-1)/(Hout_act-1)*2^{12};$

VOP_SCALER_FRAME_ST

Address: Operational Base + offset (0x00a8)

scaler output scanning start setting register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	scaler_dsp_frame_vst scalerdsp frame vertical start point

15:12	RO	0x0	reserved
11:0	RW	0x000	scaler_dsp_frame_hst scalerdsp frame horizontal start point

VOP_SCALER_DSP_HOR_TIMING

Address: Operational Base + offset (0x00ac)

scaler output scanning horizontal timing

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	scaler_dsp_hs_end scalerdsphs end
15:12	RO	0x0	reserved
11:0	RW	0x000	scaler_dsp_htotal scalerdsphtotal

VOP_SCALER_DSP_HACT_TIMING

Address: Operational Base + offset (0x00b0)

scaler output horizontal active window

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:16	RW	0x000	scaler_dsp_hact_st scalerdsp horizontal active start point
15:12	RO	0x0	reserved
11:0	RW	0x000	scaler_dsp_hact_end scalerdsp horizontal active end point

VOP_SCALER_DSP_VER_TIMING

Address: Operational Base + offset (0x00b4)

scaler output scanning vertical timing

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	scaler_dsp_vs_end scalerdspvs end
15:12	RO	0x0	reserved
11:0	RW	0x000	scaler_dsp_vtotal scalerdspvtotal

VOP_SCALER_DSP_VACT_TIMING

Address: Operational Base + offset (0x00b8)

scaler output vertical active window

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	scaler_dsp_vact_st scalerdsp vertical active start point
15:12	RO	0x0	reserved
11:0	RW	0x000	scaler_dsp_vact_end scalerdsp vertical active end point

VOP_SCALER_DSP_HBOR_TIMING

Address: Operational Base + offset (0x00bc)

scaler output horizontal border window

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:16	RW	0x000	scaler_dsp_hbor_st scalerdsp horizontal border start point
15:12	RO	0x0	reserved
11:0	RW	0x000	scaler_dsp_hbor_end scalerdsp horizontal border end point

VOP_SCALER_DSP_VBOR_TIMING

Address: Operational Base + offset (0x00c0)

scaler output vertical border window

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	scaler_dsp_vbor_st scalerdsp vertical border start point
15:12	RO	0x0	reserved
11:0	RW	0x000	scaler_dsp_vbor_end scalerdsp vertical border end point

VOP_BCSH_CTRL

Address: Operational Base + offset (0x00d0)

Brightness/Contrast enhancement/Saturation/Hue contrl

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	sw_bcsht_r2y_en 1'b0:bypass 1'b1:enable
6	RW	0x0	sw_bcsht_y2r_en 1'b0:bypass 1'b1:enable
5:4	RW	0x0	sw_bcsht_y2r_csc_mode Color space conversion: 2'b00: mpeg 2'b01: jpeg 2'b10: hd 2'b11: Bypass

3:2	RW	0x0	video_mode 2'b00 : black 2'b01 : blue 2'b10 : color bar 2'b11 : normal video
1	RW	0x0	sw_bcsrh_r2y_csc_mode Color space conversion: 1'b0: BT601 1'b1: BT709
0	RW	0x0	bcsrh_en 1'b0 : bcsrh bypass 1'b1 : bcsrh enable

VOP_BCSH_COL_BAR

Address: Operational Base + offset (0x00d4)

Colorbar YUV value

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	color_bar_v
15:8	RW	0x00	color_bar_u
7:0	RW	0x00	color_bar_y

VOP_BCSH_BCS

Address: Operational Base + offset (0x00d8)

Brightness/Contrast enhancement/Saturation

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

24:16	RW	0x000	sat_con Saturation*Contrast*256 : 0,1.992*1.992
15:8	RW	0x00	contrast Contrast*256 : 0,1.992
7:6	RO	0x0	reserved
5:0	RW	0x00	brightness Brightness : -128,127

VOP_BCSH_H

Address: Operational Base + offset (0x00dc)

Hue

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	cos_hue cos hue value
7:0	RW	0x00	sin_hue sin hue value

VOP_FRC_LOWER01_0

Address: Operational Base + offset (0x00e0)

Frc algorithm configuration register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	lower01_frm1
15:0	RW	0x0000	lower01_frm0

VOP_FRC_LOWER01_1

Address: Operational Base + offset (0x00e4)

Frc algorithm configuration register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	lower01_frm3
15:0	RW	0x0000	lower01_frm2

VOP_FRC_LOWER10_0

Address: Operational Base + offset (0x00e8)

Frc algorithm configuration register 3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	lower10_frm1
15:0	RW	0x0000	lower10_frm0

VOP_FRC_LOWER10_1

Address: Operational Base + offset (0x00ec)

Frc algorithm configuration register 4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	lower10_frm3
15:0	RW	0x0000	lower10_frm2

VOP_FRC_LOWER11_0

Address: Operational Base + offset (0x00f0)

Frc algorithm configuration register 5

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	lower11_frm1
15:0	RW	0x0000	lower11_frm0

VOP_FRC_LOWER11_1

Address: Operational Base + offset (0x00f4)

Frc algorithm configuration register 6

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	lower11_frm3
15:0	RW	0x0000	lower11_frm2

VOP_TV_CTRL

Address: Operational Base + offset (0x0200)

tv mode control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	cvbs_mode 1'b0: Encoding is NTSC 1'b1: Encoding is PAL
23:20	RO	0x0	reserved
19:18	RW	0x0	clk_upstream_en 2'b00: Reserved 2'b01: Upstream enable is 1 * pix_clk (HDTV) 2'b10: Upstream enable is 1/2* pix_clk (SDTV)

			2'b11: Reserved
17:16	RW	0x0	timing_en 2'b00: Reserved 2'b01: Timing enable is 1 * pix_clk (HDTV) 2'b10: Timing enable is 1 * pix_clk (SDTV) 2'b11: Reserved
15:11	RO	0x0	reserved
10:9	RW	0x0	filter_gain 2'b00: Luma Filter gain is 1.0 (default value) 2'b01: Luma Filter gain is 0.5 2'b10: Luma Filter gain is 2.0 2'b11: Reserved
8	RW	0x0	filter_upsample 1'b0: Luma Filter output is sampled at 13.5 MHz 1'b1: Luma Filter output is up-sampled to 27 MHz
7:3	RO	0x0	reserved
2:1	RW	0x0	csc_input_sel 2'b00: RGB input to RGB+YUV444 output 2'b01: YUV422 input to YUV444 output 2'b10: Reserved 2'b11: Bypass (YUV444 input to YUV444 output)
0	RO	0x0	reserved

VOP_TV_SYNC_TIMING

Address: Operational Base + offset (0x0204)

sync timing ctrl register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved

23:16	RW	0x00	h_fp Delay to the end of colour burst from H Sync Start , num of 27Mhz clk cycle
15:8	RW	0x00	h_bp Delay to the start of the colour burst from H Sync Start , num of 27Mhz clk cycle
7:0	RW	0x00	h_pw Width of horizontal sync from H Sync Start , num of 27Mhz clk cycle

VOP_TV_ACT_TIMING

Address: Operational Base + offset (0x0208)

act timing ctrl register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	tv_syncs 4'b0000: Sync generator is stopped 4'b0001: 525/60 interlaced 4'b0010: 625/50 interlaced other : reserved
27:16	RW	0x000	h_act_end Delay to the end of active video from H Sync Start , num of 27Mhz clk cycle
15:13	RO	0x0	reserved
12	RW	0x0	sc_rst_en Subcarrier reset enable 1'b0: Subcarrier phase is not reset regularly 1'b1: Subcarrier phase is reset every four fields (NTSC) or eight fields (PAL)

11:0	RW	0x000	h_act_st Delay to the start of the active video from H Sync Start , num of 27Mhz clk cycle
------	----	-------	--

VOP_TV_ADJ_TIMING

Address: Operational Base + offset (0x020c)

adjust timing register

Bit	Attr	Reset Value	Description
31:28	RW	0x0	luma_delay_1 Programmable delay for Luma relative to Chroma
27:16	RW	0x000	h_total Total number of output pixels per line. $525/60 = 1716$ $625/50 = 1728$
15:8	RO	0x0	reserved
7:0	RW	0x00	sc_phase Phase offset applied if subcarrier phase is reset (1.4 degrees per lsb)

VOP_TV_FRQ_SC

Address: Operational Base + offset (0x0210)

Sub-carrier Frequency

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dds_incr colour subcarrier counter . Recommended settings (for 27.00 MHz) are: NTSC: 0x21F07BD7 PAL : 0x2A098ACB

VOP_TV_FILTER0

Address: Operational Base + offset (0x0214)

Filter 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	coeff3 filter coefficients 3
23:16	RW	0x00	coeff2 filter coefficients 2
15:8	RW	0x00	coeff1 filter coefficients 1
7:0	RW	0x00	coeff0 filter coefficients 0

VOP_TV_FILTER1

Address: Operational Base + offset (0x0218)

Filter 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	coeff7 filter coefficients 7
23:16	RW	0x00	coeff6 filter coefficients 6
15:8	RW	0x00	coeff5 filter coefficients 5
7:0	RW	0x00	coeff4 filter coefficients 4

VOP_TV_FILTER2

Address: Operational Base + offset (0x021c)

Filter 2

Bit	Attr	Reset Value	Description
31:24	RW	0x00	coeff11 filter coefficients 11
23:16	RW	0x00	coeff10 filter coefficients 10
15:8	RW	0x00	coeff9 filter coefficients 9
7:0	RW	0x00	coeff8 filter coefficients 8

VOP_TV_ACT_ST

Address: Operational Base + offset (0x0234)

picture start offset

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x000	v_offset Vertical picture start offset (independent of picture blanking)
15:12	RO	0x0	reserved
11:0	RW	0x000	h_offset Horizontal picture start offset (independent of picture blanking)

VOP_TV_ROUTING

Address: Operational Base + offset (0x0238)

Routing

Bit	Attr	Reset Value	Description
31	RW	0x0	RB_swap

			1'b0: No swap 1'b1: Swap Blue/Pb and Red/Pr channels on DACs, relative to the routing shown in 30:28																																			
30:28	RW	0x0	<p>dac_fmt</p> <table> <thead> <tr> <th></th> <th>DAC3</th> <th>DAC2</th> <th>DAC1</th> <th>DAC0</th> </tr> </thead> <tbody> <tr> <td>3'b000:</td> <td>CVBS</td> <td>Blue</td> <td>Green</td> <td>Red</td> </tr> <tr> <td>3'b001:</td> <td>CVBS1</td> <td>CVBS2</td> <td>Y</td> <td>C</td> </tr> <tr> <td>3'b011:</td> <td>Y</td> <td>CVBS2</td> <td>CVBS1</td> <td>C</td> </tr> <tr> <td>3'b100:</td> <td>CVBS</td> <td>Pb</td> <td>Y</td> <td>Pr</td> </tr> <tr> <td>3'b101:</td> <td>C</td> <td>Pb</td> <td>Y</td> <td>Pr</td> </tr> </tbody> </table>		DAC3	DAC2	DAC1	DAC0	3'b000:	CVBS	Blue	Green	Red	3'b001:	CVBS1	CVBS2	Y	C	3'b011:	Y	CVBS2	CVBS1	C	3'b100:	CVBS	Pb	Y	Pr	3'b101:	C	Pb	Y	Pr					
	DAC3	DAC2	DAC1	DAC0																																		
3'b000:	CVBS	Blue	Green	Red																																		
3'b001:	CVBS1	CVBS2	Y	C																																		
3'b011:	Y	CVBS2	CVBS1	C																																		
3'b100:	CVBS	Pb	Y	Pr																																		
3'b101:	C	Pb	Y	Pr																																		
27:24	RW	0x0	<p>sense_on</p> <p>When ON, outputs a steady mid level on the selected DAC(s) to allow load sensing via DAC or external comparators (application dependent). May be applied singly or together</p> <table> <thead> <tr> <th></th> <th>DAC3</th> <th>DAC2</th> <th>DAC1</th> <th>DAC0</th> </tr> </thead> <tbody> <tr> <td>4'b0000:</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> </tr> <tr> <td>4'b0001:</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>ON</td> </tr> <tr> <td>4'b0010:</td> <td>OFF</td> <td>OFF</td> <td>ON</td> <td>OFF</td> </tr> <tr> <td>4'b0100:</td> <td>OFF</td> <td>ON</td> <td>OFF</td> <td>OFF</td> </tr> <tr> <td>4'b1000:</td> <td>ON</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> </tr> <tr> <td>4'b1111:</td> <td>ON</td> <td>ON</td> <td>ON</td> <td>ON</td> </tr> </tbody> </table>		DAC3	DAC2	DAC1	DAC0	4'b0000:	OFF	OFF	OFF	OFF	4'b0001:	OFF	OFF	OFF	ON	4'b0010:	OFF	OFF	ON	OFF	4'b0100:	OFF	ON	OFF	OFF	4'b1000:	ON	OFF	OFF	OFF	4'b1111:	ON	ON	ON	ON
	DAC3	DAC2	DAC1	DAC0																																		
4'b0000:	OFF	OFF	OFF	OFF																																		
4'b0001:	OFF	OFF	OFF	ON																																		
4'b0010:	OFF	OFF	ON	OFF																																		
4'b0100:	OFF	ON	OFF	OFF																																		
4'b1000:	ON	OFF	OFF	OFF																																		
4'b1111:	ON	ON	ON	ON																																		
23:20	RO	0x0	reserved																																			
19:16	RW	0x0	<p>setup_on</p> <p>When ON, adds 7.5IRE setup to the specified channel</p> <p>Luma Red/Pr Green Blue/Pb</p> <table> <thead> <tr> <th></th> <th>4'b0000:</th> <th>OFF</th> <th>OFF</th> <th>OFF</th> <th>OFF</th> </tr> </thead> <tbody> <tr> <td>4'bXXX1:</td> <td></td> <td></td> <td></td> <td></td> <td>ON</td> </tr> <tr> <td>4'bXX1X:</td> <td></td> <td></td> <td></td> <td></td> <td>ON</td> </tr> <tr> <td>4'bX1XX:</td> <td></td> <td></td> <td></td> <td></td> <td>ON</td> </tr> <tr> <td>4'b1XXX:</td> <td></td> <td></td> <td></td> <td></td> <td>ON</td> </tr> </tbody> </table>		4'b0000:	OFF	OFF	OFF	OFF	4'bXXX1:					ON	4'bXX1X:					ON	4'bX1XX:					ON	4'b1XXX:					ON					
	4'b0000:	OFF	OFF	OFF	OFF																																	
4'bXXX1:					ON																																	
4'bXX1X:					ON																																	
4'bX1XX:					ON																																	
4'b1XXX:					ON																																	
15:12	RW	0x0	<p>AGC_pluse_on</p> <p>When ON, adds AGC and EOF pulses to the</p>																																			

			<p>specified channel.</p> <p>N.B. This field should not be used for disabling macrovision functionality.</p> <p>Y/C Red/Pr Green Blue/Pb</p> <p>4'b0000: OFF OFFOFFOFF</p> <p>4'bXXX1: ON</p> <p>4'bXX1X: ON</p> <p>4'bX1XX: ON</p> <p>4'b1XXX: ON</p>
11:8	RW	0x0	<p>video_on</p> <p>When ON, adds appropriate video format to the specified channel</p> <p>Y/C Red/Pr Green Blue/Pb</p> <p>4'b0000 : OFF OFFOFFOFF</p> <p>4'bXXX1: ON</p> <p>4'bXX1X: ON</p> <p>4'bX1XX: ON</p> <p>4'b1XXX: ON</p>
7:4	RW	0x0	<p>sync_on</p> <p>When ON, adds composite sync to the specified channel</p> <p>N.B. Macrovision pseudo-sync pulses (if enabled) will be added to each channel that has sync enabled.</p> <p>Luma Red/Pr Green Blue/Pb</p> <p>4'b0000: OFF OFFOFFOFF</p> <p>4'bXXX1: ON</p> <p>4'bXX1X: ON</p> <p>4'bX1XX: ON</p> <p>4'b1XXX: ON</p>
3	RW	0x0	<p>YPP</p> <p>1'b0: Component output is RGB</p> <p>1'b1: Component output is YPbPr</p>
2	RW	0x0	<p>chroma_off</p> <p>1'b0: Chroma is switched ON (normal colour display)</p> <p>1'b1: Chroma is switched OFF (monochrome)</p>

			display)
1	RW	0x0	Picture_Sync_amplitudes1 1'b0 : Composite has picture/sync ratio of 714/286 (NTSC) 1'b1 : Composite has picture/sync ratio of 700/300 (PAL)
0	RO	0x0	reserved

VOP_TV_SYNC_ADJUST

Address: Operational Base + offset (0x0250)

Sync Adjust

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RW	0x0	line_adj Signed number of lines by which to change frame length. Affects every frame whilst a non-zero value.
7:0	RW	0x00	pix_adj Signed number of (27MHz) pixels by which to adjust frame length. Affects every frame whilst a non-zero value.

VOP_TV_STATUS

Address: Operational Base + offset (0x0254)

TV Status register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	pixel_manager_vstt Pixel Manager Vertical Start State 1'b0: Vertical Start not active 1'b1: Vertical Start active
6:4	RO	0x0	pixel_manager_sta

			Pixel Manager Flow Controller State 3'b000: Idle 3'b001: Seeking 3'b010: Ready 3'b011: First pixel 3'b100: Active 3'b111: Other
3:2	RO	0x0	reserved
1	RO	0x0	sync_gen_VBI Sync Generator Vertical Blanking Interval 1'b0: In VBI 1'b1: Not in VBI
0	RO	0x0	sync_gen_FID Sync Generator Field Identity 1'b0: First Field 1'b1: Second Filed

VOP_TV_RST

Address: Operational Base + offset (0x0268)

tv reset Control

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	disp_RST Software reset for all other TVE functions: 1'b0: Normal operation. 1'b1: Software reset (please do NOT reset the TVE every field or frame).
0	RO	0x0	reserved

VOP_TV_SATURATION

Address: Operational Base + offset (0x0278)

Colour Burst and Saturation

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	burst_size Colour burst amplitude
15:8	RW	0x00	V_weight Conversion factor for Cr to V
7:0	RW	0x00	U_weight Conversion factor for Cb to U

VOP_TV_BANDWIDTH_CTRL

Address: Operational Base + offset (0x028c)

Chroma bandwidth

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:4	RW	0x0	chroma_bandwidth 2'b00: Chroma bandpass filter is bypassed 2'b01: Chroma bandpass filter is centred on 3.58 MHz 2'b10: Chroma bandpass filter is centred on 4.43 MHz
3:0	RW	0x0	cdiff_bandwidth 4'b0000: Colour difference filters OFF (no colour) 4'b0001: Colour difference bandwidth is 0.6 MHz 4'b0010: Colour difference bandwidth is 1.3 MHz 4'b0011: Colour difference bandwidth is 2.0 MHz

VOP_TV_BRIGHTNESS_CONTRAST

Address: Operational Base + offset (0x0290)

Brightness and Contrast

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	Contrast This value determines the gain of the luma channel. It does not affect the chroma gain.
7:0	RW	0x00	Brightness This value determines the black level on the luma channel during active video only. The setup (if applicable) is applied in addition to this value.

VOP_MMU_DTE_ADDR

Address: Operational Base + offset (0x0300)

MMU current page Table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MMU_DTE_ADDR

VOP_MMU_STATUS

Address: Operational Base + offset (0x0304)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGE_FAULT_BUS_ID Index of master responsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE The direction of access for last page fault: 0 = Read 1 = Write
4	RO	0x0	REPLAY_BUFFER_EMPTY The MMU replay buffer is empty

3	RO	0x0	MMU_IDLE The MMU is idle when accesses are being translated and there are no unfinished translated accesses.
2	RO	0x0	STAIL_ACTIVE MMU stall mode currently enabled. The mode is enabled by command
1	RO	0x0	PAGE_FAULT_ACTIVE MMU page fault mode currently enabled . The mode is enabled by command.
0	RO	0x0	PAGING_ENABLED Paging is enabled

VOP_MMU_COMMAND

Address: Operational Base + offset (0x0308)

MMU command register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	WO	0x0	MMU_CMD MMU_CMD. This can be: 3'b000: MMU_ENABLE_PAGING 3'b001: MMU_DISABLE_PAGING 3'b010: MMU_ENABLE_STALL 3'b011: MMU_DISABLE_STALL 3'b100: MMU_ZAP_CACHE 3'b101: MMU_PAGE_FAULT_DONE 3'b110: MMU_FORCE_RESET

VOP_MMU_PAGE_FAULT_ADDR

Address: Operational Base + offset (0x030c)

MMU logical address of last page fault

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	PAGE_FAULT_ADDR address of last page fault

VOP_MMU_ZAP_ONE_LINE

Address: Operational Base + offset (0x0310)

MMU Zap cache line register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE address to be invalidated from the page table cache

VOP_MMU_INT_RAWSTAT

Address: Operational Base + offset (0x0314)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error
0	RW	0x0	PAGE_FAULT page fault

VOP_MMU_INT_CLEAR

Address: Operational Base + offset (0x0318)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	WO	0x0	READ_BUS_ERROR read bus error

0	WO	0x0	PAGE_FAULT page fault
---	----	-----	--------------------------

VOP_MMU_INT_MASK

Address: Operational Base + offset (0x031c)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error
0	RW	0x0	PAGE_FAULT page fault

VOP_MMU_INT_STATUS

Address: Operational Base + offset (0x0320)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR read bus error
0	RO	0x0	PAGE_FAULT page fault

VOP_MMU_AUTO_GATING

Address: Operational Base + offset (0x0324)

mmu auto gating

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

0	RW	0x0	mmu_auto_gating mmu auto gating when it is 1'b1, the mmu will auto gating it self
---	----	-----	---

VOP_HWC_LUT_ADDR

Address: Operational Base + offset (0x0800)

Access entry for HWC LUT memory(size is word only)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	dsp_lut_addr Access entry for DSP LUT memory(size is word only)

VOP_DSP_LUT_ADDR

Address: Operational Base + offset (0x0c00)

Access entry for DSP LUT memory(size is word only)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	dsp_lut_addr Access entry for DSP LUT memory(size is word only)

18.5 Timing Diagram

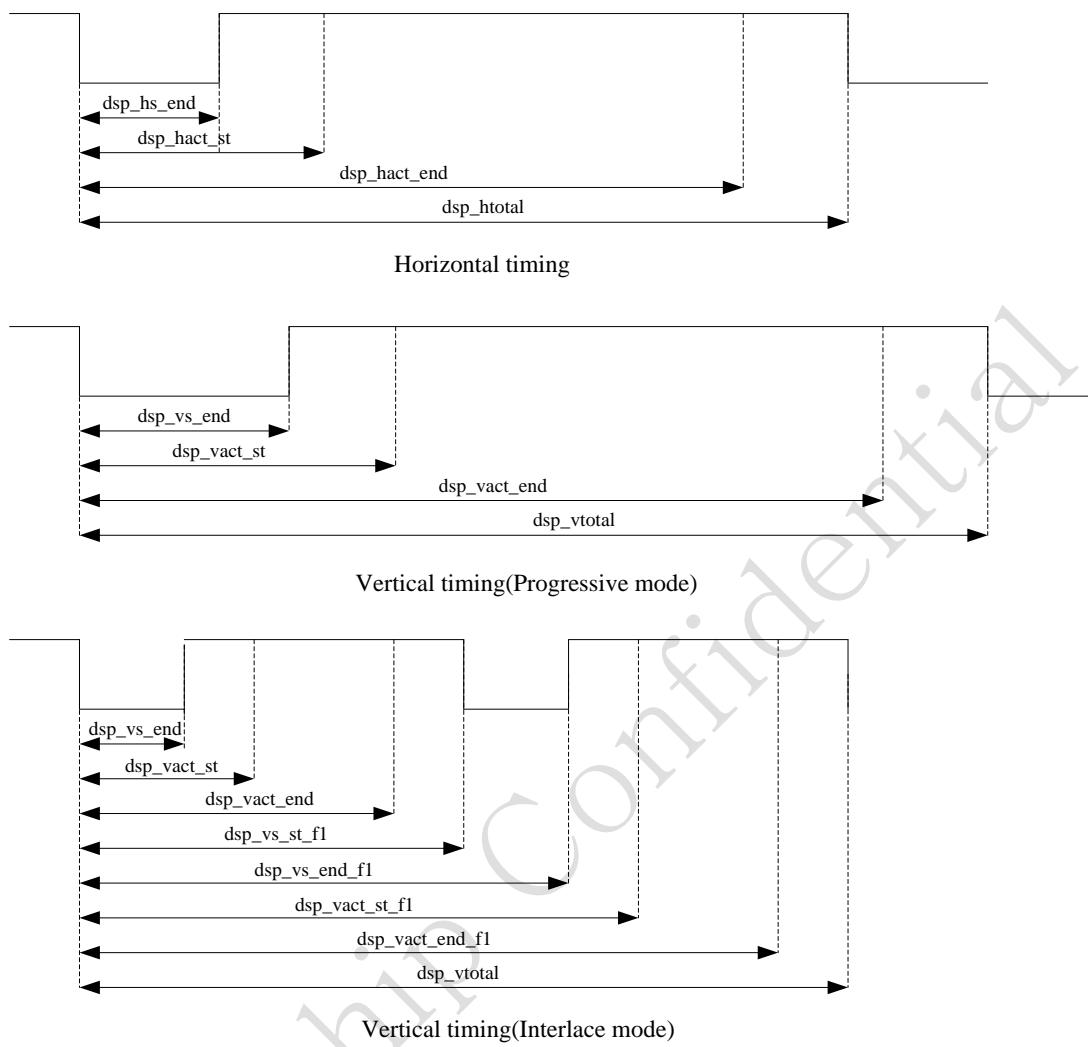


Fig. 18- 19VOP RGB interface timing setting

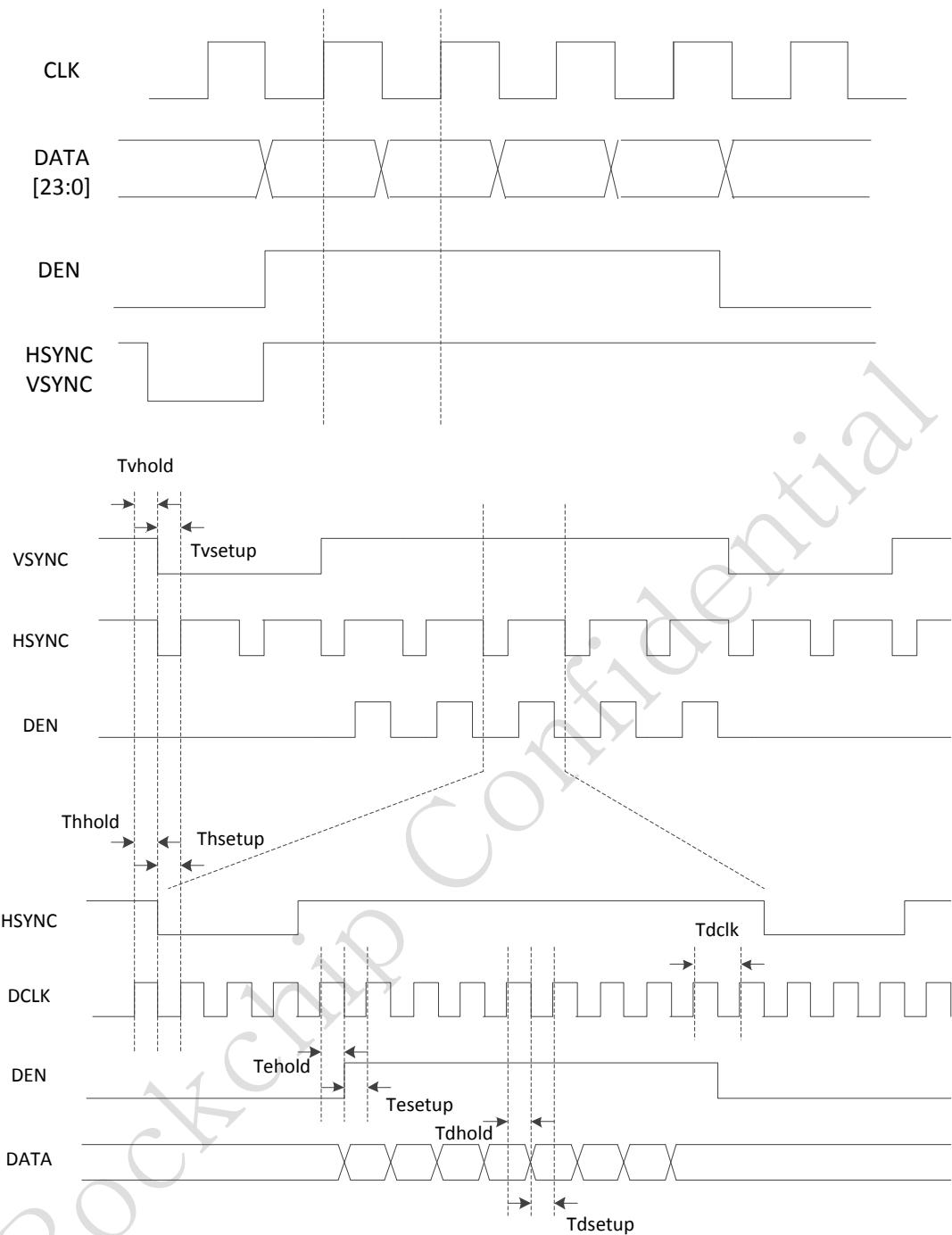


Fig. 18- 20 VOP RGB Interface Timing(SDR)

Table 18- 3VOP RGB interface (SDR) signal timing constant

(VDD_core = 0.9V to 1.1V, VDD_IO=3.0V to 3.6V , TA = -40°C to 125°C)

Item	Symbol	Min	Typ	Max	Unit
Display clock period	Tdclk	2.2		6	ns
VSYNC setup to DCLK falling edge	Tvsetup	0.5		0.6	ns
VSYNC hold from DCLK falling edge	Tvhold	1.8		2.0	ns

Hsync setup to DCLK falling edge	Thsetup	0.50		0.58	ns
Hsync hold from DCLK falling edge	Thhold	1.89		2.13	ns
DEN setup to DCLK falling edge	Tesetup	0.48		0.56	ns
DEN hold from DCLK falling edge	Tehold	1.85		2.1	ns
DATA setup to DCLK falling edge	Tdsetup	0.47		0.51	ns
DATA hold from DCLK falling edge	Tdhold	4.79		4.26	ns

18.6 Interface Description

18.6.1 VOP Outputs

VOP supports RGB, LVDS, HDMI, MIPI, and TVE output. the LVDS HDMI MIPI output interface is same as RGB interface, TVE output is connected to a DAC(10bit). VOP is suitable for different display mode by different usage, which is shown as follows.

Table 18- 4 VOP Control Pins Definition

Display mode	RGB	RGB	RGB Parallel 16-bit
	Parallel 24-bit	Parallel 18-bit	
DCLK	DCLK	DCLK	DCLK
VSYNC	VSYNC	VSYNC	VSYNC
Hsync	Hsync	Hsync	Hsync
DEN	DEN	DEN	DEN
DATA	DATA[23:0]	DATA[17:0]	DATA[15:0]

18.6.2 IOMUXdescription

18.7 Application Notes

18.7.1 DMA transfer mode

There are three DMA transfer modes for loading win0 or win1 frame data determined by following parameters(X=0,1):

dma_burst_length

winX_no_outstanding

winX_gather_en

winX_gather_thres

1. auto outstanding transfer mode(random transfer)

When `winX_no_outstanding` is 0, multi-bursts transfer command could be sent out to AXI master interface continuously if the internal memory has enough space to store new data. The continuous random burst number is in the range of 1 to 4, mainly depending on the empty level of internal memory, `dma_burst_length`, data format and active image width.

2. configured outstanding transfer mode(fixed transfer)

When `winX_gather_en` is 1, fixed-number of bursts transfer command should be sent out to AXI master interface continuously if the internal memory has enough space to store new data. The fixed-number is determined by `winX_gather_thres`. Since the internal memory size is limited, there is some restriction for the `winX_gather_thres` as follows.

Table 18- 5Gather configuration for all format

Gather Threshold	<code>dma_burst_length</code> =2' b00 (burst16)	<code>dma_burst_length</code> =2' b01 (burst8)	<code>dma_burst_length</code> =2' b10 (burst4)
YCbCr420	0	0, 1, 2	0, 1, 2, 3
YCbCr422			
YCbCr444			
ARGB888	0, 1, 2, 3	0, 1, 2, 3	0, 1, 2, 3
RGB888			
RGB565			
8BPP	0, 1, 2, 3	0, 1, 2, 3	0, 1, 2, 3

18.7.2 HWC data load

Hardware cursor data is refreshed when `hwc_load_en` is high, but not needed for every frame. And `hwc_load_en` will be high until hwc loading finished.

The HWC data size is 32x32 or 64x64, determined by `hwc_size`. The data in external memory should be 32-bit aligned or 64-bit aligned, and stored in VOP internal memory in 64-bit aligned as follows.

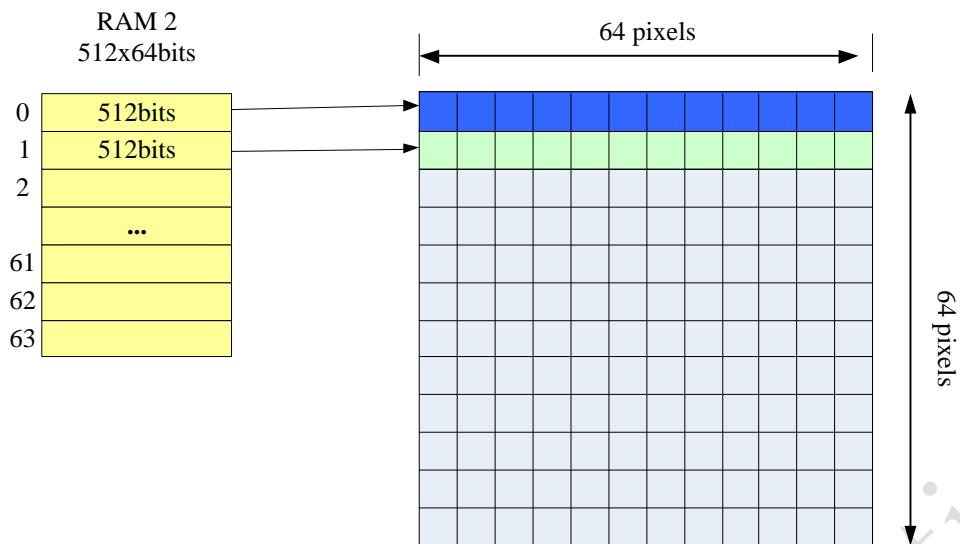


Fig. 18- 21 hwc data format

18.7.3 IEP direct path

There are two data source for win0/win1: external memory and IEP internal memory. However, the IEP data is just active for one layer at one frame time when IEP data path(SYS_CTRL[11]) is enable, determined by direct_path_layer_sel (SYS_CTRL[12]) signal. Moreover, it is suitable for win0 with scaling, but there is some limitation when picture scale down(because of there is not enough fifo to store data for scaling down) more than 4 times(horizontal multiple by vertical).

Direct path interface (DPI) can be used for VOP to display images from other image processing IPs, which also has DPI (slave).

There are programming flows for both DPI display on sequence and DPI display off sequence.

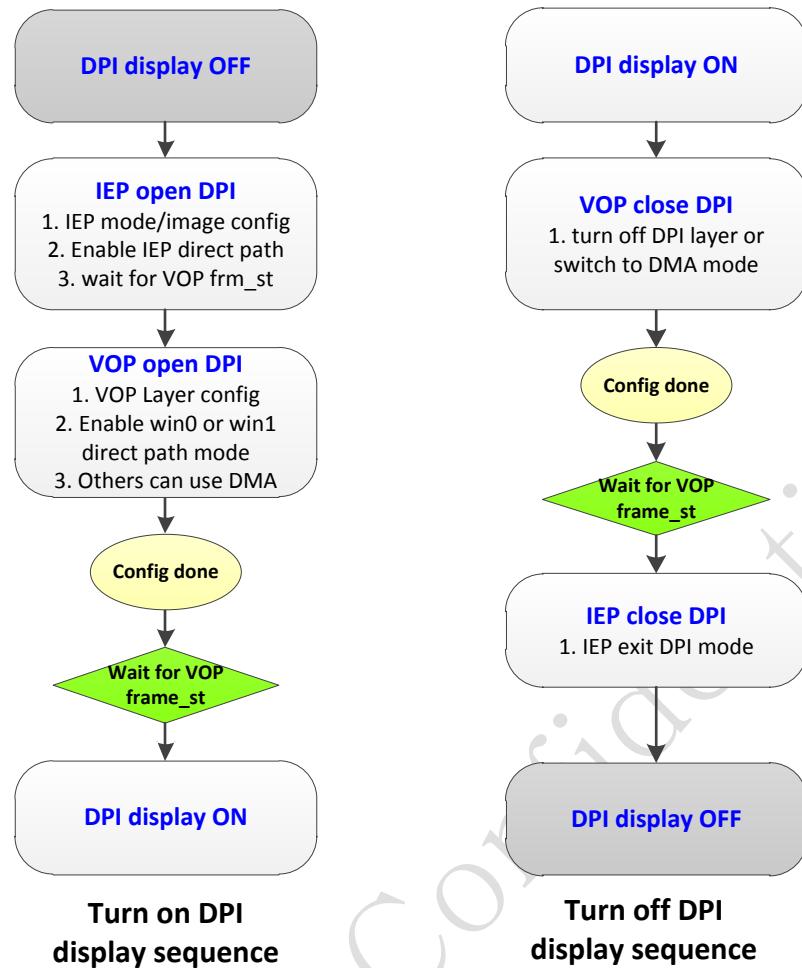


Fig. 18- 22 VOP DPI Programming Flow

1.Turn on DPI display

First, configure IEP into DPI mode. After doing image information and image processing mode configuration, IEP DPI mode can be turn on for display. IEP is in idle mode only if VOP's frame start input signal is valid.

Second, configure VOP for DPI display. Note that only one layer (Win0 or Win1) can use DPI in same frame. Other layers still can use internal DMA.

Finally, set VOP "config_done" to confirm all the new configuration and waiting frame sync to start DPI display actually.

2.Turn off DPI display

First, close VOP layer's DPI mode by turning off DPI layer or switching it to DMA mode. Then set VOP "config_done" to confirm new configuration.

Second, wait for VOP's frame synchronization to close DPI display in VOP.

Finally, turn off IEP's DPI mode.

18.7.4 GAMMA LUT

When dsp_lut_en is 0, the DSP LUT data should be refreshed by software. i.e., writing dsplut data to the internal memory with the start address DSP_LUT_MST.

The memory size is 256x24, i.e, lower 24bits valid, and the writing data number is determined by software.

18.7.5 DMA control (QoS/Hurry/Outstanding)

If you want to get higher priority for VOP to access external memory when the frame data is urgent, a QoS and hurry request can be generated and sent out basing on the configured values:

sw_noc_qos_en: AXI_BUS_CTRL[0]

sw_noc_qos_value: AXI_BUS_CTRL[2:1]

sw_noc_hurry_en: AXI_BUS_CTRL[3]

sw_noc_hurry_value: AXI_BUS_CTRL[5:4]

sw_noc_hurry_threshold: AXI_BUS_CTRL[9:6]

sw_axi_max_outstand_en: AXI_BUS_CTRL[11]

sw_axi_max_outstand_num: AXI_BUS_CTRL[16:12]

QoS request for higher bus priority for win1

NOC hurry for higher bus priority for VIO when win0 needs higher priority.

Max Outstanding num is configurable, but limited at 31 when mmu enables or 32 when mmu disables.

18.7.6 Interrupt

VOP interrupt is comprised of 9 interrupt sources:

- ✧ frame start interrupt
- ✧ line flag interrupt
- ✧ bus error interrupt
- ✧ win0 empty interrupt
- ✧ win1 empty interrupt
- ✧ scaler empty interrupt
- ✧ irq_mmu
- ✧ irq_tve
- ✧ dsp hold intrrupt

Every interrupt has independent interrupt enable signal(VOP_INT_EN), interrupt clear signal(VOP_INT_CLR), interrupt status signal (VOP_INT_STATUS).

MMU's and TVE's interrupt enable and clear signal are set in their own group, and are combined with others in top module. The last interrupt to outside is just a combined signal and high active.

There are 2 raw interruption:

- ✧ Line flag raw Interrupt status (INT_STATUS[31])

- ✧ Frame start raw interrupt status(INT_STATUS[30])

18.7.7 RGB display mode

RGB display mode is used for RGBpanel display. It is a continuous frames display mode.

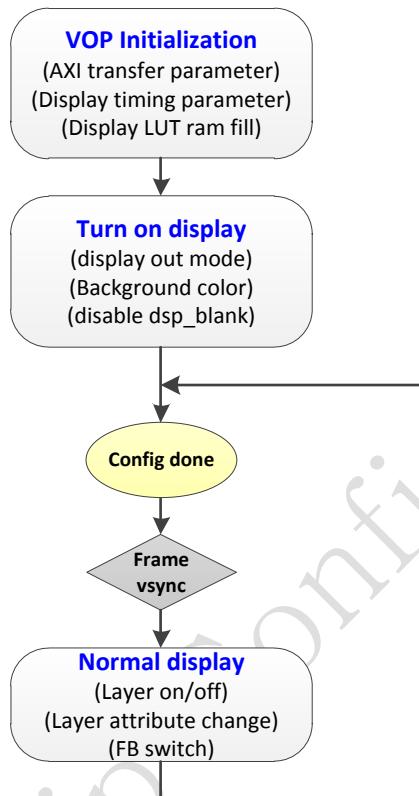


Fig. 18- 23 VOP RGB Mode Programming Flow

1.VOP initialization

VOP initialization should be done before turning display on.

First, AXI bus parameter (VOP_AXI_BUS_CTRL) should be set for DMA transfer.

Second, display panel/interface timing should be set for display output. The registers are: VOP_DSP_HTOTAL_HS_END/ VOP_DSP_HACT_ST_END/ VOP_DSP_VTOTAL_HS_END/ VOP_DSP_VACT_ST_END/ VOP_DSP_VS_ST_END_F1/ VOP_DSP_VACT_ST_END_F1

2.Background display

Before normal display, the background display could be turn on.

First, set display output mode (VOP_DSP_CTRL0/1) according to display device.

Second, disable dsp_blank mode, which would not be enable until frame synchronization.

Finally, writing '1' to "VOP_REG_CFG_DONE" register then all the frame-sync registers will be enable at the beginning of next frame.

3.Normal display

In normal display, all the display layers' attribute could be different according display scenario. So there is a programming loop in this mode.

First, configure all the display layers' attribute registers for the change of image format, location, size, scaling factor, alpha and overlay and so on. Those register would not be enable until frame synchronization.

Finally, write 1 to "VOP_REG_CFG_DONE" register then all the frame-sync registers will be enable at the beginning of next frame.

18.7.8 Immediately control register

There are two type registers in VOP , one is effective immediately, the other is effective by frame sync. Effective immediately registers list as follows, other registers are all effective by frame sync.

Table 18- 6effective immediately register table

register address	description
0x000	some ctrl function bits
0x004	some dsp ctrl function bits
0x008	background color register dsp control function bits
0x014	Alpha control register
0x018	win0 color key register
0x01c	win1 color key register
0x06c~0x080	dsp_timing register
0xe0~0xf4	Frc configuration bits

18.7.9 Output Polarity Control

There are five channels output ,list as follow:

Tve_dac_dclk_en ,tve_dac_dclk_inv;

Hdmi_dclk_en,hdmi_dclk_inv;

Rgb_dclk_en,rgb_dclk_inv;

Lvds_dclk_en,lvds_dclk_inv;

Mipi_dclk_en,mipi_dclk_inv.

The xxx_dclk_en should be set to 1 when output select the xxx channel, and the other channel's xxx_dclk_en should be set to 0 to gate the output clk and data.

The rgb, lvds, mipi channel are mutual exclusion . They share one pair of output

polarity control registers(DSP_CTRL0[6:4]).

HDMI channel polarity control registers maps to INT_SCALER[6:4].

When using RGB panel, the dclk should be tied to "0" or "1" in some scenarios.

In this case ,you should enable sw_io_pad_clk_sel ,to tie dclk to "0". IF enable rgb_dclk_inv at the same time, the dclk will tie to "1".

18.7.10 Scaler

1. Program sequence

(1) set PLL and CPLL as slow mode;

```
set CRU_MODE_CON = 0xff000000;
```

(2) config dclk and sclk

You can use CPLL or GPLL for sclk or dclk.

sclk needs PLL fraction mode to get precise clock.

sclk has been figured out as the flowing table.

(3) set CPLL/GPLL as normal mode;

(4) set VOP including scaler configuration, refer to the following table.

Assert scaler_en after all scaler configuration registers are set.

Table 18- 7VOP Scaler configuration

HDMI				PAD				SCALER configuration			
TYPE	dclk	width	height	sclk	width	height	in_vst	dsp_vst	dsp_hst	h_factor	v_factor
1080P @60Hz	148.5	1920	1080	70.400002	1280	800	4	32	1059	0x1801	0x159b
				57.599998	1024	768	4	17	1946	0x1e03	0x1682
				50.400002	1024	600	4	4	1040	0x1e03	0x1cd2
				26.73	800	480	9	23	1586	0x266d	0x2402
				39.599998	800	600	4	4	1339	0x266d	0x1cd2
				39.599998	800	600	4	28	19	0x266d	0x1cd2

1080P @50Hz	148.5	1920	1080	58.666668	1280	800	1	30	1233	0x1801	0x159b
				48	1024	768	1	15	2297	0x1e03	0x1682
				42	1024	600	1	2	1210	0x1e03	0x1cd2
				22.275	800	480	9	23	1865	0x266d	0x2402
				33	800	600	4	4	1568	0x266d	0x1cd2
				33	800	600	4	27	2624	0x266d	0x1cd2
720P @60Hz	74.25	1280	720	70.400002	1280	800	1	19	143	0x1000	0xe65
				57.999998	1024	768	1	9	587	0x1401	0xeff
				50.400002	1024	600	1	0	684	0x1401	0x1334
				26.73	800	480	1	8	957	0x199c	0x1804
				39.599998	800	600	4	2	833	0x199c	0x1334
				39.599998	800	600	4	18	173	0x199c	0x1334
720P @50Hz	74.25	1280	720	58.666668	1280	800	1	19	120	0x1000	0xe65
				48	1024	768	1	9	652	0x1401	0xeff
				42	1024	600	1	0	769	0x1401	0x1334
				22.275	800	480	1	8	1096	0x199c	0x1804
				33	800	600	4	2	948	0x199c	0x1334
				33	800	600	4	18	156	0x199c	0x1334
576P @50Hz	27	720	480	61.111111	1280	800	1	39	774	0x8fc	0xb83
				50	1024	768	1	32	96	0xb3e	0xbfe
				43.75	1024	600	1	24	828	0xb3e	0xf5b
				23.203123	800	480	1	31	424	0xe68	0x1334
				34.375	800	600	4	27	26	0xe68	0xf5b
				34.375	800	600	4	39	441	0xe68	0xf5b
480P @60Hz	27	720	480	73.846153	1280	800	1	33	81	0x8fc	0x997
				60.419579	1024	768	1	25	521	0xb3e	0x9fd
				52.867134	1024	600	1	20	555	0xb3e	0xccb

				28.038462	800	480	1	26	77	0xe68	0x1000
				73.846153	800	600	1	19	606	0xe68	0xccb
				60.419579	800	600	1	30	92	0xe68	0xccb

```
*in_vst:scl_in_vsync_vst;
dsp_vst:scl_dsp_frame_vst;
dsp_hst:scl_dsp_frame_hst;
```

2. sclk Requirement

Sclk(the output pixel clock) is asynchronous with input pixel clock to support different display resolution. But the input/output frame frequency should be same (FSin=FSout)because there is no frame buffer in SCALER. So there is strict clock requirement for output pixel clock, which is generated from CPLL or GPLL.

Assume that Fout is the output pixel clock frequency and Fin is the input pixel clock frequency.

```
dsp_in_vtotal*dsp_in_htotal*Fin=dsp_vtotal*dsp_in_htotal*Fout;
```

The total output active frame period (include horizontal blank period, but without vertical blank period) must be close to the total input active frame period. The difference between them should be less than Xmax input active line period(include horizontal blank period).

3. Output frame scanning start point setting

To meet the requirement of active pixel start point, there are dsp_frm_hst/dsp_frm_vst settings for frame scanning start point adjustment.

The timing delay of output active frame pixel start point behind input active frame pixel start point can be calculated as following equation.

Usually,we set Delta_T as follow:

```
ifv_scale_ratio<2,Delta_T=4* dsp_in_htotal*1/Fin; or Delta_T=5*
dsp_in_htotal*1/Fin;
ifv_scale_ratio>=2,Delta_T=12* dsp_in_htotal*1/Fin;
```

$$T_{frm_st} = (T_{BP_in} + Delta_T - T_{BP_out});$$

In where,

$$T_{frm_st} = (dsp_frm_vst*dsp_in_htotal+dsp_frm_hst)/Fin;$$

T_{BP_in} is the blank period of input frame;

T_{BP_out} is the blank period of output frame and.

if($T_{frm_st}<0$),Delta_T should be adjusted as below:

$$T_{frm_st}=T_{in} - T_{frm_st};$$

then we can calc out dsp_frm_vst and dsp_frm_hst as below:

```
dsp_frm_vst = floor((T_frm_st*Fin)/dsp_in_htotal);  
dsp_frm_hst = (T_frm_st-dsp_frm_vst*dsp_in_htotal/Fin)*Fin;
```

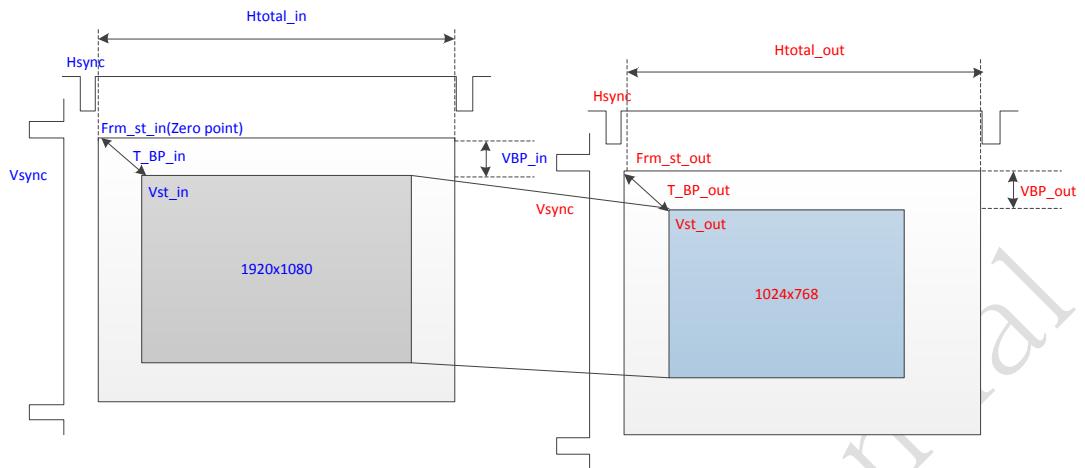


Fig. 18- 24Scaler frame scanning start point

Chapter 19 RGA1_plus

19.1 Overview

RGA is a separate 2D raster graphic acceleration unit. It accelerates 2D graphics operations, such as point/line drawing, image scaling, rotation, BitBLT, alpha blending and image blur/sharpness.

19.1.1 Features

- **Data format**
 - Input data: ARGB/RGB888/RGB565/YUV420/YUV422
 - Output data: ARGB/RGB888/RGB565 (YUV420/YUV422 for blur/sharpness);
 - Pixel Format conversion, BT.601/BT.709
 - Dither operation
 - Max resolution: 8192x8192 source image, 2048x2048 frame buffer
- **Scaling**
 - Down-scaling and up-scaling
 - Three sampling modes: Nearest sampling (Stretched BitBLT), Bi-linear filter or Bi-cubic filter
 - Arbitrary non-integer scaling ratio, from 1/2 to 8
 - Average filter pre-scaling (2's Down-scaling bypass path, not available with other 2D operation)
- **Rotation**
 - Arbitrary rotation, minimum 1 degree step
 - No per-pixel alpha in arbitrary rotation (without 90, 180, 270)
 - x-mirror, y-mirror
- **BitBLT**
 - Block transfer
 - Color palette (with transparency mode)/Color fill
 - Transparency mode (color keying/stencil test, specified value/range)
- **Alpha Blending**
 - Per-pixel/user-specified alpha blending (Porter-duff alpha support)
 - Fading
 - Anti-aliasing (for rotation)
- **Raster operation**
 - ROP2/ROP3/ROP4
 - No ROP in arbitrary rotation (except 90/180/270 degree)
- **YUV_Output**
 - Output data: YUV422SP/YUV420SP, not used in Pre_scaling Line/point drawing and Blur/sharp, Alpha and ROP.
 - ColorPalette alpha 1/8bit mode
- **RW_Align**
 - AXI read/write can be DDR-Align in Line scan mode, high performance.
- **Line/Point drawing**
 - Bresonham algorithm, Specified width
 - Anti-aliasing
- **Blur/sharpness**
 - Bypass post processing path (not available with other 2D operation)
 - Tile_based

19.2 Block Diagram

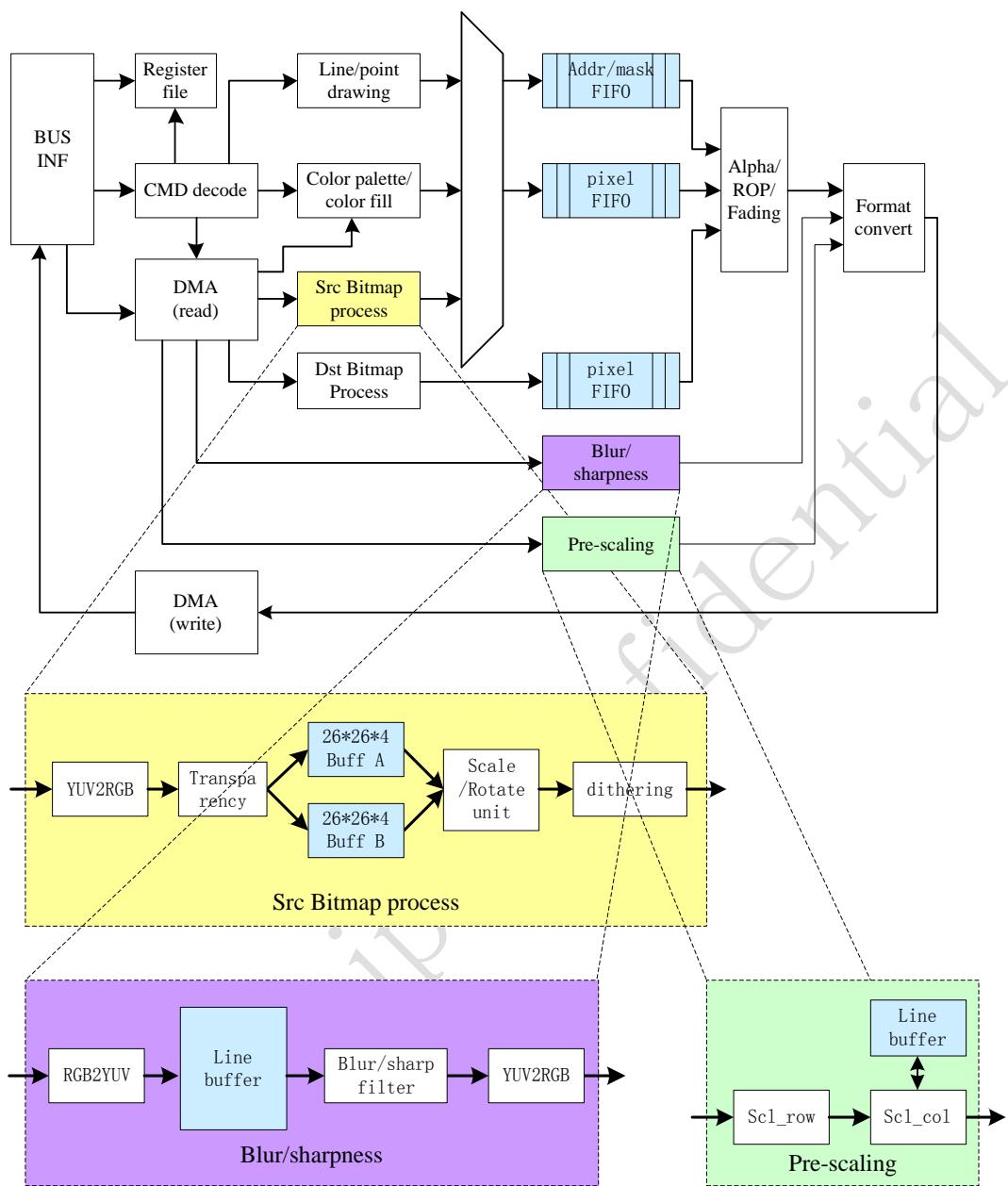


Fig. 19-1 RGA Block Diagram

19.3 Function Description

19.3.1 Data Format

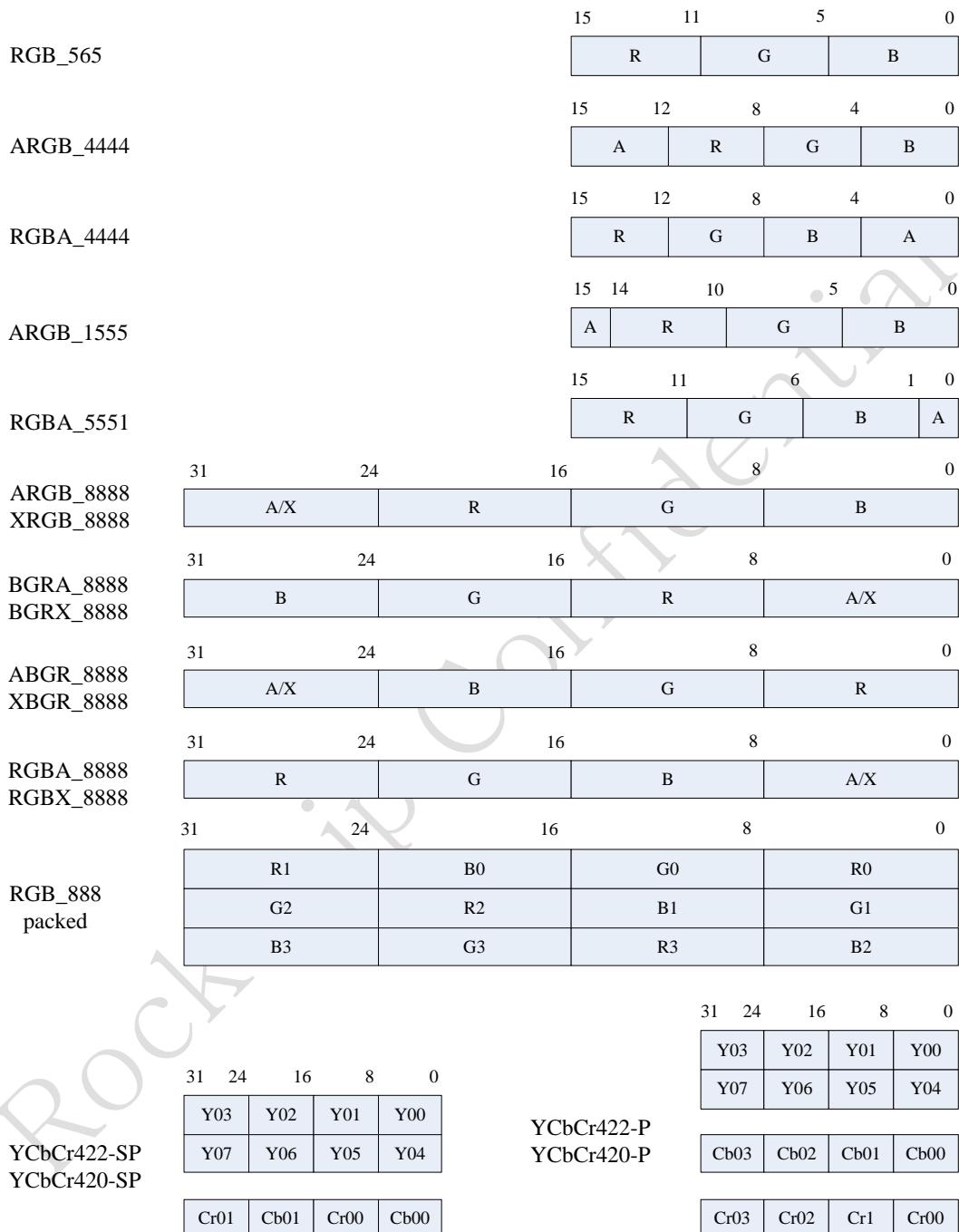


Fig. 19-2 RGA Input Data Format

All input datas (defined by SRC_IN_FMT/DST_IN_FMT) are converted to ABGR8888. The results are converted to the output data format (defined by DST_OUT_FMT).

19.3.2 Dithering

There could have dithering operation for source image when the source image format is not RGB565 and the destination format is RGB565.

The down-dithering is done using Dither Matrix.

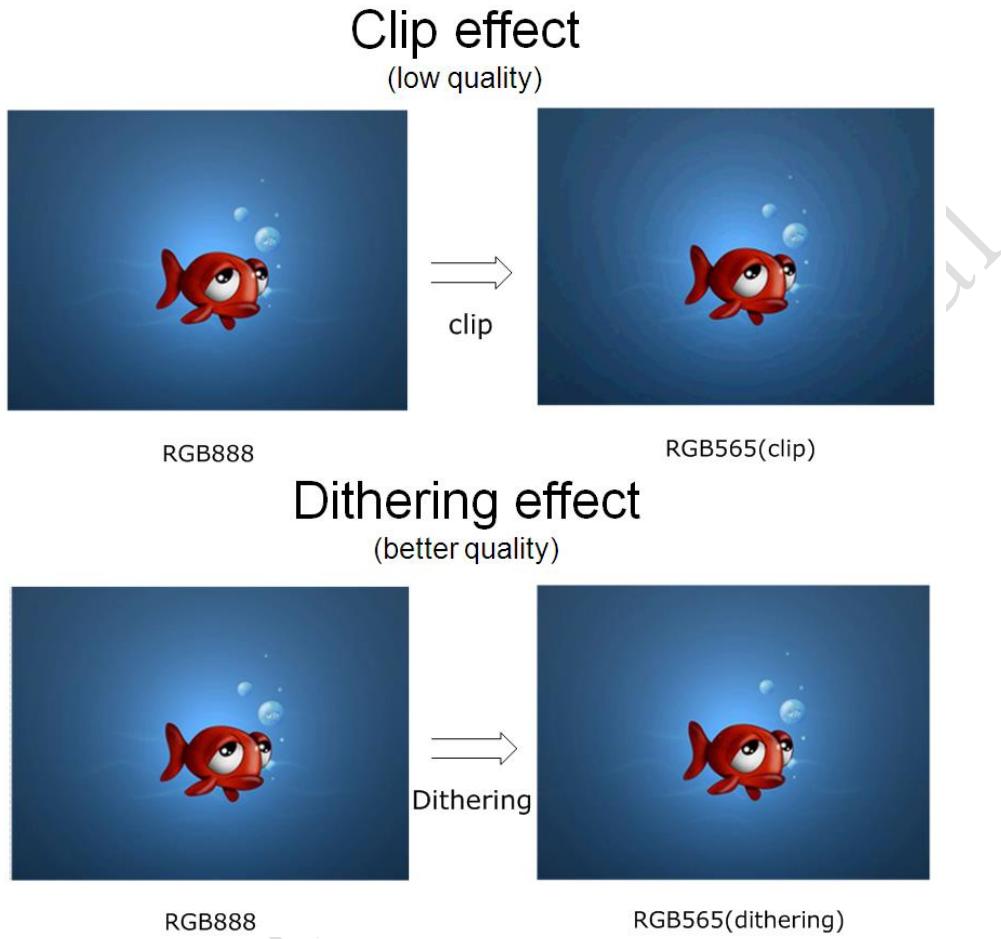


Fig. 19-3 RGA Dither effect

19.3.3 Scaling

The scaling operation is the imageresizing processing of source image. Scaling is done base on ARGB8888 format.

There are three sampling modes: Nearest sampling (Stretched BitBLT), Bi-linear filter or Bi-cubic filter.

19.3.4 Rotation

Arbitrary rotation and x-mirror, y-mirror operation is supported in RGA. The rotation operation is combined with scaling operation.

Alpha is available only if there is no rotation or 90-degree/180-degree /270-degree rotation or x-mirror/y-mirror. Anti-aliasing is done by the alpha blending of the boundary pixels.

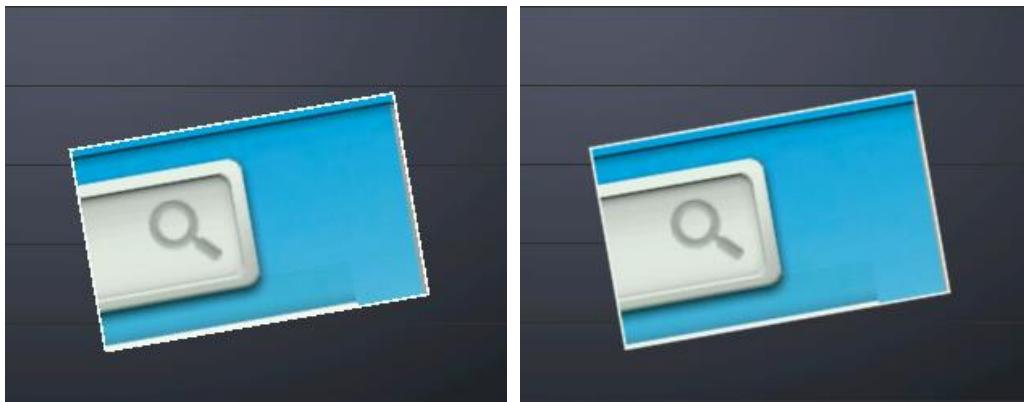


Fig. 19-4 RGA Rotation AA effect

19.3.5 Bitmap Block Transfer

BitBlit is a computer graphics operation in which several bitmaps are combined into one using a raster operator. The bitmap rectangular block is transferred from source frame buffer to destination frame buffer. There are three bitmap block transfer type: bitmap block transfer (RGB/YCbCr), color expansion, and solid fill.

RGA also supports transparency mode in BitBLT. There are two transparency modes (stencil test): normal mode and inverted mode.

There are 4 enable control bits for ARGB color channel for stencil test, which can be set independently.

1. Transparency mode

- (1) Normal Stencil test (Color keying)

Pixels with the same color or in the range of user-specified colors are discarded.

- (2) Inverted stencil test

Pixels with the different color or out the range of user-specified colors are discarded.

2. Color palette

1bpp/2bpp/4bpp/8bpp palette data formats are support in RGA source layer. 1bpp color expansion can be BG color and FG color, transparency and FG color according the alpha enable bit. There is a 256x25bit LUT in RGA for 2bpp/4bpp/8bpp color palette. The following is the table of 8bpp with alpha enable bit in the MSB.

Table 19-1 RGA 8bpp color palette LUT

INDEX\Bit Pos.	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0
00H	A E N	R 7	R 6	R 5	R 4	R 3	R 2	R 1	R 0	G 7	G 6	G 5	G 4	G 3	G 2	G 1	G 0	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
01H	A E N	R 7	R 6	R 5	R 4	R 3	R 2	R 1	R 0	G 7	G 6	G 5	G 4	G 3	G 2	G 1	G 0	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
.....
FFH	A E N	R 7	R 6	R 5	R 4	R 3	R 2	R 1	R 0	G 7	G 6	G 5	G 4	G 3	G 2	G 1	G 0	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0

3. Color fill

Two modes of color fill can be done by RGA: solid fill and gradient fill.

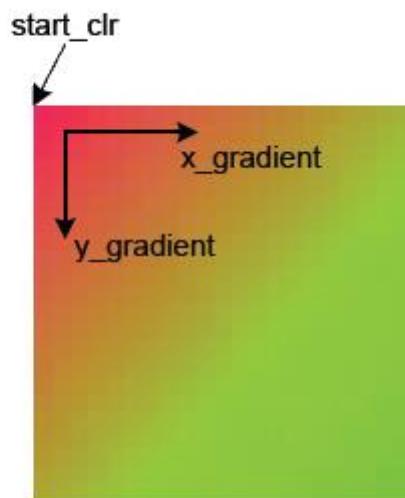


Fig. 19-5 RGA Gradient Fill

Gradient fill using following equations for ARGB calculation of every pixel in different coordinate.

$$\begin{aligned}
 A_{cur} &= (A_{start} + x*x_A_gradient) + y*y_A_gradient; \\
 R_{cur} &= (R_{start} + x*x_R_gradient) + y*y_R_gradient; \\
 G_{cur} &= (G_{start} + x*x_G_gradient) + y*y_G_gradient; \\
 B_{cur} &= (B_{start} + x*x_B_gradient) + y*y_B_gradient;
 \end{aligned}$$

A_{start} , R_{start} , G_{start} , B_{start} is the ARGB value of start point. There are four pairs of values for horizontal and vertical gradient. Saturation operation could be enabled or disabled if the color overflows 255 or underflows 0.

19.3.6 Alpha Blending

Alpha blending is divided to two stages. The first stage is mix alpha

(per-pixel/user-specified), where Porter-Duff (pre-multiplied) alpha is supported. The second stage is fading.

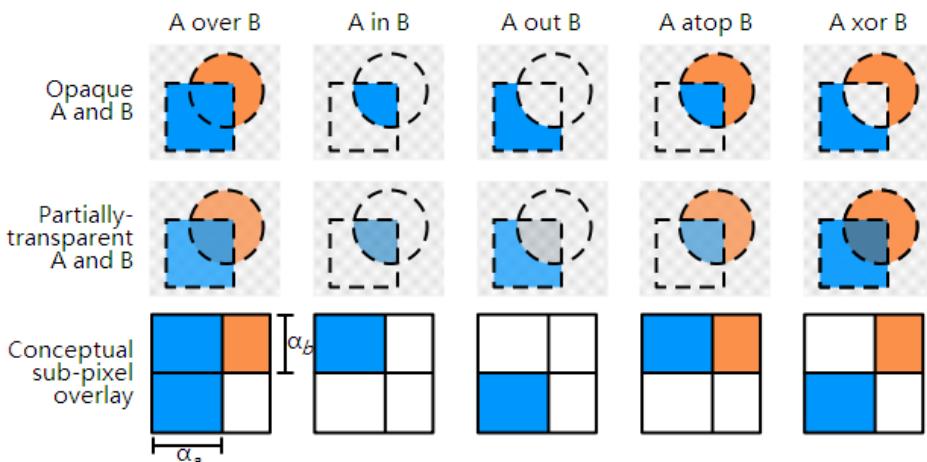


Fig. 19-6 RGA Alpha blending

1. Mix (per-pixel/user-specified) alpha

$$data = (source \times (ALPHA + 1) + destination \times (255 - ALPHA)) \gg 8$$

2. Porter-Duff

Porter-Duff alpha is a premultiplied alpha between two layers.

Porter-Duff formula:

$$C_r = C_s * F_s + C_d * F_d$$

$$A_r = A_s * F_s + A_d * F_d$$

(C – color, A – alpha, s – source, d – destination, r – result, F – factor)

There are 12 different mix types for Fs and Fd factor.

Table 19-2 RGA Porter-Duff alpha factor

NO.	type	Source factor	Destination factor
1	CLEAR	0	0
2	SRC	1	0
3	DST	0	1
4	SRC OVER	1	(1-As)
5	DST OVER	(1-Ad)	1
6	SRC IN	Ad	0
7	DST IN	0	As
8	SRC OUT	(1-Ad)	0
9	DST OUT	0	(1-As)
10	SRC ATOP	Ad	(1-As)
11	DST ATOP	(1-Ad)	As
12	XOR	(1-Ad)	(1-As)

3. Fading

$$data = ((source \times (ALPHA + 1) \gg 8) + fading_offset$$

19.3.7 Raster Operation (ROP)

Raster operation (ROP) is a Boolean operation between operands, which involve AND, OR, XOR, and NOT operations. For ROP2, operands are P (select pan) and D (Destination bitmap). For ROP3, operands are P (pattern), S (source bitmap) and D (Destination bitmap). For ROP4, operands are P (pattern), S (source bitmap), D (Destination bitmap) and MASK.

Table 19-3 RGA ROP Boolean operations

Operator	Meaning
a	Bitwise AND
n	Bitwise NOT (inverse)
o	Bitwise OR
x	Bitwise exclusive OR (XOR)

19.3.8 Line/Point Drawing

Line operation draws a line, which coordinates for two points are given: start point and end point. The end point can be drawn or not drawn. Lines are rendered using the Bresenham algorithm.

The width of the line can be user-specified. Anti-aliasing is done to improve the display quality.

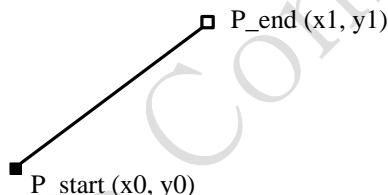


Fig. 19-7 RGA Line drawing

19.3.9 Blur/sharpness

Blur and sharpness is a post processing for destination image. A 8x8 matrix filter is used.

Blur/sharpness unit is also a bypass unit, which is not available when doing other 2D graphic operations.

19.4 Register Description

19.4.1 Register Summary

Name	Offset	Size	Reset Value	Description

Name	Offset	Size	Reset Value	Description
RGA_SYS_CTRL	0x0000	W	0x00000000	RGA system control register
RGA_CMD_CTRL	0x0004	W	0x00000000	RGA command code control
RGA_CMD_ADDR	0x0008	W	0x00000000	RGA command codes start address register
RGA_STATUS	0x000c	W	0x00000000	RGA status register
RGA_INT	0x0010	W	0x00000000	RGA interrupt register
RGA_AXI_ID	0x0014	W	0x49850210	RGA AXI ID setting register
RGA_MMU_STA_CTRL	0x0018	W	0x00000000	RGA MMU statistic ctrl
RGA_MMU_STA	0x001c	W	0x00000000	RGA MMU statistic data
RGA_WORK_CNT	0x0020	W	0x00000000	RGA work cycle counter
RGA_VERSION	0x0028	W	0x02018632	RGA version num
RGA_MODE_CTRL	0x0100	W	0x00000000	RGA mode control register
RGA_SRC_Y_MST	0x0104	W	0x00000000	Source image Y/RGB/line drawing start addr
RGA_SRC_CB_MST	0x0108	W	0x00000000	Source image Cb/Cbr start addr
RGA_SRC_CR_MST	0x010c	W	0x00000000	Source image Cr/color palette start addr
RGA_SRC_VIR_INFO	0x0110	W	0x00000000	Source image virtual width
RGA_SRC_ACT_INFO	0x0114	W	0x00000000	Source image active width/height
RGA_SRC_X_PARA	0x0118	W	0x00000000	Source image horizontal scaling/rotation parameter
RGA_SRC_Y_PARA	0x011c	W	0x00000000	Source image vertical scaling/rotation parameter
RGA_SRC_TILE_XINFO	0x0120	W	0x00000000	Source tile start point coordinate, Source tile width
RGA_SRC_TILE_YINFO	0x0124	W	0x00000000	Source tile start point coordinate, Source tile height
RGA_SRC_TILE_H_INCR	0x0128	W	0x00000000	Source tile horizontal X/Y increment value
RGA_SRC_TILE_V_INCR	0x012c	W	0x00000000	Source tile vertical X/Y increment value
RGA_SRC_TILE_OFFSETX	0x0130	W	0x00000000	Source tile start point x for DST tile start point remap
RGA_SRC_TILE_OFFSETY	0x0134	W	0x00000000	Source tile start point y for DST tile start point remap
RGA_SRC_BG_COLOR	0x0138	W	0x00000000	Source image background color

Name	Offset	Size	Reset Value	Description
RGA_SRC_FG_COLOR	0x013c	W	0x00000000	Source image foreground color
RGA_SRC_TR_COLOR0	0x0140	W	0x00000000	Source image transparency color min value
RGA_CP_GR_A	0x0140	W	0x00000000	RGA color gradient fill step register (color fill mode)
RGA_SRC_TR_COLOR1	0x0144	W	0x00000000	Source image transparency color max value, Color gradient fill st
RGA_CP_GR_B	0x0144	W	0x00000000	RGA color gradient fill step register (color fill mode)
RGA_LINE_DRAW	0x0148	W	0x00000000	Point/line drawing setting
RGA_PAT_ST_POINT	0x0148	W	0x00000000	RGA pattern start point
RGA_DST_MST	0x014c	W	0x00000000	Destination image start addr
RGA_DST_VIR_INFO	0x0150	W	0x00000000	Destination image virtual width
RGA_DST_CTR_INFO	0x0154	W	0x00000000	Destination image control window active width/height
RGA_ALPHA_CON	0x0158	W	0x00000000	Alpha blending/ROP mode register
RGA_PAT_CON	0x015c	W	0x00000000	Pattern size/offset
RGA_DST_VIR_WIDTH	0x015c	W	0x00000000	Register0000 Abstract
RGA_ROP_CON0	0x0160	W	0x00000000	Raster operation code0 control register
RGA_CP_GR_G	0x0160	W	0x00000000	Color gradient fill step of green
RGA_PRESCL_CB_MST	0x0160	W	0x00000000	RGA pre-scale Cb destination start addr
RGA_ROP_CON1	0x0164	W	0x00000000	Raster operation code1 control register
RGA_CP_GR_R	0x0164	W	0x00000000	Color gradient fill step of red
RGA_PRESCL_CR_MST	0x0164	W	0x00000000	RGA pre-scale Cr destination start addr
RGA_MMU_CTRL	0x0168	W	0x00000000	MMU control register
RGA_MMU_TLB	0x016c	W	0x00000000	RGA MMU TLB base address
RGA_YUV_OUT_CFG	0x0170	W	0x00000000	Destination YUV output configuration
RGA_DST_UV_MST	0x0174	W	0x00000000	Destination image UV start add

Notes: Size : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** - WORD (32 bits) access

19.4.2 Detail Register Description

RGA_SYS_CTRL

Address: Operational Base + offset (0x0000)

RGA system control register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	acg_en RGA auto clock gating enable bit 0: disable 1: enable
2	RW	0x0	cmd_mode RGA command mode 0: slave mode 1: master mode
1	W1C	0x0	op_st RGA operation start bit Only used in passive (slave) control mode
0	W1C	0x0	soft_reset RGA soft reset write '1' to this would reset the RGA engine except config registers.

RGA_CMD_CTRL

Address: Operational Base + offset (0x0004)

RGA command code control

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:3	RW	0x000	cmd_incr_num RGA command increment number
2	RW	0x0	cmd_stop RGA command stop mode Command execution would stop after the current graphic operation finish if set this bit to '1'.
1	W1C	0x0	cmd_incr_valid RGA command increment valid (Auto cleared) When setting this bit, 1. The total cmd number would increase by the RGA_INCR_CMD_NUM. 2. RGA would continue running if idle.
0	W1C	0x0	cmd_line_fet_st RGA command line fetch start (command line reset) (Auto cleared) When fetch start, the total cmd number would reset to RGA_INCR_CMD_NUM.

RGA_CMD_ADDR

Address: Operational Base + offset (0x0008)

RGA command codes start address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cmd_addr RGA command codes start address

RGA_STATUS

Address: Operational Base + offset (0x000c)

RGA status register

Bit	Attr	Reset Value	Description
31:20	RO	0x000	cmd_total_num RGA command total number
19:8	RO	0x000	cur_cmd_num RGA current command number
7:1	RO	0x0	reserved
0	RO	0x0	engine_status RGA engine status 0: idle 1: working

RGA_INT

Address: Operational Base + offset (0x0010)

RGA interrupt register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10	RW	0x0	all_cmd_finish_int_en All command finished interrupt enable
9	RW	0x0	mmu_int_en MMU interrupt enable
8	RW	0x0	error_int_en Error interrupt enable
7	W1C	0x0	cur_cmd_finish_int_clr Current command finished interrupt clear(auto clear)
6	W1C	0x0	all_cmd_finish_int_clr All command finished interrupt clear(auto clear)
5	W1C	0x0	mmu_int_clr MMU interrupt clear(auto clear)
4	W1C	0x0	error_int_clr Error interrupt clear(auto clear)

Bit	Attr	Reset Value	Description
3	RO	0x0	cur_cmd_finish_int_flag Current command finished interrupt flag
2	RO	0x0	all_cmd_finish_int_flag All command finished interrupt flag
1	RO	0x0	mmu_int_flag MMU interrupt flag
0	RO	0x0	error_int_flag Error interrupt flag

RGA_AXI_ID

Address: Operational Base + offset (0x0014)

RGA AXI ID setting register

Bit	Attr	Reset Value	Description
31:30	RW	0x1	mmu_rid MMU read channel address mapping axi bus ID Note: Don't use the same ID with RGA axi bus ID.2'b11, [31:30].
29:28	RW	0x0	mmu_wid MMU write channel address mapping axi bus ID Note: Don't use the same ID with RGA axi bus ID.2'b11, [29:28].
27:24	RW	0x9	mask_rid mask read AXI ID
23:20	RW	0x8	cmd_rid CMD fetch AXI ID
19:16	RW	0x5	dst_wid DST write AXI ID
15:12	RW	0x4	dst_rid DST/LUT/PAT read AXI ID
11:8	RW	0x2	src_cr_rid SRC Cr read AXI ID
7:4	RW	0x1	src_cb_rid SRC Cb read AXI ID
3:0	RW	0x0	src_yrgb_rid SRC YRGB read AXI ID

RGA_MMU_STA_CTRL

Address: Operational Base + offset (0x0018)

RGA MMU statistic ctrl

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	W1C	0x0	mmu_sta_cnt_clr TLB statistic counter clear(auto clear) 0: no clear 1: clear After be set to 1, this bit will clear by itself 1 cycle later
2	W1C	0x0	mmu_sta_resume TLB statistic resume(auto clear) After be set to 1, this bit will clear by itself 1 cycle later.
1	RW	0x0	mmu_sta_pause TLB statistic pause Note: before reading MMU_TLB_STATISTIC, this bit must be set to 1.
0	RW	0x0	mmu_sta_en TLB statistic enable 0: disable 1: enable

RGA_MMU_STA

Address: Operational Base + offset (0x001c)

RGA MMU statistic data

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	tlb_miss_sta_cnt TLB miss statistic counter
15:0	RO	0x0000	tlb_hit_sta_cnt TLB hit statistic counter

RGA_WORK_CNT

Address: Operational Base + offset (0x0020)

RGA work cycle counter

Bit	Attr	Reset Value	Description
31:27	RO	0x0000	Reserved
26:0	RO	0x0000	Rga_work_cnt Rga work cycle counter

RGA_VERSION

Address: Operational Base + offset (0x0028)

RGA Version num

Bit	Attr	Reset Value	Description
31:24	RO	0x0002	MAJOR: RGA1: 1; RGA1_Plus: 2; RGA2: 3; RGA2_Lite: 4;
23:16	RO	0x0001	MINOR; 1: for AUDI;

Bit	Attr	Reset Value	Description
15:0	RO	0x8632	ASCII code: mmip svn num=8632

RGA_MODE_CTRL

Address: Operational Base + offset (0x0100)

RGA mode control register

Bit	Attr	Reset Value	Description
31	RW	0x0	alpha_source_sel alpha select for alpha mix mode 0: SRC alpha: SRC*As+DST*(1-As) 1: DST alpha: SRC*(1-Ad)+DST*Ad
30	RW	0x0	alpha_zero_key_mode ARGB888 alpha zero key mode 0x000000 would be changed to 0x000100(RGB888)/0x0020(RGB565)for ARGB888 to RGBX/RGB565 color key 0: disable 1: enable
29	RW	0x0	cur_cmd_finish_int_en Current command finished interrupt enable
28	RW	0x0	endian_swap Color palette endian swap 0: big endian 1: little endian
27	RW	0x0	dst_alpha_swap Destination bitmap data alpha swap 0: ABGR 1: BGRA
26	RW	0x0	dst_rb_swap Destination bitmap data RB swap 0: BGR 1: RGB
25	RW	0x0	dst_rgb_pack Destination bitmap BGR packed 0: ABGR 1: BGR packed
24:23	RW	0x0	dst_data_fmt Destination bitmap data format(Collor fill/ROP pattern data format) 00: XBGR888/ABGR888 01: RGB565 10: ARGB1555 11: ARGB4444
22	RW	0x0	pat_mode Color fill/ROP4 pattern 0: solid color 1: pattern color

Bit	Attr	Reset Value	Description
21:20	RW	0x0	src_filter_type SRC rotation/mirror mode[3:2]: filter type 00: nearest neighbor 01: bi-linear 10: bi-cubic
19:18	RW	0x0	src_rotate_mode SRC rotation/mirror mode[1:0] 00: bypass 01: rotation 10: x mirror 11: y mirror
17:14	RW	0x0	src_trans_en Source transparency enable bits [3]: A value stencil test enable bit [2]: B value stencil test enable bit [1]: G value stencil test enable bit [0]: R value stencil test enable bit
13	RW	0x0	src_trans_mode Source color key mode 0: normal stencil test 1: inverted stencil test
12:11	RW	0x0	src_yuv2rgb_mode Source bitmap YUV2RGB conversion mode 00: BT.601-MPEG 01: BT.601-JPEG 10: BT.709 11: BT.601-MPEG
10	RW	0x0	src_uv_swap Source Cb-Cr swap 0: CrCb 1: CbCr
9	RW	0x0	src_alpha_swap Source bitmap data alpha swap 0: ABGR 1: BGRA
8	RW	0x0	src_rb_swap Source bitmap data RB swap 0: BGR 1: RGB
7:4	RW	0x0	src_data_fmt Source bitmap data format 0000: XBGR888/ABGR888 0001: RGB565 0010: ARGB1555 0011: ARGB4444 0100: YUV422SP 0101: YUV422P 0110: YUV420SP 0111: YUV420P 1000: 1BPP (color palette) 1001: 2BPP (color palette) 1010: 4BPP (color palette) 1011: 8BPP (color palette)

Bit	Attr	Reset Value	Description
3	RW	0x0	src_rga_pack Source bitmap RGB packed 0: ABGR 1: BGR packed
2:0	RW	0x0	render_mode RGA 2D render mode 000: Bitblt 001: Color palette 010: Color fill (pattern fill) 011: Line/point drawing 100: Blur/sharp filter 101: Pre-scaling 110: Update palette LUT 111: Update pattern buffer

RGA_SRC_Y_MST

Address: Operational Base + offset (0x0104)

Source image Y/RGB/line drawing start addr

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_y_mst RGA source image Y/RGB start address register

RGA_SRC_CB_MST

Address: Operational Base + offset (0x0108)

Source image Cb/Cbr start addr

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_cb_mst RGA source image Cb/Cbr start address register source image Cb start address(YUV422/420-P); source image Cb/Cr start address(YUV422/420-SP); mask start address in ROP4 mode

RGA_SRC_CR_MST

Address: Operational Base + offset (0x010c)

Source image Cr/color palette start addr

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_cr_mst source image Cr start address(YUV422/420-P)

RGA_SRC_VIR_INFO

Address: Operational Base + offset (0x0110)

Source image virtual width

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:0	RW	0x0000	src_vir_stride source image virtual stride(words)

RGA_SRC_ACT_INFO

Address: Operational Base + offset (0x0114)

Source image active width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x0000	src_act_height source image active height
15:13	RO	0x0	reserved
12:0	RW	0x0000	src_act_width source image active width

RGA_SRC_X_PARA

Address: Operational Base + offset (0x0118)

Source image horizontal scaling/rotation parameter

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	src_h_para1 Source image horizontal scaling/rotation parameter1 $\sin(a)/ZoomX$ (signed 2.14)
15:0	RW	0x0000	src_h_para0 Source image horizontal scaling/rotation parameter0 $\cos(a)/ZoomX$ (signed 2.14)

RGA_SRC_Y_PARA

Address: Operational Base + offset (0x011c)

Source image vertical scaling/rotation parameter

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	src_v_para1 Source image vertical scaling/rotation parameter1 $\cos(a)/ZoomY$ (signed 2.14)
15:0	RW	0x0000	src_v_para0 Source image vertical scaling/rotation parameter0 $-\sin(a)/ZoomY$ (signed 2.14)

RGA_SRC_TILE_XINFO

Address: Operational Base + offset (0x0120)

Source tile start point coordinate, Source tile width

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	src_tile_width Source tile width (unsigned 5.11)
15:0	RW	0x0000	src_tile_xst Source tile start point x coordinate (signed13.3)

RGA_SRC_TILE_YINFO

Address: Operational Base + offset (0x0124)

Source tile start point coordinate, Source tile height

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	src_tile_height Source tile height (unsigned 5.11)
15:0	RW	0x0000	src_tile_yst Source tile start point y coordinate (signed13.3)

RGA_SRC_TILE_H_INCR

Address: Operational Base + offset (0x0128)

Source tile horizontal X/Y increment value

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	src_h_tile_y_incr Source horizontal tile Y increment value (signed 6.10)
15:0	RW	0x0000	src_h_tile_x_incr Source horizontal tile X increment value (signed 6.10)

RGA_SRC_TILE_V_INCR

Address: Operational Base + offset (0x012c)

Source tile vertical X/Y increment value

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	src_v_tile_y_incr Source vertical tile Y increment value (signed 6.10)
15:0	RW	0x0000	src_v_tile_x_incr Source vertical tile X increment value (signed 6.10)

RGA_SRC_TILE_OFFSETX

Address: Operational Base + offset (0x0130)

Source tile start point x for DST tile start point remap

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	src_tile_xoff Source tile start point offset X for DST tile start point remap(unsigned 5.14)

RGA_SRC_TILE_OFFSETY

Address: Operational Base + offset (0x0134)

Source tile start point y for DST tile start point remap

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RW	0x00000	src_tile_yoff Source tile start point offset Y for DST tile start point remap(unsigned 5.14)

RGA_SRC_BG_COLOR

Address: Operational Base + offset (0x0138)

Source image background color

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_bg_color Source image background color "0" bit color for mono expansion.

RGA_SRC_FG_COLOR

Address: Operational Base + offset (0x013c)

Source image foreground color

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_fg_color Source image foreground color "1" bit color for mono expansion. Line/point color, Color fill color, Pan color

RGA_SRC_TR_COLORO

Address: Operational Base + offset (0x0140)

Source image transparency color min value

Bit	Attr	Reset Value	Description
31:24	RW	0x00	src_trans_a_min source image transparency color A min value
23:16	RW	0x00	src_trans_b_min source image transparency color B min value

Bit	Attr	Reset Value	Description
15:8	RW	0x00	src_trans_g_min source image transparency color G min value
7:0	RW	0x00	src_trans_r_min source image transparency color R min value

RGA_CP_GR_A

Address: Operational Base + offset (0x0140)

RGA color gradient fill step register (color fill mode)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	a_gr_y Y gradient value of Alpha (signed 8.8)
15:0	RW	0x0000	a_gr_x X gradient value of Alpha (signed 8.8)

RGA_SRC_TR_COLOR1

Address: Operational Base + offset (0x0144)

Source image transparency color max value,Color gradient fill st

Bit	Attr	Reset Value	Description
31:24	RW	0x00	src_trans_a_max source image transparency color A max value
23:16	RW	0x00	src_trans_b_max source image transparency color B max value
15:8	RW	0x00	src_trans_g_max source image transparency color G max value
7:0	RW	0x00	src_trans_r_max source image transparency color R max value

RGA_CP_GR_B

Address: Operational Base + offset (0x0144)

RGA color gradient fill step register (color fill mode)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	b_gr_y Y gradient value of Blue (signed 8.8)
15:0	RW	0x0000	b_gr_x X gradient value of Blue (signed 8.8)

RGA_LINE_DRAW

Address: Operational Base + offset (0x0148)

Point/line drawing setting

Bit	Attr	Reset Value	Description
31	RW	0x0	line_draw_aa Line drawing Anti-alising operation 0: disable 1: enable
30	RW	0x0	line_draw_last_point_en Line drawing last point drawing 0: Don't draw 1: Draw
29	RW	0x0	line_draw_semi_dir Direction of semi-major axis 0: Increase 1: Decrease
28	RW	0x0	line_draw_major_dir Direction of major axis 0: Increase 1: Decrease
27:16	RW	0x000	line_draw_incr Line drawing X/Y delta step (unsigned 0.12) X delta step value if X is major axis; Y delta step value if Y is major axis;
15:12	RW	0x0	line_draw_width Line width (1~16 pixel)
11	RW	0x0	line_draw_dir Line drawing direction 0: X is major axis 1: Y is major axis
10:0	RW	0x000	line_draw_length Line drawing X/Y length of line (unigned 11) X length if X is major axis Y length if Y is major axis

RGA_PAT_ST_POINT

Address: Operational Base + offset (0x0148)

RGA pattern start point

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	pat_st_point Pattern start point in pattern ram

RGA_DST_MST

Address: Operational Base + offset (0x014c)

Destination image start addr

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_mst destiniation image RGB start address source image color palette table start address(color palette mode)

RGA_DST_VIR_INFO

Address: Operational Base + offset (0x0150)

Destination image virtual width

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	mask_vir_stride mask image virtual stride[6:0] (words) destination image virtual height[11:0] for line point drawing
15:12	RO	0x0	reserved
11:0	RW	0x000	dst_vir_stride destination image virtual stride(words)

RGA_DST_CTR_INFO

Address: Operational Base + offset (0x0154)

Destination image control window active width/height

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dst_ctrl_win_height destination image control window height (11bits) Start Y in line drawing mode (11bits) Pre_scaling active height in pre_scaling mode (12bits)
15:12	RO	0x0	reserved
11:0	RW	0x000	dst_ctrl_win_width destination image control window width (11bits) Start X in line drawing mode (11bits) Pre_scaling active width in pre_scaling mode (12bits)

RGA_ALPHA_CON

Address: Operational Base + offset (0x0158)

Alpha blending/ROP mode register

Bit	Attr	Reset Value	Description
31	RW	0x0	rataate_aa Rotation AA mode 0:no AA 1:AA

Bit	Attr	Reset Value	Description
30	RW	0x0	gr_cal_mode Gradient calculation mode 0:clip 1:not-clip
29	RW	0x0	dither_down_en Dither_down_en 0:disable 1:enable
28	RW	0x0	alpah_cal_sel Alpha_cal_sel 0:alpha' = alpha + (alpha>>7) 1:alpha' = alpha
27:26	RW	0x0	bs_filter_type Blur/sharp Filter type 00:weakest 01:weaker 10:stronger 11:strongest
25	RW	0x0	bs_mode Blur/sharp filter mode 0: Blur 1: sharp
24	RW	0x0	pre_scl_yuv_out_fmt Pre-scale YCbCr output format 0: The same with source format 1: all is semi-planar
23:22	RW	0x0	pre_scl_v_ratio Pre_scaler vertical scaling ratio: 00: 1 01: 1/2 10: 1/4 11: 1/8
21:20	RW	0x0	pre_scl_h_ratio Pre_scaler horizontal scaling ratio: 00: 1 01: 1/2 10: 1/4 11: 1/8
19:18	RW	0x0	rop_mode ROP mode select 00: ROP 2 01: ROP 3 10: ROP 4
17	RW	0x0	fading_en Fading enable 0: disable 1: enable
16	RW	0x0	mix_alpha_mode Mix alpha mode or porter-duff alpha mode

Bit	Attr	Reset Value	Description
15:8	RW	0x00	user_set_alpha User set alpha constant value/fading_alpha_value
7:4	RW	0x0	port_duff_mode Porter-duff mode 0: CLEAR 1: SRC 2: DST 3: SRC OVER 4: DST OVER 5: SRC IN 6: DST IN 7: SRC OUT 8: DST OUT 9: SRC ATOP 10: DST ATOP 11: XOR
3:2	RW	0x0	alpha_mode Per pixel alpha or user set alpha 00: user set alpha 01: per pixel alpha 10: per pixel alpha & user set alpha 11: un-defined
1	RW	0x0	alpha_rop_sel Alpha or ROP sel: 0: alpha 1: ROP
0	RW	0x0	alpha_rop_en Alpha or ROP enable 0: disable 1: enable

RGA_PAT_CON

Address: Operational Base + offset (0x015c)

Pattern size/offset

Bit	Attr	Reset Value	Description
31:24	RW	0x00	pat_yoff Pattern y offset
23:16	RW	0x00	pat_xoff Pattern x offset
15:8	RW	0x00	pat_height Pattern height
7:0	RW	0x00	pat_width Pattern width Pattern total number when doing pattern load

RGA_DST_VIR_WIDTH

Address: Operational Base + offset (0x015c)

Register0000 Abstract

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	dst_vir_width_pixel destination image virtual width(pixel)

RGA_ROP_CON0

Address: Operational Base + offset (0x0160)

Raster operation code0 control register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	rop3_code0 Rop3 code 0 control bits

RGA_CP_GR_G

Address: Operational Base + offset (0x0160)

Color gradient fill step of green

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	g_gr_y Y gradient value of Green (signed 8.8)
15:0	RW	0x0000	g_gr_x X gradient value of Green (signed 8.8)

RGA_PRESCL_CB_MST

Address: Operational Base + offset (0x0160)

RGA pre-scale Cb destination start addr

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pre_scl_uv_dst_mst Pre-scale Cb/Cr destination start addr

RGA_ROP_CON1

Address: Operational Base + offset (0x0164)

Raster operation code1 control register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	rop3_code1 Rop3 code 1 control bits

RGA_CP_GR_R

Address: Operational Base + offset (0x0164)

Color gradient fill step of red

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	r_gr_y X gradient value of Red (signed 8.8)
15:0	RW	0x0000	r_gr_x X gradient value of Red (signed 8.8)

RGA_PRESCL_CR_MST

Address: Operational Base + offset (0x0164)

RGA pre-scale Cr destination start addr

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pre_scl_v_dst_mst Pre-scale Cr destination start addr

RGA_MMU_CTRL

Address: Operational Base + offset (0x0168)

MMU control register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	fading_b Fading offset B value
23:16	RW	0x00	fading_g Fading offset G value
15:8	RW	0x00	fading_r Fading offset R value
7:6	RO	0x0	reserved
5:4	RW	0x0	mmu_page_table_size RGA MMU Page table size 00: 1KB page 01: 2KB page 10: 4KB page 11: 8KB page
3	RW	0x0	cmd_flush_en MMU TLB CMD channel flush enable bit (auto clear) 00: 1KB page 01: 2KB page 10: 4KB page 11: 8KB page
2	RW	0x0	dst_flush_en MMU TLB DST channel flush enable bit (auto clear) 0: no flush 1: flush
1	RW	0x0	src_flush_en MMU TLB SRC channel flush enable bit (auto clear) 0: no flush 1: flush

Bit	Attr	Reset Value	Description
0	RW	0x0	mmu_en RGA MMU enable

RGA_MMU_TLB

Address: Operational Base + offset (0x016c)

RGA MMU TLB base address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mmu_tlb_addr RGA MMU TLB base address(word)

RGA_YUV_OUT_CFG

Address: Operational Base + offset (0x170)

RGA Destination output configuration

bit	Attr	Reset Value	Description
31:15	RO	0x0	Reserved
14:8	RW	0x0	sw_axi_rw_align_e_n AXI WR/RD Align in line scan; Suggested value=7'b0, Align enable; [6:0] each bit value: 0, enable; 1: disable; [6]: DST BGR565/4444/1555 Read align disable [5]: DST ABGR888 Read align disable [4]: DST BGR565/4444/1555 Write align disable [3]: DST ABGR888 Write align disable [2]: SRC Y channel read Ralign disable [1]: SRC BGR565/4444/1555 Read align disable [0]: SRC ABGR888 Read align disable
7	RW	0x0	sw_cp_alpha_bit_sel alpha bits width select for ColorPalette; default=1'b0; 1: 1bit alpha; 0: 8bit alpha;
6	RW	0x0	sw_dst_csc_clip BGR2YUV Clip mode(from 0~255 clip to 36~235) 1: clip enable; 0: unclip
5:4	RW	0x1	sw_dst_csc_mode DST bitmap RGB2YUV conversion mode 00: Bypass mode, only used in y2y mode 01: BT.601-range0 10: BT.601-range1 11: BT.709-range0
3	RW	0x0	sw_dst_uv_swap Destination Cb-Cr swap 0: CrCb 1: CbCr
2	RW	0x0	reserved
1	RW	0x0	sw_dst_yuv_fmt Destination bitmap data format 0: YUV422SP 1: YUV420SP
0	RW	0x0	sw_dst_yuv_en Destination YUV output enable, Could not used

			Pre_scaling/Line-point drawing and Blur/Sharp, Alpha and ROP. 1: enable; 0: disable;
--	--	--	--

RGA_DST_UV_MST

Address: Operational Base + offset (0x174)

RGA Destination image UV start addr

bit	Attr	Reset Value	Description
31:0	RW	0x0	sw_dst_uv_mst destination image UV start address

19.5 Programming Guide

19.5.1 Register Partition

There are two types of register in RGA. The first 8 registers (0x0 - 0x1C) are general registers for system configuration including command mode, command parameter, RGA status, general interrupts. The other registers (range from 0x100~0x178) are command registers for command codes.

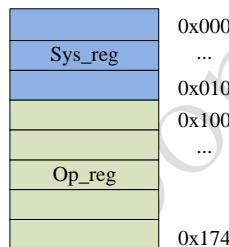


Fig. 19-8 HDMI TX Software Main Sequence Diagram

19.5.2 Command Modes

RGA has two command modes: slave mode and master mode. In slave mode (RGA_SYS_CTRL[2] = 1'b0), 2D graphic command only could be run one by one. CPU set all the command registers in RGA and then start RGA running by setting RGA_SYS_CTRL[2] to '1'. In master mode (RGA_SYS_CTRL[2] = 1'b1), 2D graphic commands could be run sequentially. After setting command's number to RGA_CMD_CTRL[12:3], writing '1' to RGA_CMD_CTRL[0] will start the command fetch, then Internal command DMA fetch commands from external command line.

Command line is a collection of several command codes with continuous address. At the first start, the command start address (RGA_CMD_ADDR) and command number (RGA_CMD_CTRL[12:3]) should be set, then write '1' to cmd_line_st (RGA_CMD_CTRL[0]) to start the command line fetch. Incremental command is supported by setting cmd_incr_num (RGA_CMD_CTRL[12:3]) and cmd_incr_valid (RGA_CMD_CTRL[1]=1'b1)

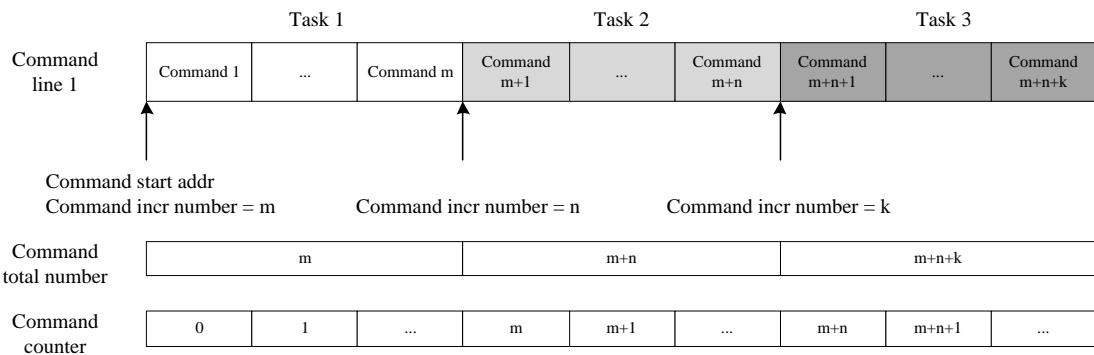


Fig. 19-9 RGA command line and command counter

19.5.3 Command Sync

In slave command mode, command sync is controlled by CPU.

In master command mode, user can enable the current_cmd_int (RGA_MODE_CTRL[25] = 1'b1) command by command to generate a interrupt at the end point of target command operation.

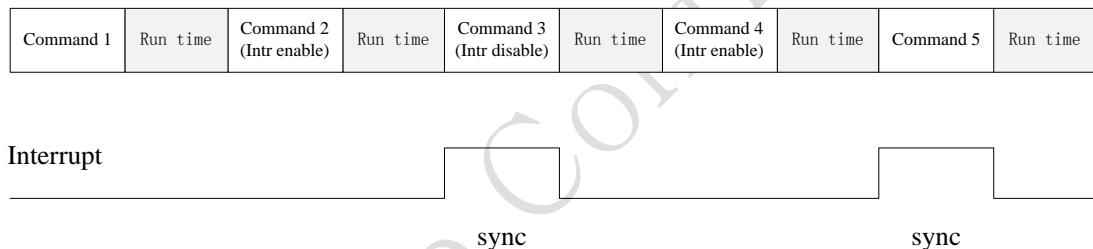


Fig. 19-10 RGA command sync generation

Rockchip Confidential

Chapter 20 Image Enhancement Processor (IEP)

20.1 Overview

The Image Enhancement Processor (IEP) receives data from and transmits data to system main memory by AXI bus, or output the data to Video Output Processor (VOP) directly.

The features of IEP are as follow:

- **Image format**
 - Input data: XRGB/RGB565/YUV420/YUV422
 - Output data: ARGB/RGB565/YUV420/YUV422
 - ARGB/XRGB/RGB565/YUV swap
 - YUV semi-planar/planar
 - BT601_I/BT601_f/BT709_I/BT709_f color space conversion
 - RGB dither up/down conversion
 - YUV up/down sampling conversion
 - Max resolution for static image up to 8192x8192
 - Max resolution for dynamic image up to 1920x1080
- **Enhancement**
 - Gamma adjustment with programmable mapping table
 - Hue/Saturation/Brightness/Contrast enhancement
 - Color enhancement with programmable coefficient
 - Detail enhancement with filter matrix up to 5x5
 - Edge enhancement with filter matrix up to 5x5
 - Programmable difference table for detail enhancement
 - Programmable distance table for detail and edge enhancement
- **Noise reduction**
 - Compression noise reduction with filter matrix up to 5x5
 - Programmable difference table for compression noise reduction
 - Programmable distance table for compression noise reduction
- **De-interlace**
 - Input 4 fields, output 2 frames mode
 - Input 4 fields, output 1 frames mode
 - Input 2 fields, output 1 frames mode
 - Programmable motion detection coefficient
 - Programmable high frequency factor
 - Programmable edge interpolation parameter
- **Interface**
 - Programmable direct path to VOP
 - 32bit AHB bus slave
 - 64bit AXI bus master
 - Combined interrupt output

20.2 Block Diagram

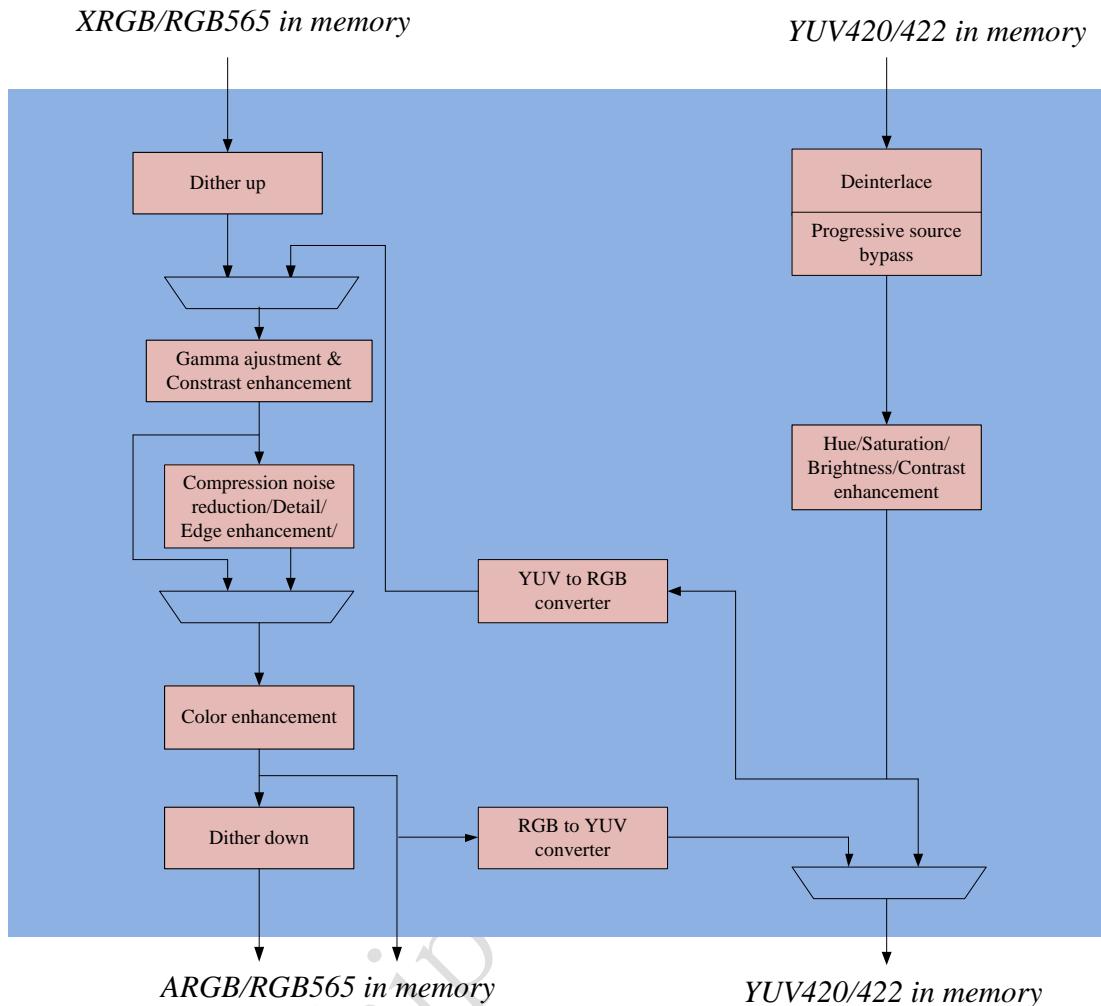


Fig. 20-1 IEP block diagram

The data path in IEP is in the previously diagram. The IEP comprises with:

- Deinterlace

There are five deinterlace mode including I4O2 (input 4 fields and output 2 frames once), I4O1B, I4O1T, I2O1B, I2O1T in the deinterlace block. YUV bypass is also supported. Pay attention if compression noise reduction, detail or edge enhancement work together with deinterlace is not allowed.

- Enhancement

Not only hue, saturation, brightness, contrast enhancement, but also blue screen, black screen and color bar are supported in YUV domain enhancement block. Gamma adjustment, edge enhancement, detail enhancement and color enhancement are supported in RGB domain enhancement block.

- Noise Reduction

Spatial and temporal sampling noise can be reduced in YUV domain noise reduction block. Compression noise can be reduced in RGB domain noise reduction block.

20.3 Function description

20.3.1 Deinterlace

There are five deinterlace mode including I4O2, I4O1B, I4O1T, I2O1B and I2O1T in the deinterlace block. The I4O2 mode represents for 4 fields of input images and 2 frames of output images, so all of the two groups of source address registers and two groups of destination address registers need to be configured. For example, if source and destination format are both YUV420, the source address register IEP_SRC_ADDR_YRGB, IEP_SRC_ADDR_CBCR are used for source field0 and field 1, the source address register IEP_SRC_ADDR_Y1, IEP_SRC_ADDR_CBCR1 are used for source field2 and field3. The I4O1B and I4O1T mode have the same input images as the I4O2 mode, but only one frame output is generated once. The I2O1B and I2O1T mode have the same output as I4O1B and I4O1T mode, but only two fields input are needed. If bypass mode is selected, there are not any deinterlace operations. The parameter dil_ei_sel, dil_ei_radius, dil_ei_smooth, dil_ei_mode, dil_hf_en and dil_hf_fct in register IEP_CONFIG0 and registers IEP_DIL_MTN_TAB0~7 may have different influence in deinterlace effect depend on the type of the image source.

20.3.2 Noise reduction

Compression noise reduction is used for reducing the noise after the decompression of picture or video. Before the compression noise reduction is enabled, the IEP_ENH_DDE_COE0/1 from address 0x400 to 0x5FC for difference and distance coefficients must be written firstly. The filter matrix can be selected from 3x3/5x5 and the filter weight can be programmed by configuring IEP_ENH_RGB_CNFG.

20.3.3 Enhancement

Not only hue, saturation, brightness, contrast enhancement, but also blue screen, black screen and color bar are supported in this block. IEP_ENH_YUV_CNFG_0/1/2 registers can be configured to modify the YUV enhance parameters to satisfied with the requirement.

Before the gamma adjustment or contrast enhancement is enabled in RGB domain, the IEP_ENH_CG_TAB from address 0x100 to 0x3FC for B, G, R mapping must be written firstly. If the color enhancement is enabled, the IEP_ENH_C_COE must be written the required value.

Before the edge or detail enhancement is enabled, the IEP_ENH_DDE_COE0/1 from address 0x400 to 0x5FC for difference and distance coefficients must be written firstly. The filter matrix can be selected from 3x3/5x5 and the filter weight can be programmed by configuring IEP_ENH_RGB_CNFG.

20.3.4 Format conversion

The color space conversion either from RGB to YUV or from YUV to RGB has the selections including BT601/709_L/F mode, and the input can be clipped or not.

If the source format is RGB565, dither up must be enabled. In contrary to the destination format is RGB565, dither down must be enabled.

20.3.5 Shadow registers

The configuration registers can be configured at any time, but they cannot have any effect immediately unless config_done is available and a new frame_start is enabled. The registers IEP_RAW_CONFIG0/1, IEP_RAW_VIR_IMG_WIDTH, IEP_RAW_IMG_SCL_FCT, IEP_RAW_SRC_IMG_SIZE, IEP_RAW_ENH_YUV_CNFG_0/1/2 corresponding to the registers have the similar names but without letters _RAW. They are used for raw register value reading before the configurations really have effect on the new frame.

20.3.6 VOP direct path

The IEP_DST_ADDR for DMA writing is useless if vop_path_en bit is set, because all RGB or YUV data is supplied for VOP directly from local bus via VOP and IEP.

20.4 Register description

20.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
IEP_CONFIG0	0x00000	W	0x00000000	configuration register0
IEP_CONFIG1	0x00004	W	0x00000000	configuration register1
IEP_STATUS	0x00008	W	0x00000000	status register
IEP_INT	0x0000c	W	0x00000000	interrupt register
IEP_FRM_START	0x00010	W	0x00000000	frame start
IEP_CONFIG_DONE	0x00018	W	0x00000000	configuration done
IEP_FRM_CNT	0x0001c	W	0x00000000	frame counter
IEP_VIR_IMG_WIDT H	0x00020	W	0x01400140	Image virtual width
IEP_SRC_IMG_SIZE	0x00028	W	0x00f00140	Source image width/height
IEP_DST_IMG_SIZE	0x0002c	W	0x00f00140	Destination image width/height
IEP_DST_IMG_WIDT H_TILE0	0x00030	W	0x00000000	Destination image tile0 width
IEP_DST_IMG_WIDT H_TILE1	0x00034	W	0x00000000	Destination image tile1 width
IEP_DST_IMG_WIDT H_TILE2	0x00038	W	0x00000000	Destination image tile2 width
IEP_DST_IMG_WIDT H_TILE3	0x0003c	W	0x00000000	Destination image tile3 width
IEP_ENH_YUV_CNFG_0	0x00040	W	0x00000000	brightness,contrast,saturation adjustment
IEP_ENH_YUV_CNFG_1	0x00044	W	0x00000000	Hue configuration
IEP_ENH_YUV_CNFG_2	0x00048	W	0x00000000	color bar configuration
IEP_ENH_RGB_CNFG	0x0004c	W	0x00000000	enhancement RGB configuration
IEP_ENH_C_COE	0x00050	W	0x00000000	rgb color enhancement coefficient

Name	Offset	Size	Reset Value	Description
IEP_RAW_CONFIG0	0x00058	W	0x00000000	configuration register0
IEP_RAW_CONFIG1	0x0005c	W	0x00000000	configuration register1
IEP_RAW_VIR_IMG_WIDTH	0x00060	W	0x1400140	Image virtual width
IEP_RAW_SRC_IMG_SIZE	0x00068	W	0x0f00140	Source image width/height
IEP_RAW_DST_IMG_SIZE	0x0006c	W	0x0f00140	Destination image width/height
IEP_RAW_ENH_YUV_CNFG_0	0x00070	W	0x00000000	brightness,contrast,saturation adjustment
IEP_RAW_ENH_YUV_CNFG_1	0x00074	W	0x00000000	Hue configuration
IEP_RAW_ENH_YUV_CNFG_2	0x00078	W	0x00000000	color bar configuration
IEP_RAW_ENH_RGB_CNFG	0x0007c	W	0x00000000	enhancement RGB configuration
IEP_SRC_ADDR_YRB	0x00080	W	0x00000000	Start address of source image(Y/RGB)
IEP_SRC_ADDR_CBCR	0x00084	W	0x00000000	Start address of source image(Cb/Cr)
IEP_SRC_ADDR_CR	0x00088	W	0x00000000	Start address of source image(Cr)
IEP_SRC_ADDR_Y1	0x0008c	W	0x00000000	Start address of source image(Y)
IEP_SRC_ADDR_CBCR1	0x00090	W	0x00000000	Start address of source image(Cb/Cr)
IEP_SRC_ADDR_CR1	0x00094	W	0x00000000	Start address of source image(Cr)
IEP_DST_ADDR_YRB	0x000b0	W	0x00000000	Start address of destination image(Y/RGB)
IEP_DST_ADDR_CBCR	0x000b4	W	0x00000000	Start address of destination image(Cb/Cr)
IEP_DST_ADDR_CR	0x000b8	W	0x00000000	Start address of destination image(Cr)
IEP_DST_ADDR_Y1	0x000bc	W	0x00000000	Start address of destination image(Y)
IEP_DST_ADDR_CBCR1	0x000c0	W	0x00000000	Start address of destination image(Cb/Cr)
IEP_DST_ADDR_CR1	0x000c4	W	0x00000000	Start address of destination image(Cr)
IEP_DIL_MTN_TAB0	0x000e0	W	0x00000000	Deinterlace motion table0
IEP_DIL_MTN_TAB1	0x000e4	W	0x00000000	Deinterlace motion table1
IEP_DIL_MTN_TAB2	0x000e8	W	0x00000000	Deinterlace motion table2
IEP_DIL_MTN_TAB3	0x000ec	W	0x00000000	Deinterlace motion table3
IEP_DIL_MTN_TAB4	0x000f0	W	0x00000000	Deinterlace motion table4

Name	Offset	Size	Reset Value	Description
IEP_DIL_MTN_TAB5	0x000f4	W	0x00000000	Deinterlace motion table5
IEP_DIL_MTN_TAB6	0x000f8	W	0x00000000	Deinterlace motion table6
IEP_DIL_MTN_TAB7	0x000fc	W	0x00000000	Deinterlace motion table7
IEP_ENH(CG)_TAB	0x00100	W	0x00000000	contrast and gamma enhancement table
IEP_ENH(DDE)_COE0	0x00400	W	0x00000000	denoise,detail and edge enhancement coefficient
IEP_ENH(DDE)_COE1	0x00500	W	0x00000000	denoise,detail and edge enhancement coefficient
IEP_MMU_DTE_ADDR	0x00800	W	0x00000000	MMU current page table address
IEP_MMU_STATUS	0x00804	W	0x00000018	MMU status register
IEP_MMU_CMD	0x00808	W	0x00000000	MMU command register
IEP_MMU_PAGE_FAULT_ADDR	0x0080c	W	0x00000000	MMU logic address of last page fault
IEP_MMU_ZAP_ONE_LINE	0x00810	W	0x00000000	MMU zap cache line register
IEP_MMU_INT_RAW_STAT	0x00814	W	0x00000000	MMU raw interrupt status register
IEP_MMU_INT_CLEAR	0x00818	W	0x00000000	MMU interrupt clear register
IEP_MMU_INT_MASK	0x0081c	W	0x00000000	MMU interrupt mask register
IEP_MMU_INT_STATUS	0x00820	W	0x00000000	MMU interrupt status register
IEP_MMU_AUTO_GATING	0x00824	W	0x00000001	MMU clock auto gating register

Notes: *Size* : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** - WORD (32 bits) access

20.4.2 Detail Register Description

IEP_CONFIG0

Address: Operational Base + offset (0x00000)
configuration register0

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RW	0x0	dil_ei_sel deinterlace edge interpolation select
22:21	RW	0x0	dil_ei_radius deinterlace edge interpolation radius

Bit	Attr	Reset Value	Description
20	RW	0x0	rgb_con_gam_order RGB contrast enhancement and gamma adjustment operation order select. 0:CG prior to DDE 1:DDE prior to CG (CG represent for contrast & gamma operation, and DDE represent for denoise, detail or edge enhancement operation)
19:18	RW	0x0	rgb_enh_sel RGB enhancement select 00: no operation 01: denoise 10: detail enhancement 11: edge enhancement
17	RW	0x0	rgb_con_gam_en RGB contrast enhancement and gamma adjustment enable 0:disable 1:enable
16	RW	0x0	rgb_color_enh_en RGB color enhancement enable 0:disable 1:enable
15	RW	0x0	dil_ei_smooth deinterlace edge interpolation for smooth effect 0: disable 1: enable
14	RW	0x0	yuv_enh_en yuv enhancement enable 0:disable 1:enable
13	RO	0x0	reserved
12	RW	0x0	dil_ei_mode deinterlace edge interpolation 0: disable 1: enable
11	RW	0x0	dil_hf_en deinterlace high frequency calculation enable 0: disable 1: enable
10:8	RW	0x0	dil_mode Deinterlace mode select: 000: YUV deinterlace and bypass path disable; 001: I4O2 mode 010: I4O1B mode 011: I4O1T mode 100: I2O1B mode 101: I2O1T mode 110: bypass mode

Bit	Attr	Reset Value	Description
7:1	RW	0x00	dil_hf_fct deinterlace high frequency factor
0	RW	0x0	vop_path_en VOP direct path enable 0:disable 1:enable

IEP_CONFIG1

Address: Operational Base + offset (0x00004)

configuration register1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	glb_alpha global alpha value only valid when destination format is ARGB
23	RW	0x0	rgb2yuv_input_clip RGB to YUV input range 0:R/G/B=[0,255] 1:R/G/B=[16,235]
22	RW	0x0	yuv2rgb_input_clip YUV to RGB input range 0:Y/U/V=[0,255] 1:Y=[16,235],U/V=[16,240]
21	RW	0x0	rgb_to_yuv_en RGB to YUV conversion enable 0: disable 1: enable
20	RW	0x0	yuv_to_rgb_en YUV to RGB conversion enable 0: disable 1: enable
19:18	RW	0x0	rgb2yuv_coe_sel rgb2yuv coefficient select 00:bt601_1 01:bt601_f 10:bt709_1 11:bt709_f
17:16	RW	0x0	yuv2rgb_coe_sel yuv2rgb coefficient select 00:bt601_1 01:bt601_f 10:bt709_1 11:bt709_f
15	RW	0x0	dthr_down_en dither down enable 0: disable 1: enable
14	RW	0x0	dthr_up_en dither up enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
13:12	RW	0x0	dst_yuv_swap destination YUV swap 00:SP UV 01:SP VU 10, 11:P
11:10	RW	0x0	dst_rgb_swap destination RGB swap ARGB destination 00:ARGB 01:ABGR 10:RGBA 11:BGRA RGB565 destination 00,10:RGB 01,11:BGR
9:8	RW	0x0	dst_fmt Output image Format 00 : ARGB 01 : RGB565 10 : YUV422 11 : YUV420
7:6	RO	0x0	reserved
5:4	RW	0x0	src_yuv_swap source YUV swap 00:SP UV 01:SP VU 10, 11:P
3:2	RW	0x0	src_rgb_swap source RGB swap XRGB source 00:XRGB 01:XBGR 10:RGBX 11:BGRX RGB565 source 00,10:RGB 01,11:BGR
1:0	RW	0x0	src_fmt Input image Format 00 : XRGB 01 : RGB565 10 : YUV422 11 : YUV420

IEP_STATUS

Address: Operational Base + offset (0x00008)
status register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved

Bit	Attr	Reset Value	Description
19	RW	0x0	rrgb_idle_ack RGB read DMA idle acknowlege
18	RW	0x0	wrgb_idle_ack RGB write DMA idle acknowlege
17	RW	0x0	ryuv_idle_ack YUV read DMA idle acknowlege
16	RW	0x0	wyuv_idle_ack YUV write DMA idle acknowlege
15:9	RO	0x0	reserved
8	RO	0x0	voi_sts vop direct path status 00:idle 01:working
7	RO	0x0	rrgb_sts RGB DMA read status 00:idle 01:working
6	RO	0x0	wrgb_sts RGB DMA write status 00:idle 01:working
5	RO	0x0	ryuv_sts YUV DMA read status 00:idle 01:working
4	RO	0x0	wyuv_sts YUV DMA write status 00:idle 01:working
3	RO	0x0	dde_sts RGB denoise/enhancement status 00:idle 01:working
2	RO	0x0	dil_sts de-interlace or yuv bypass status 00:idle 01:working
1	RO	0x0	scl_sts scaling status 00:idle 01:working
0	RO	0x0	dns_sts YUV 3D denoise status 00:idle 01:working

IEP_INT

Address: Operational Base + offset (0x0000c)
 interrupt register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	W1C	0x0	frm_done_int_clr Frame process done interrupt clear After be set to 1, this bit will be clear automatically.
15:9	RO	0x0	reserved
8	RW	0x0	frm_done_int_en Frame process done interrupt enable: 0: disable; 1: enable;
7:1	RO	0x0	reserved
0	RO	0x0	frm_done_int Frame process done interrupt 0: inactive; 1: active;

IEP_FRM_START

Address: Operational Base + offset (0x00010)
 frame start

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1C	0x0	frm_start frame start Write 1, self clear.

IEP_CONFIG_DONE

Address: Operational Base + offset (0x00018)
 configuration done

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	config_done configuration done Wait for frame start to update raw register configuration to really used registers.

IEP_FRM_CNT

Address: Operational Base + offset (0x0001c)
 frame counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	frm_cnt frame counter Self increase one after a frame operation is finished. Write arbitrary value to clear to zero.

IEP_VIR_IMG_WIDTH

Address: Operational Base + offset (0x00020)

Image virtual width

Bit	Attr	Reset Value	Description
31:16	RW	0x0140	dst_vir_image_width Destination virtual image width
15:0	RW	0x0140	src_vir_image_width Source virtual image width

IEP_SRC_IMG_SIZE

Address: Operational Base + offset (0x00028)

Source image width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x00f0	src_image_height source image height
15:13	RO	0x0	reserved
12:0	RW	0x0140	src_image_width source image width

IEP_DST_IMG_SIZE

Address: Operational Base + offset (0x0002c)

Destination image width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x00f0	dst_image_height Destination image height
15:13	RO	0x0	reserved
12:0	RW	0x0140	dst_image_width Destination image width

IEP_DST_IMG_WIDTH_TILE0

Address: Operational Base + offset (0x00030)

Destination image tile0 width

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	dst_image_width_tile0 Destination image tile0 width

IEP_DST_IMG_WIDTH_TILE1

Address: Operational Base + offset (0x00034)

Destination image tile1 width

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:0	RW	0x000	dst_image_width_tile1 Destination image tile1 width

IEP_DST_IMG_WIDTH_TILE2

Address: Operational Base + offset (0x00038)

Destination image tile2 width

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:0	RW	0x000	dst_image_width_tile2 Destination image tile2 width

IEP_DST_IMG_WIDTH_TILE3

Address: Operational Base + offset (0x0003c)

Destination image tile3 width

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:0	RW	0x000	dst_image_width_tile3 Destination image tile3 width

IEP_ENH_YUV_CNF0

Address: Operational Base + offset (0x00040)

brightness,contrast,saturation adjustment

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:16	RW	0x000	sat_con YUV saturation and contrast adjustment saturation * contrast range from 0 to 1.992*1.992, and this value is saturation* contrast * 128
15:8	RW	0x00	contrast YUV contrast adjustment contrast value range from 0 to 1.992, and this value is contrast*128.
7:6	RO	0x0	reserved
5:0	RW	0x00	brightness YUV brightness adjustment range from -32 to 31 000000:0; 000001:1; 011111:31; 100000:-32; 100001:-31; 111110:-2; 111111:-1;

IEP_ENH_YUV_CNFG_1

Address: Operational Base + offset (0x00044)

Hue configuration

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	cos_hue the cos function value for hue adjustment sin function value range from 0.866 to 1 ,and this value is cos * 128 ,no sign bit
7:0	RW	0x00	sin_hue the sin function value for hue adjustment sin function value range from -0.5 to 0.5 ,and this value is sin * 128 ,and the high bit is sign bit

IEP_ENH_YUV_CNFG_2

Address: Operational Base + offset (0x00048)

color bar configuration

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:24	RW	0x0	video_mode video mode 00:black screen 01:blue screen 10:color bars 11:normal video
23:16	RW	0x00	color_bar_v color bar v value
15:8	RW	0x00	color_bar_u color bar u value
7:0	RW	0x00	color_bar_y color bar y value

IEP_ENH_RGB_CNFG

Address: Operational Base + offset (0x0004c)

enhancement RGB configuration

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	enh_threshold enhancement threshold In denoise and detail enhancement operation, more than the threshold, considering as detail; but if less than the threshold, considering as noise, need to be filtered.
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:8	RW	0x00	enh_alpha enhancement alpha value 0000000:0 0000001:1/16 0000010:2/16 0001111:15/16 0010000:1 0010001:1+1/16; 0010010:1+2/16; 0010011:1+3/16; 0100000:2; 0110000:3; 1000000:4; 1010000:5; 1100000:6; other : reserved
7:2	RO	0x0	reserved
1:0	RW	0x0	enh_radius enhancement radius 00:R=1 01:R=2 10:R=3 11:R=4

IEP_ENH_C_COE

Address: Operational Base + offset (0x00050)

rgb color enhancement coefficient

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:5	RW	0x0	c_int_coe color enhancement integer coefficient
4:0	RW	0x00	c_frac_coe color enhancement fraction coefficient

IEP_RAW_CONFIG0

Address: Operational Base + offset (0x00058)

configuration register0

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RW	0x0	dil_ei_sel deinterlace edge interpolation select

Bit	Attr	Reset Value	Description
22:21	RW	0x0	dil_ei_radius deinterlace edge interpolation radius
20	RW	0x0	rgb_con_gam_order RGB contrast enhancement and gamma adjustment operation order select. 0:CG prior to DDE 1:DDE prior to CG (CG represent for contrast & gamma operation, and DDE represent for denoise, detail or edge enhancement operation)
19:18	RW	0x0	rgb_enh_sel RGB enhancement select 00: no operation 01: denoise 10: detail enhancement 11: edge enhancement
17	RW	0x0	rgb_con_gam_en RGB contrast enhancement and gamma adjustment enable 0:disable 1:enable
16	RW	0x0	rgb_color_enh_en RGB color enhancement enable 0:disable 1:enable
15	RW	0x0	dil_ei_smooth deinterlace edge interpolation for smooth effect 0: disable 1: enable
14	RW	0x0	yuv_enh_en yuv enhancement enable 0:disable 1:enable
13	RO	0x0	reserved
12	RW	0x0	dil_ei_mode deinterlace edge interpolation 0: disable 1: enable
11	RW	0x0	dil_hf_en deinterlace high frequency calculation enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
10:8	RW	0x0	dil_mode Deinterlace mode select: 000: YUV deinterlace and bypass path disable; 001: I4O2 mode 010: I4O1B mode 011: I4O1T mode 100: I2O1B mode 101: I2O1T mode 110: bypass mode
7:1	RW	0x00	dil_hf_fct deinterlace high frequency factor
0	RW	0x0	vop_path_en VOP direct path enable 0:disable 1:enable

IEP_RAW_CONFIG1

Address: Operational Base + offset (0x0005c)

configuration register1

Bit	Attr	Reset Value	Description
31:24	RO	0x00	glb_alpha global alpha value only valid when destination format is ARGB
23	RO	0x0	rgb2yuv_input_clip RGB to YUV input range 0:R/G/B=[0,255] 1:R/G/B=[16,235]
22	RO	0x0	yuv2rgb_input_clip YUV to RGB input range 0:Y/U/V=[0,255] 1:Y=[16,235],U/V=[16,240]
21	RO	0x0	rgb_to_yuv_en RGB to YUV conversion enable 0: disable 1: enable
20	RO	0x0	yuv_to_rgb_en YUV to RGB conversion enable 0: disable 1: enable
19:18	RO	0x0	rgb2yuv_coe_sel rgb2yuv coefficient select 00:bt601_1 01:bt601_f 10:bt709_1 11:bt709_f

Bit	Attr	Reset Value	Description
17:16	RO	0x0	yuv2rgb_coe_sel yuv2rgb coefficient select 00:bt601_1 01:bt601_f 10:bt709_1 11:bt709_f
15	RO	0x0	dthr_down_en dither down enable 0: disable 1: enable
14	RO	0x0	dthr_up_en dither up enable 0: disable 1: enable
13:12	RO	0x0	dst_yuv_swap destination YUV swap 00:SP UV 01:SP VU 10, 11:P
11:10	RO	0x0	dst_rgb_swap destination RGB swap ARGB destination 00:ARGB 01:ABGR 10:RGBA 11:BGRA RGB565 destination 00,10:RGB 01,11:BGR
9:8	RO	0x0	dst_fmt Output image Format 00 : ARGB 01 : RGB565 10 : YUV422 11 : YUV420
7:6	RO	0x0	reserved
5:4	RO	0x0	src_yuv_swap source YUV swap 00:SP UV 01:SP VU 10, 11:P
3:2	RO	0x0	src_rgb_swap source RGB swap XRGB source 00:XRGB 01:XBGR 10:RGBX 11:BGRX RGB565 source 00,10:RGB 01,11:BGR

Bit	Attr	Reset Value	Description
1:0	RO	0x0	src_fmt Input image Format 00 : XRGB 01 : RGB565 10 : YUV422 11 : YUV420

IEP_RAW_VIR_IMG_WIDTH

Address: Operational Base + offset (0x00060)

Image virtual width

Bit	Attr	Reset Value	Description
31:16	RO	0x0140	dst_vir_image_width Destination virtual image width
15:0	RO	0x0140	src_vir_image_width Source virtual image width

IEP_RAW_SRC_IMG_SIZE

Address: Operational Base + offset (0x00068)

Source image width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RO	0x00f0	src_image_height source image height
15:13	RO	0x0	reserved
12:0	RO	0x0140	src_image_width source image width

IEP_RAW_DST_IMG_SIZE

Address: Operational Base + offset (0x0006c)

Destination image width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RO	0x00f0	dst_image_height Destination image height
15:13	RO	0x0	reserved
12:0	RO	0x0140	dst_image_width Destination image width

IEP_RAW_ENH_YUV_CNFG_0

Address: Operational Base + offset (0x00070)

brightness,contrast,saturation adjustment

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:16	RO	0x000	sat_con YUV saturation and contrast adjustment saturation * contrast range from 0 to 1.992*1.992, and this value is saturation* contrast * 128
15:8	RO	0x00	contrast YUV contrast adjustment contrast value range from 0 to 1.992, and this value is contrast*128.
7:6	RO	0x0	reserved
5:0	RO	0x00	brightness YUV brightness adjustment range from -32 to 31 000000:0; 000001:1; 011111:31; 100000:-32; 100001:-31; 111110:-2; 111111:-1;

IEP_RAW_ENH_YUV_CNFG_1

Address: Operational Base + offset (0x00074)

Hue configuration

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RO	0x00	cos_hue the cos function value for hue adjustment sin function value range from 0.866 to 1 ,and this value is cos * 128 ,no sign bit
7:0	RO	0x00	sin_hue the sin function value for hue adjustment sin function value range from -0.5 to 0.5 ,and this value is sin * 128 ,and the high bit is sign bit

IEP_RAW_ENH_YUV_CNFG_2

Address: Operational Base + offset (0x00078)

color bar configuration

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:24	RO	0x0	video_mode video mode 00:black screen 01:blue screen 10:color bars 11:normal video

Bit	Attr	Reset Value	Description
23:16	RO	0x00	color_bar_v color bar v value
15:8	RO	0x00	color_bar_u color bar u value
7:0	RO	0x00	color_bar_y color bar y value

IEP_RAW_ENH_RGB_CNFG

Address: Operational Base + offset (0x0007c)

enhancement RGB configuration

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	enh_threshold enhancement threshold In denoise and detail enhancement operation, more than the threshold, considering as detail; but if less than the threshold, considering as noise, need to be filtered.
15	RO	0x0	reserved
14:8	RW	0x00	enh_alpha enhancement alpha value 0000000:0 0000001:1/16 0000010:2/16 0001111:15/16 0010000:1 0010001:1+1/16; 0010010:1+2/16; 0010011:1+3/16; 0100000:2; 0110000:3; 1000000:4; 1010000:5; 1100000:6; other : reserved
7:2	RO	0x0	reserved
1:0	RW	0x0	enh_radius enhancement radius 00:R=1 01:R=2 10:R=3 11:R=4

IEP_SRC_ADDR_YRGB

Address: Operational Base + offset (0x00080)

Start address of source image(Y/RGB)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_image_yrgb_mst Source image data YRGB start address in Memory

IEP_SRC_ADDR_CBCR

Address: Operational Base + offset (0x00084)

Start address of source image(Cb/Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_image_cbcr_mst Source image data CbCr start address in Memory

IEP_SRC_ADDR_CR

Address: Operational Base + offset (0x00088)

Start address of source image(Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_image_cr_mst Source image data Cr start address in Memory

IEP_SRC_ADDR_Y1

Address: Operational Base + offset (0x0008c)

Start address of source image(Y)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_image_y_mst Source image data Y start address in Memory

IEP_SRC_ADDR_CBCR1

Address: Operational Base + offset (0x00090)

Start address of source image(Cb/Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_image_cbcr_mst Source image data CbCr start address in Memory

IEP_SRC_ADDR_CR1

Address: Operational Base + offset (0x00094)

Start address of source image(Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_image_cr_mst Source image data Cr start address in Memory

IEP_DST_ADDR_YRGB

Address: Operational Base + offset (0x000b0)

Start address of destination image(Y/RGB)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_yrgb_mst Destination image data YRGB start address in Memory

IEP_DST_ADDR_CBCR

Address: Operational Base + offset (0x000b4)

Start address of destination image(Cb/Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_cbcu_mst Destination image data CBCR start address in Memory

IEP_DST_ADDR_CR

Address: Operational Base + offset (0x000b8)

Start address of destination image(Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_cr_mst Destination image data CR start address in Memory

IEP_DST_ADDR_Y1

Address: Operational Base + offset (0x000bc)

Start address of destination image(Y)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_y_mst Destination image data Y start address in Memory

IEP_DST_ADDR_CBCR1

Address: Operational Base + offset (0x000c0)

Start address of destination image(Cb/Cr)

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_cbcr_mst Destination image data CbCr start address in Memory

IEP_DST_ADDR_CR1

Address: Operational Base + offset (0x000c4)

Start address of destination image(Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_cr_mst Destination image data Cr start address in Memory

IEP_DIL_MTN_TAB0

Address: Operational Base + offset (0x000e0)

Deinterlace motion table0

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

IEP_DIL_MTN_TAB1

Address: Operational Base + offset (0x000e4)

Deinterlace motion table1

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

IEP_DIL_MTN_TAB2

Address: Operational Base + offset (0x000e8)

Deinterlace motion table2

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

IEP_DIL_MTN_TAB3

Address: Operational Base + offset (0x000ec)

Deinterlace motion table3

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

IEP_DIL_MTN_TAB4

Address: Operational Base + offset (0x000f0)

Deinterlace motion table4

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

IEP_DIL_MTN_TAB5

Address: Operational Base + offset (0x000f4)

Deinterlace motion table5

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

IEP_DIL_MTN_TAB6

Address: Operational Base + offset (0x000f8)

Deinterlace motion table6

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3

Bit	Attr	Reset Value	Description
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

IEP_DIL_MTN_TAB7

Address: Operational Base + offset (0x000fc)

Deinterlace motion table7

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

IEP_ENH(CG)_TAB

Address: Operational Base + offset (0x00100)

contrast and gamma enhancement table

Bit	Attr	Reset Value	Description
31:24	RW	0x00	cg_tab_3 cg table 3 pixel value 3,7,11,15,.....mapping
23:16	RW	0x00	cg_tab_2 cg table 2 pixel value 2,6,10,14,.....mapping
15:8	RW	0x00	cg_tab_1 cg table 1 pixel value 1,5,9,13,.....mapping

Bit	Attr	Reset Value	Description
7:0	RW	0x00	cg_tab_0 cg table 0 256x8bit contrast & gamma mapping table pixel value 0,4,8,12,.....mapping

IEP_ENH_DDE_COE0

Address: Operational Base + offset (0x00400)

denoise,detail and edge enhancement coefficient

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	RW	0x00	dde_coe_3 dde coefficient 3 coefficient number 3,7,11,15,.....
23:22	RO	0x0	reserved
21:16	RW	0x00	dde_coe_2 dde coefficient 2 coefficient number 2,6,10,14,.....
15:14	RO	0x0	reserved
13:8	RW	0x00	dde_coe_1 dde coefficient 1 coefficient number 1,5,9,13,.....
7:6	RO	0x0	reserved
5:0	RW	0x00	dde_coe_0 dde coefficient 0 256x6bit coefficient for denoise and detail enhancement coefficient number 0,4,8,12,.....

IEP_ENH_DDE_COE1

Address: Operational Base + offset (0x00500)

denoise,detail and edge enhancement coefficient

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	RW	0x00	dde_coe_3 dde coefficient 3 coefficient number 3,7,11,15,.....
23:22	RO	0x0	reserved
21:16	RW	0x00	dde_coe_2 dde coefficient 3 coefficient number 2,6,10,14,.....
15:14	RO	0x0	reserved
13:8	RW	0x00	dde_coe_1 dde coefficient 1 coefficient number 1,5,9,13,.....
7:6	RO	0x0	reserved
5:0	RW	0x00	dde_coe_0 dde coefficient 1 81x6bit coefficient for denoise and detail enhancement coefficient number 0,4,8,12,.....

IEP_MMU_DTE_ADDR

Address: Operational Base + offset (0x00800)

MMU current page table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mmu_dte_addr page table address

IEP_MMU_STATUS

Address: Operational Base + offset (0x00804)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RW	0x00	mmu_page_fault_bus_id Index of master responsible for the last page fault
5	RW	0x0	mmu_page_fault_is_write The direction of access for last page fault: 0: read 1:write
4	RW	0x1	mmu_replay_buffer_empty The MMU replay buffer is empty.
3	RW	0x1	mmu_idle the MMU is idle when accesses are being translated and there are no unfinished translated access. The MMU_IDLE signal only reports idle when the MMU processor is idle and accesses are active on the external bus. Note: the MMU can be idle in page fault mode.
2	RW	0x0	mmu_stall_active MMU stall mode currently enabled. The mode is enabled by command.
1	RW	0x0	mmu_page_fault_active MMU page fault mode currently enabled. The mode is enabled by command
0	RW	0x0	mmu_paging_enabled mmu paging is enabled

IEP_MMU_CMD

Address: Operational Base + offset (0x00808)

MMU command register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	mmu_cmd 0: MMU_ENABLE_PAGING. enable paging. 1: MMU_DISABLE_PAGING. disable paging. 2: MMU_ENABLE_STALL. turn on stall mode. 3: MMU_DISABLE_STALL. turn off stall mode. 4: MMU_ZAP_CACHE. zap the entire page table cache. 5: MMU_PAGE_FAULT_DONE. leave page fault mode. 6: MMU_FORCE_RESET. reset the mmu. The MMU_ENABLE_STALL command can always be issued. Other commands are ignored unless the MMU is idle or stalled.

IEP_MMU_PAGE_FAULT_ADDR

Address: Operational Base + offset (0x0080c)

MMU logic address of last page fault

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mmu_page_fault_addr address of last page fault

IEP_MMU_ZAP_ONE_LINE

Address: Operational Base + offset (0x00810)

MMU zap cache line register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	mmu_zap_one_line address to be invalidated from the page table cache.

IEP_MMU_INT_RAWSTAT

Address: Operational Base + offset (0x00814)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error read bus error
0	RW	0x0	page_fault page fault

IEP_MMU_INT_CLEAR

Address: Operational Base + offset (0x00818)

MMU interrupt clear register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error_clear read bus error interrupt clear. write 1 to this register can clear read bus error interrupt.
0	RW	0x0	page_fault_clear page fault interrupt clear, write 1 to this register can clear page fault interrupt.

IEP_MMU_INT_MASK

Address: Operational Base + offset (0x0081c)

MMU interrupt mask register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error_int_en read bus error interrupt enable
0	RW	0x0	page_fault_int_en page fault interrupt enable

IEP_MMU_INT_STATUS

Address: Operational Base + offset (0x00820)

MMU interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error read bus error interrupt
0	RW	0x0	page_fault page fault interrupt

IEP_MMU_AUTO_GATING

Address: Operational Base + offset (0x00824)

MMU clock auto gating register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_gating mmu clock auto gating when it is 1, the mmu will auto gating itself

20.5 Application Notes

20.5.1 VOP path disabled configure flow

1. Keep IEP direct path disabled.
2. Configure all registers which are needed at any time.
3. Configure IEP_CONFIG_DONE.
4. Configure IEP_FRM_START.

20.5.2 VOP path enabled configure flow

1. Keep IEP direct path enabled.
2. Configure all IEP registers which are needed.
3. Configure VOP related registers which are needed.
4. Configure CONFIG_DONE register in VOP only.
5. Wait for frame start from VOP and IEP direct path.

20.5.3 VOP path turn on flow

1. Configure all IEP registers which are needed.
2. Configure VOP related registers which are needed.
3. Enable IEP direct path.
4. Enable VOP direct path.
5. Configure CONFIG_DONE register in VOP only.
6. Wait for frame start from VOP and IEP direct path.

20.5.4 VOP path turn off flow

1. Disable VOP direct path.
2. Disable IEP direct path, so IEP do not receive any other CONFIG_DONE and frame start from VOP immediately.
3. Configure CONFIG_DONE register in VOP.
4. Wait for frame start from VOP and IEP direct path, so VOP quit direct path mode completely.
5. Configure IEP registers which are needed at any time.
6. Configure IEP_CONFIG_DONE.
7. Configure IEP_FRM_START, IEP is working at write back mode now.

Chapter 21 Camera Interface (CIF)

21.1 Overview

The Camera interface, receives the data from Camera or CCIR656 encoder, and transfers the data into system main memory by AXI bus.

The features of camera interface are as follow:

- Support YCbCr422 input
- Support Raw8bit input
- Support CCIR656(PAL/NTSC) input
- Support JPEG input
- Support YCbCr422/420 output
- Support UYVY/VYUY/YUYV/YVYU configurable
- Support up to 8192x8192 resolution source
- Support picture in picture
- Support arbitrary size window crop
- Support error/terminate interrupt and combined interrupt output
- Support clk/vsync/href polarity configurable
- Support one frame stop/ping-pong mode

21.2 Block Diagram

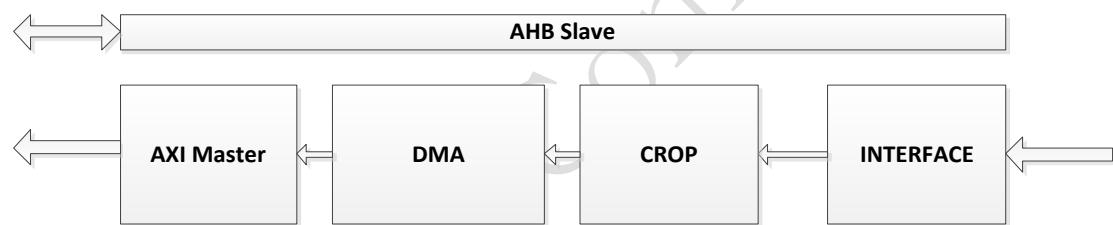


Fig. 21-1CIF block diagram

The CIF comprises with:

- AHB Slave
Host configure the registers via the AHB Slave
- AXI Master
Transmit the data to chip memory via the AXI Master
- INTERFACE
Translate the input video data into the requisite data format
- CROP
Bypass or crop the source video data to a smaller size destination
- DMA
Control the operation of AXI Master

21.3 Function description

This chapter is used to illustrate the operational behavior of how CIF works. If YUV422 or ccir656 signal is received from external devices, CIF translate it into YUV422/420 data, and separate the data to Y and UV data, then store them to different memory via AXI bus separately. But if raw data is received, there are not any translations happened, the 8 data is considered as 16bit data and write directly to memory.

21.3.1 Support Vsync high active or low active

- Vsync Low active as below

Vertical sensor timing (line by line)

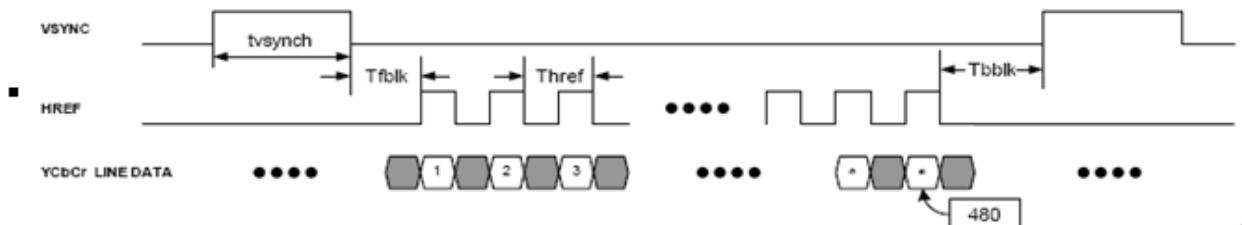


Fig. 21-2 Timing diagram for CIF when vsync low active

- Vsync High active

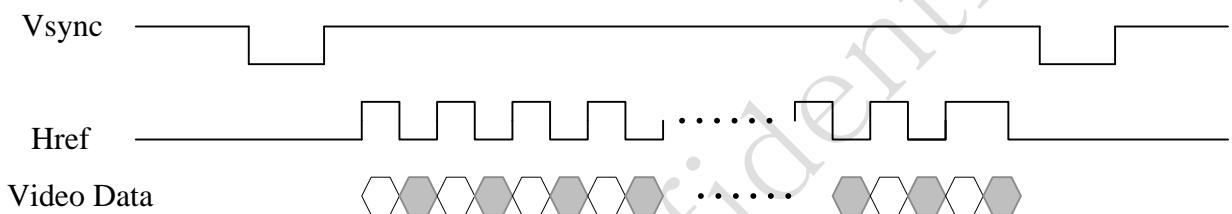


Fig. 21-3 Timing diagram for CIF when vsync high active

21.3.2 Support href high active or low active

- Href high active

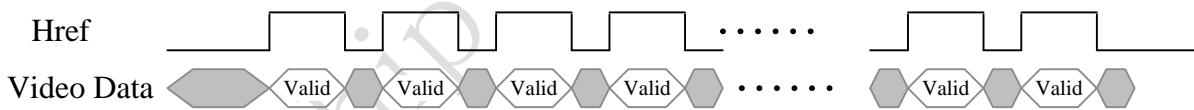


Fig. 21-4 Timing diagram for CIF when href high active

- Href Low active

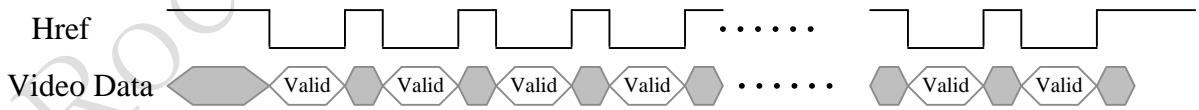


Fig. 21-5 Timing diagram for CIF when href low active

- Y first

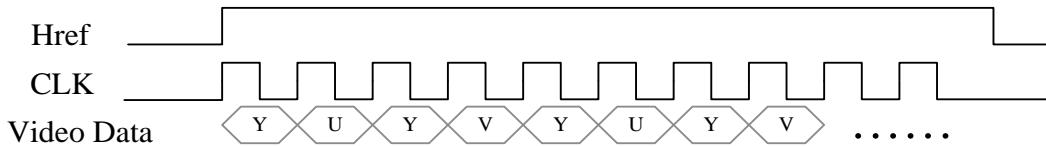


Fig. 21-6 Timing diagram for CIF when Y data first

- U first

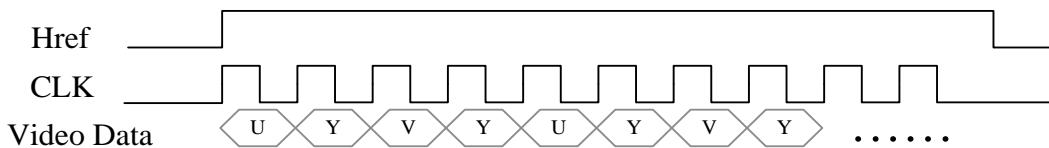


Fig. 21-7 Timing diagram for CIF when U data first

21.3.3 Support CCIR656 (NTSC and PAL)

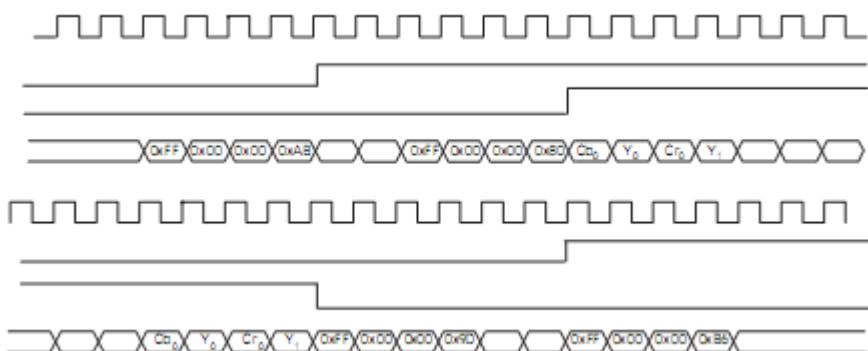


Fig. 21-8 CCIR656 timing

21.3.4 Support Raw data(8-bit) or JPEG

Pixel Data Timing Example

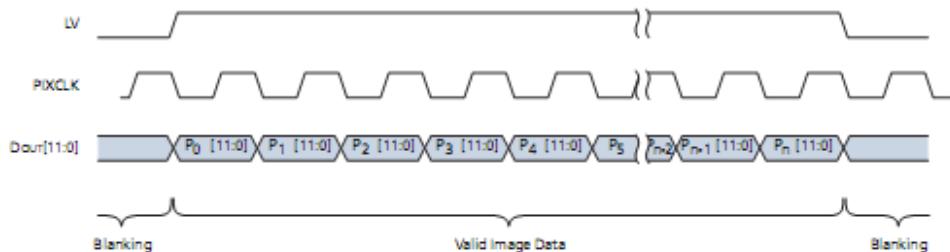


Fig. 21-9 Raw Data or JPEG Timing

CIF module can work in three modes: one frame stop mode, ping-pong mode.

One frame stop mode

In this mode, configure the parameter WORK_MODE to one frame stop mode. After one frame captured, CIF will automatic stop. After capturing, the image Y, UV data will be stored at main memory location defined by CIF_FRM0_ADDR_Y, FRM0_ADDR_UV separately.

Ping-Pong mode

After one frame(F1) captured, CIF will start to capture the next frame(F2) automatically, and host must assign new address pointer of frame1 and clear the frame1 status, thus CIF will capture the third frame automatically(by new F1 address) without any stop and so on for the following frames. But if host did not update the frame buffer address, the CIF will cover the pre-frame data stored in the memory with the following frame data.

Storage

Difference between the YUV mode and raw mode is that in the YUV mode or ccir656 mode, data will be storage in the Y data buffer and UV data buffer; but in the raw or jpeg mode, RGB data will be storage in the same buffer. In addition, in the yuv mode, the width of Y, U or V data is a byte in memory; in Raw or JPEG mode, the width is a halfword no matter the data source is 8 bit.

CROP

The parameter START_Y and START_X defines the coordinate of crop start point. And the frame size after cropping is following the value of SET_WIDTH and SET_HEIGHT.

21.4 Register description

21.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
CIF_CIF_CTRL	0x0000	W	0x00007000	CIF control
CIF_CIF_INTEN	0x0004	W	0x00000000	CIF interrupt enable
CIF_CIF_INTSTAT	0x0008	W	0x00000000	CIF interrupt status
CIF_CIF_FOR	0x000c	W	0x00000000	CIF format
CIF_CIF_FRM0_ADDR_Y	0x0014	W	0x00000000	CIF frame0 y address
CIF_CIF_FRM0_ADDR_UV	0x0018	W	0x00000000	CIF frame0 uv address
CIF_CIF_FRM1_ADDR_Y	0x001c	W	0x00000000	CIF frame1 y address
CIF_CIF_FRM1_ADDR_UV	0x0020	W	0x00000000	CIF frame1 uv address
CIF_CIF_VIR_LINE_WIDTH	0x0024	W	0x00000000	CIF virtual line width
CIF_CIF_SET_SIZE	0x0028	W	0x01e002d0	CIF frame set size
CIF_CIF_CROP	0x0044	W	0x00000000	CIF crop start point
CIF_CIF_SCL_CTRL	0x0048	W	0x00000000	CIF scale control
CIF_CIF_FIFO_ENTRY	0x0054	W	0x00000000	CIF FIFO entry
CIF_CIF_FRAME_STATUS	0x0060	W	0x00000000	CIF frame status
CIF_CIF_CUR_DST	0x0064	W	0x00000000	CIF current destination address
CIF_CIF_LAST_LINE	0x0068	W	0x00000000	CIF last frame line number
CIF_CIF_LAST_PIX	0x006c	W	0x00000000	CIF last line pixel number

Notes: **Size:** **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

21.4.2 Detail Register Description

CIF_CIF_CTRL

Address: Operational Base + offset (0x0000)

CIF control

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:12	RW	0x7	AXI_BURST_TYPE axi master burst type 0-15 : burst1~16
11:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:1	RW	0x0	WORK_MODE Working Mode 00-one frame stop mode 01-ping-pong mode 02-line loop mode 03-reserved
0	RW	0x0	CAP_EN capture enable 0-disable 1-enable

CIF_CIF_INTEN

Address: Operational Base + offset (0x0004)

CIF interrupt enable

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	BUS_ERR_EN bus error axi master or ahb slave response error 0-disable 1-enable
5:4	RO	0x0	reserved
3	RW	0x0	PIX_ERR_EN pixel err interrupt enable the pixel number of last line not equal to the set height 0-disable 1-enable
2	RW	0x0	LINE_ERR_EN line err interrupt enable the line number of last frame not equal to the set height 0-disable 1-enable
1	RW	0x0	LINE_END_EN line end interrupt enable 0-disable 1-enable
0	RW	0x0	FRAME_END_EN frame end interrupt enable after dma transfer the frame data 0-disable 1-enable

CIF_CIF_INTSTAT

Address: Operational Base + offset (0x0008)

CIF interrupt status

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	W1C	0x0	BUS_ERR bus error axi master or ahb slave response error 0-no interrupt 1-interrupt
5:4	RO	0x0	reserved
3	W1C	0x0	PIX_ERR pixel err interrupt the pixel number of last line not equal to the set height 0-no interrupt 1-interrupt
2	W1C	0x0	LINE_ERR line err interrupt the line number of last frame not equal to the set height 0-no interrupt 1-interrupt
1	W1C	0x0	LINE_END line end interrupt enable 0-no interrupt 1-interrupt
0	W1C	0x0	FRAME_END frame end interrupt after dma transfer the frame data 0-no interrupt 1-interrupt

CIF_CIF_FOR

Address: Operational Base + offset (0x000c)

CIF format

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19	RW	0x0	UV_STORE_ORDER UV storage order 0 - UVUV 1 - VUVU
18	RW	0x0	RAW_END raw data endian 0 - little end 1 - big end
17	RW	0x0	OUT_420_ORDER output 420 order 00 - UV in the even line 01 - UV in the odd line Note: The first line is even line(line 0).
16	RW	0x0	OUTPUT_420 output 420 or 422 0 - output is 422 1 - output is 420
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:11	RW	0x0	RAW_WIDTH raw data width must be 2'b00.
10	RW	0x0	JPEG_MODE JPEG mode 0 - other mode 1 - mode1
9	RW	0x0	FIELD_ORDER ccir input order 0-odd field first 1-even field first
8	RW	0x0	IN_420_ORDER 420 input order 00 - UV in the even line 01 - UV in the odd line Note: The first line is even line(line 0).
7	RW	0x0	INPUT_420 input 420 or 422 0 - 422 1 - 420
6:5	RW	0x0	YUV_IN_ORDER YUV input order 00 - UYVY 01 - YVYU 10 - VYUY 11 - YUYV
4:2	RW	0x0	INPUT_MODE input mode 000 - YUV 010 - PAL 011 - NTSC 100 - RAW 101 - JPEG 110 - MIPI Other - invalid
1	RW	0x0	HREF_POL href input polarity 0-high active 1-low active
0	RW	0x0	VSYNC_POL vsync input polarity 0-low active 1-high active

CIF_CIF_FRM0_ADDR_Y

Address: Operational Base + offset (0x0014)

CIF frame0 y address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM0_ADDR_Y frame0 y address

CIF_CIF_FRM0_ADDR_UV

Address: Operational Base + offset (0x0018)

CIF frame0 uv address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM0_ADDR_UV frame0 uv address

CIF_CIF_FRM1_ADDR_Y

Address: Operational Base + offset (0x001c)

CIF frame1 y address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM1_ADDR_Y frame1 y address

CIF_CIF_FRM1_ADDR_UV

Address: Operational Base + offset (0x0020)

CIF frame1 uv address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM1_ADDR_UV frame1 uv address

CIF_CIF_VIR_LINE_WIDTH

Address: Operational Base + offset (0x0024)

CIF virtual line width

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:0	RW	0x0000	VIR_LINE_WIDTH virtual line width

CIF_CIF_SET_SIZE

Address: Operational Base + offset (0x0028)

CIF frame set size

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x01e0	SET_HEIGHT set height
15:13	RO	0x0	reserved
12:0	RW	0x02d0	SET_WIDTH set width

CIF_CIF_CROP

Address: Operational Base + offset (0x0044)

CIF crop start point

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x0000	START_Y start y point
15:13	RO	0x0	reserved
12:0	RW	0x0000	START_X start x point

CIF_CIF_SCL_CTRL

Address: Operational Base + offset (0x0048)

CIF scale control

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	RAW_16B_BP raw 16 bit bypass 0-no bypass 1-bypass
4	RW	0x0	YUV_16B_BP YUV 16 bit bypass 0-no bypass 1-bypass
3:0	RO	0x0	reserved

CIF_CIF_FIFO_ENTRY

Address: Operational Base + offset (0x0054)

CIF FIFO entry

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:8	RW	0x00	UV_FIFO_ENTRY valid UV double word in FIFO write 0 clear
7	RO	0x0	reserved
6:0	RO	0x00	Y_FIFO_ENTRY valid Y double word in FIFO write 0 clear

CIF_CIF_FRAME_STATUS

Address: Operational Base + offset (0x0060)

CIF frame status

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	FRAME_NUM complete frame number write 0 to clear
15:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	F1_STS frame 0 status 0- frame 1 not ready 1- frame 1 ready write 0 clear
0	RO	0x0	F0_STS frame 0 status 0- frame 0 not ready 1- frame 0 ready write 0 clear

CIF_CIF_CUR_DST

Address: Operational Base + offset (0x0064)

CIF current destination address

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CUR_DST current destination address maybe not the current, because the clock synchronization.

CIF_CIF_LAST_LINE

Address: Operational Base + offset (0x0068)

CIF last frame line number

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:0	RO	0x0000	LAST_LINE_NUM line number of last frame

CIF_CIF_LAST_PIX

Address: Operational Base + offset (0x006c)

CIF last line pixel number

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:0	RO	0x0000	LAST_PIX_NUM pixel number of last line

21.5 Interface description

Module Pin	Direction	Pad Name	IOMUX Setting
cif_clkout	O	TS0clk_CIFclkout	GRF_CIFD_IOMUX1[6]==1'b0
cif_clkin	I	TS0valid_CIFclkin	GRF_CIFD_IOMUX1[4]==1'b0
cif_href	I	TS0err_CIFhref	GRF_CIFD_IOMUX1[2]==1'b0
cif_vsync	I	TS0sync_CIFvsync	GRF_CIFD_IOMUX1[0]==1'b0
cif_data0	I	TS0d0_CIFd0	GRF_CIFD_IOMUX[0]==1'b0
cif_data1	I	TS0d1_CIFd1	GRF_CIFD_IOMUX[2]==1'b0

cif_data2	I	TS0d2_CIFd2	GRF_CIFD_IOMUX[4]==1'b0
cif_data3	I	TS0d3_CIFd3	GRF_CIFD_IOMUX[6]==1'b0
cif_data4	I	TS0d4_CIFd4	GRF_CIFD_IOMUX[8]==1'b0
cif_data5	I	TS0d5_CIFd5	GRF_CIFD_IOMUX[10]==1'b0
cif_data6	I	TS0d6_CIFd6	GRF_CIFD_IOMUX[12]==1'b0
cif_data7	I	TS0d7_CIFd7	GRF_CIFD_IOMUX[14]==1'b0

21.6 Application Notes

The biggest configuration requirement of all operations is the CAP_EN bit must be set after all the mode selection is ready. The configuration order of the input/output data format, YUV order, the address, frame size/width, AXI burst length and other options do not need to care.

There are many debug registers to make it easy to read the internal operation information of CIF. The valid pixel number of scale result in FIFO can be known by read CIF_CIF_SCL_VALID_NUM. The line number of last frame and the pixel number of last line can be also known by read the CIF_CIF_LAST_LINE and CIF_CIF_LAST_PIX.

Chapter 22 EBC

22.1 Overview

EBC is the TCON module for Electronic Paper Display (EPD).

22.1.1 Features

- **systeminterface**
 - AHB slave for register configuration
 - AHB master for frame data transfer (DMA)
 - Interrupt output
- **EPDinterface**
 - E-ink EPD compatible
 - Up to 2048x2048 resolution
 - Up to 16 level gray scale
 - Up to 128 frames every scanning
 - LUT updateable (8KB)
 - Direct mode and LUT mode
 - All-update mode and Diff-update mode
 - Single-phase and multi-phase mode
 - Support window display
 - Source driver interface
 - Gate driver interface

22.2 Block Diagram

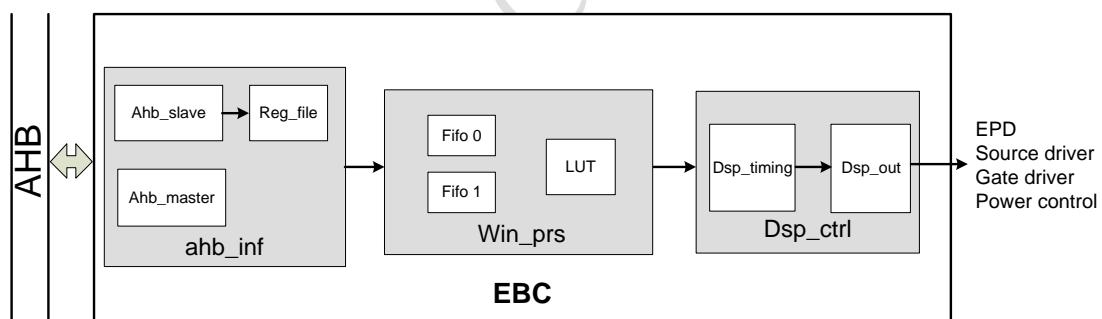


Fig.22-1EBC Block Diagram

22.3 Function description

22.3.1 Data Format

Table 22-1EBC Input Data Format

DSP_LUT MODE	WIN_FMT [1:0]	Data format	Bit Map
0	xx	S-data(2bpp)	{S3[7:0], S2[7:0], S1[7:0], S0[7:0]}
1	00	Y-data(4bpp)	{Y7[3:0], Y6[3:0], Y5[3:0], Y4[3:0], Y3[3:0], Y2[3:0], Y1[3:0], Y0[3:0]}
1	01	Y-data(8bpp)	{Y3[7:0], Y2[7:0], Y1[7:0], Y0[7:0]}
1	10	RGB888	{8'bx, R[7:0], G[7:0], B[7:0]}
1	11	RGB565	{R1[4:0], G1[5:0], B1[4:0], R0[4:0], G0[5:0], B0[4:0]}

22.3.2 Waveform generation mode

4. Direct mode

In direct scanning mode, the source data is just read by internal DMA and sent to the display output directly.

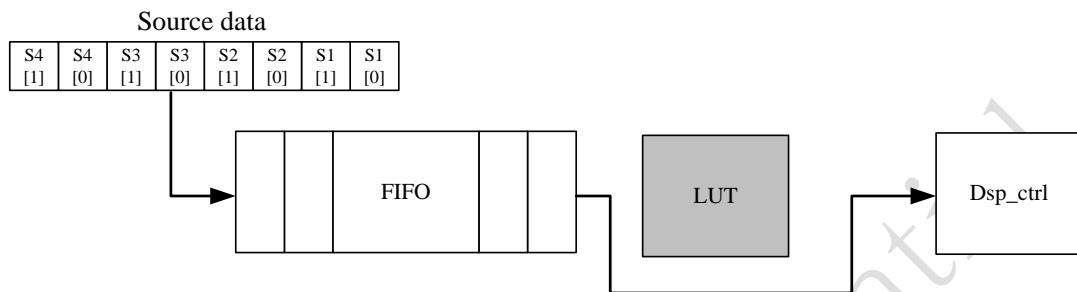


Fig.22-2 EBC Block Diagram

5. LUT mode

In LUT scanning mode, internal DMA read the original pixel data into the FIFO, then the pixel data is sent to the look-up table to be translated the EPD source data.

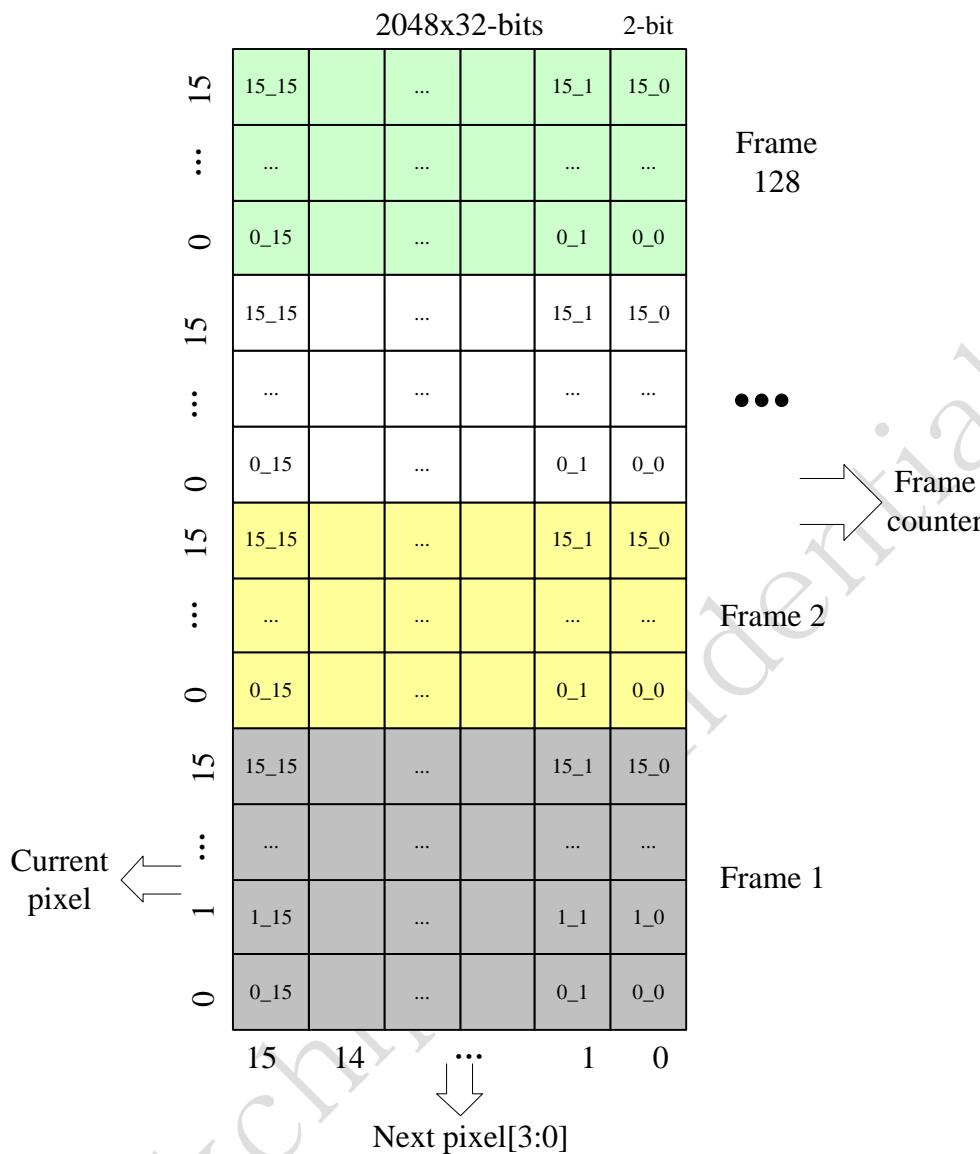


Fig.22-3 EBC LUT structure

22.3.3 Window display mode

Window display is supported in EBC, `dsp_win_width/dsp_win_height` and `dsp_win_st` should be set to define the display window. The source value of other area is background value, which is configurable.

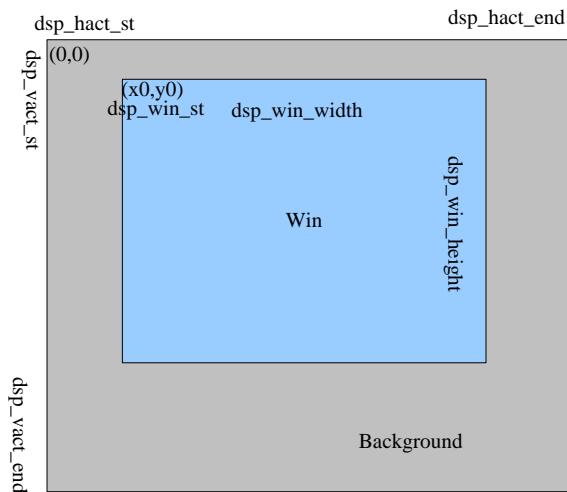


Fig.22-4 EBC window display

22.3.4 Frame control

1. Single frame mode(always used in direct mode)

In single frame mode (direct mode), every display frame is controlled by the "frame_start" command. The next "frame_start" command should be after the end of the last frame.

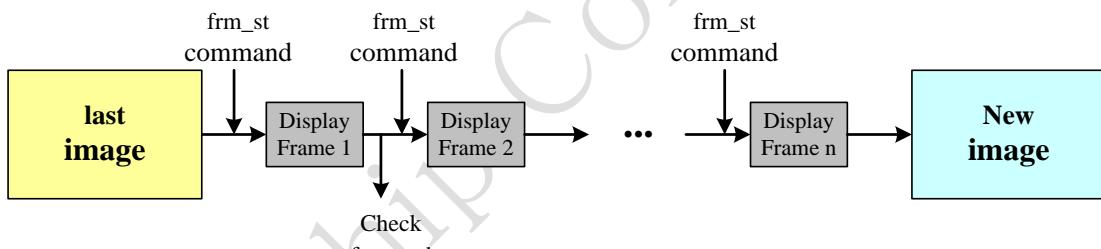


Fig.22-5 EBC single frame display

2. Multi-frame mode

In multi-frame mode, the number of the display frames should be set before the "frame_start" command. Then the frame scanning is done automatically until the end of the last frame.

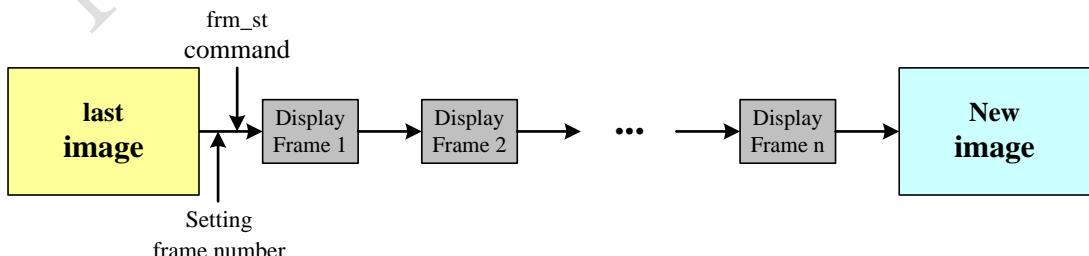


Fig.22-6 EBC multi-frame display

22.3.5 Image translation phase

1. Single phase mode

If the number of image translation frames is less than 64. Single phase is necessary for image translation

In this mode, the “vcom_en” is just turn on during the display period.

2. Multi-phase mode

If the number of image translation frames is larger than 64, multi-phase should be done for image translation because the max display frame is 64.

In this mode, the “vcom_en” should not be turn off between the idle period of different display phase. To avoid this, you can set VCOM_MODE = “1” before the first start display start command. To turn off the VCOM_EN, “0” should be write in before the last start display start command.

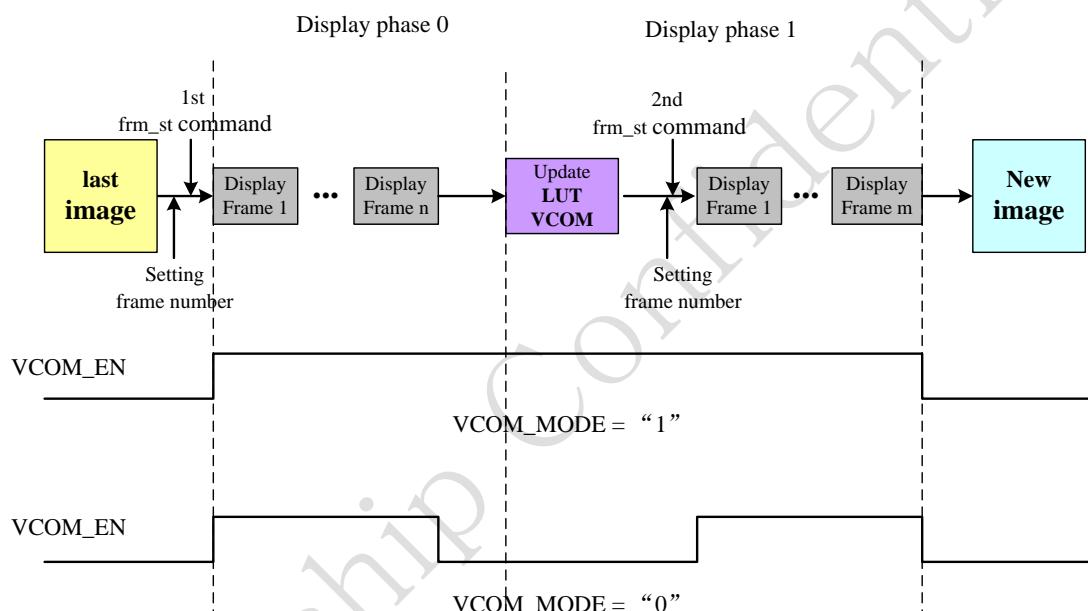


Fig.22-7 EBC display phase

22.3.6 IO description

Table 22-2EBC output pins

EINK	Description
SCLK	Source clock
SLE	Source latch data enable
SOE	Source data enable/start pulse
SCE[5:0]	
SDO[7:0]	Source data
SDIR	X scanning direction
GCLK	Gate clock
GSP	Gate start pulse
GOE	Gate output enable
GDIR	Y scanning direction
BORDER[1:0]	COM voltage
VCOM_CTRL	VCOM enable
PWR[2:0]	POWER control

1. IO MUX

EBC share IOs with LCDC and LVDS, following is the IO map.

For EBC output, first switch LVCD's TTL outputs source to EBC outputs by setting GRF_LVDS_CON0[0] = 1'b1. Then switch LCDC's output IOs to EBC outputs by setting GRF_GPIO2B[15:0] = 16'haaaa, GRF_GPIO2C[15:0] = 16'haaaa and GRF_GPIO2D[3:0] = 4'ha.

Table 22-3EBC IOMUX

IOMUX			
	lc当地	ebc_sdclk	
	lc当地_dclk	ebc_sdclk	
	lc当地_hsync	ebc_sdle	
	lc当地_vsync	ebc_sdoe	
	lc当地_den	ebc_gdclk	
lvds_p0	lc当地_d0	ebc_sddo0	
lvds_n0	lc当地_d1	ebc_sddo1	
lvds_p1	lc当地_d2	ebc_sddo2	
lvds_n1	lc当地_d3	ebc_sddo3	
lvds_p2	lc当地_d4	ebc_sddo4	
lvds_n2	lc当地_d5	ebc_sddo5	
lvds_p3	lc当地_d6	ebc_sddo6	
lvds_n3	lc当地_d7	ebc_sddo7	
lvds_clkp	lc当地_d8	ebc_sdce0	
lvds_clkn	lc当地_d9	ebc_sdce1	
	lc当地_d10	ebc_sdce2	
	lc当地_d11	ebc_sdce3	
	lc当地_d12	ebc_sdce4	
	lc当地_d13	ebc_sdce5	
	lc当地_d14	ebc_vcom	
	lc当地_d15	ebc_gdoe	
	lc当地_d16	ebc_gdsp	
	lc当地_d17	ebc_gdpwr0	
	lc当地_d18	ebc_gdrl	i2c2_sda
	lc当地_d19	ebc_sdshr	i2c2_scl
	lc当地_d20	ebc_border0	urt2_sin
	lc当地_d21	ebc_border1	urt2_sout
	lc当地_d22	ebc_gdpwr1	
	lc当地_d23	ebc_gdpwr2	

2. Source driver interface

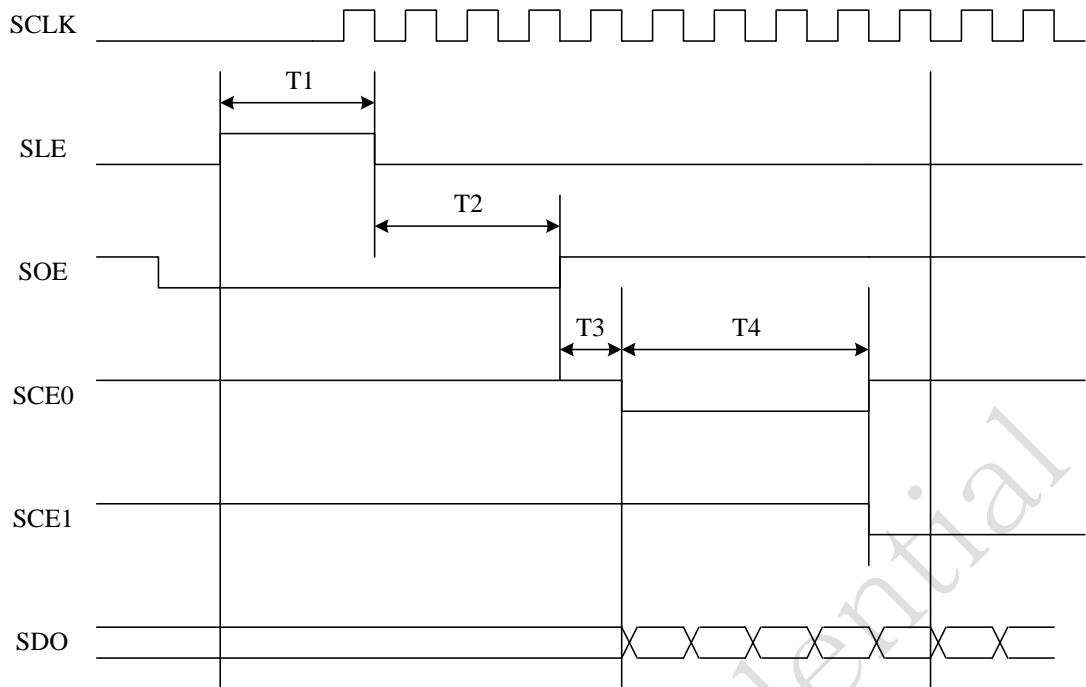


Fig.22-8 EBC source driver timing 1

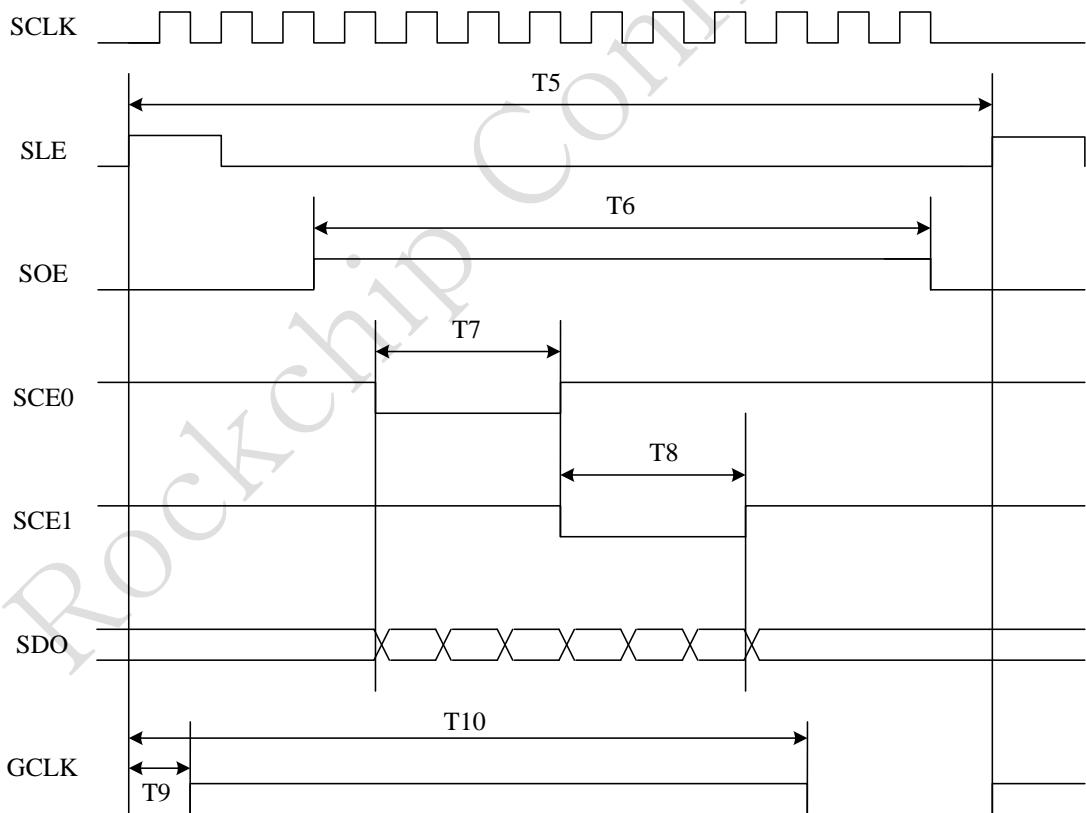


Fig.22-9 EBC source driver timing 2

Table 22-4EBC source driver setting

Timing	Value (6-inch 800x600)	Register setting
DCLK	33.25MHz	

SCLK	4 DCLK(8.3125 MHz)	(DSP_CLK_DIV+1)
T1	10 SCLK	DSP_HS_END
T2	3 SCLK	(DSP_HACT_ST - DSP_HS_END)
T3	1 SCLK	Fixed
T1+T2+T3	14 SCLK	DSP_WIN_XST
T4	100 SCLK	DSP_SCE_WIDTH
T5	315 SCLK	DSP_HTOTAL
T6	300 SCLK	(DSP_HACT_END - DSP_HACT_ST)
T7	100 SCLK	DSP_SCE_WIDTH
T8	100 SCLK	DSP_SCE_WIDTH
T9	0 SCLK	DSP_GCLK_ST
T10	215 SCLK	DSP_GCLK_END

3. Gate driver interface

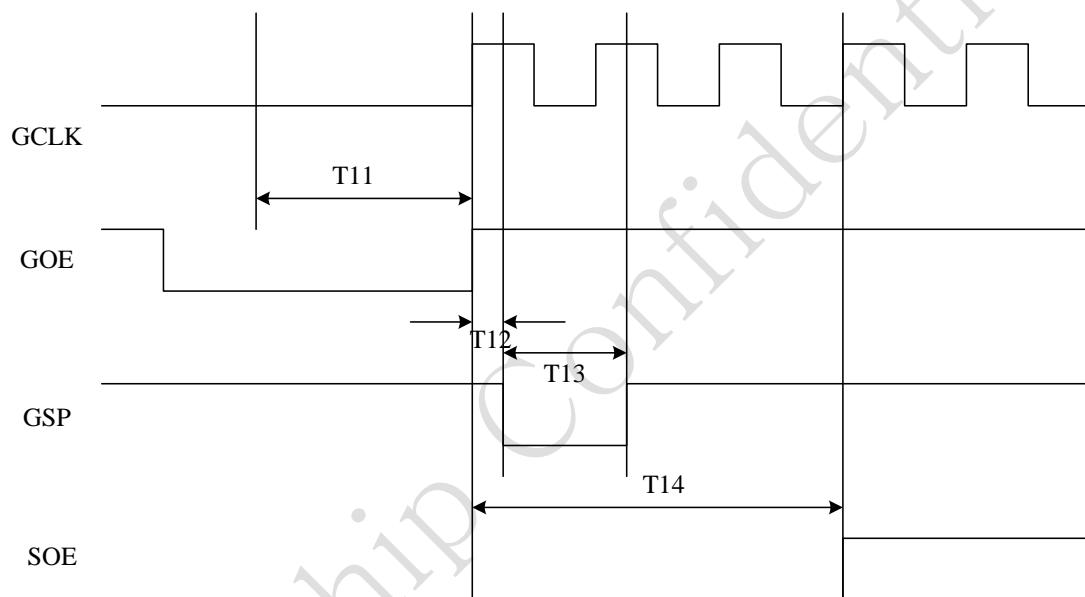


Fig.22-10 EBC gate driver timing 1

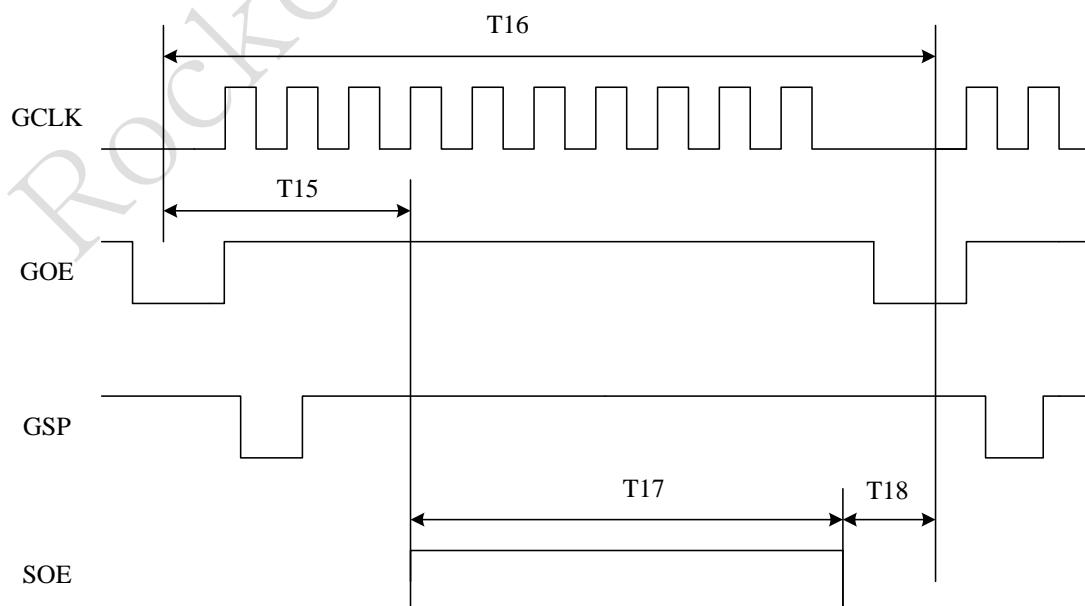


Fig.22-11 EBC gate driver timing 2

Table 22-5EBC gate driver setting

Timing	Value (6-inch 800x600)	Register setting
GCLK	315 SCLK	DSP_HTOTAL
T11	4 GCLK	DSP_VS_END
T12	107 SCLK	(DSP_GD_END/2)
T13	1 GCLK	Fixed
T14	4 GCLK	(DSP_VACT_ST - DSP_VS_END)
T15	8 GCLK	DSP_VACT_ST
T16	619 GCLK	DSP_VTOTAL
T17	601 GCLK	(DSP_VACT_END - DSP_VACT_ST)
T18	11 GCLK	(DSP_VTOTAL - DSP_VACT_END)

22.4 Register Description

22.4.1 Register summary

Name	Offset	Size	Reset Value	Description
EBC_DSP_ST	0x0000	W	0x00000000	frame start register
EBC_EPD_CTRL	0x0004	W	0x00006400	EPD control register
EBC_DSP_CTRL	0x0008	W	0x00030000	Display control register
EBC_DSP_HTIMING0	0x000c	W	0x013b000a	Display horizontal timing setting0
EBC_DSP_HTIMING1	0x0010	W	0x0139000d	Display horizontal timing setting1
EBC_DSP_VTIMING0	0x0014	W	0x026b0004	Display vertical timing setting0
EBC_DSP_VTIMING1	0x0018	W	0x02610008	Display vertical timing setting0
EBC_DSP_ACT_INFO	0x001c	W	0x025800c8	Display active width/height
EBC_WIN_CTRL	0x0020	W	0x00000000	Window control register
EBC_WIN_MST0	0x0024	W	0x00000000	Old window layer memory start address
EBC_WIN_MST1	0x0028	W	0x00000000	New window layer memory start address
EBC_WIN_VIR	0x002c	W	0x00000320	Window layer virtual width
EBC_WIN_ACT	0x0030	W	0x02580320	Window active width/height
EBC_WIN_DSP	0x0034	W	0x02580320	Window display width/height
EBC_WIN_DSP_ST	0x0038	W	0x00000000	Window display start position
EBC_INT_CTRL	0x003c	W	0x00000038	Interrupt control register
EBC_VCOM0	0x0040	W	0x00000000	VCOM0
EBC_VCOM1	0x0044	W	0x00000000	VCOM1
EBC_VCOM2	0x0048	W	0x00000000	VCOM2
EBC_VCOM3	0x004c	W	0x00000000	VCOM3
EBC_CONFIG_DONE	0x0050	W	0x00000000	CONFIG finish register
EBC_LUT_ADDR_MAP	0x1000	W	0x00000000	LUT address map

22.4.2 Detail Register Description

EBC_DSP_ST

Address: Operational Base + offset (0x0000)
frame start register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:2	RW	0x00	frm_total_num Frame total number(n) Frame_num = n+1. Up to 128
1	RO	0x0	reserved
0	RW	0x0	frm_st Frame start bit Writing'1' to trigger one frame display. Read this bit for Hold status.

EBC_EPD_CTRL

Address: Operational Base + offset (0x0004)
EPD control register

Bit	Attr	Reset Value	Description
31:27	RW	0x00	dsp_gd_st DSP_GD_ST GCLK rising edge point(SCLK), which count from the falling edge of hsync
26:16	RW	0x0d7	dsp_gd_end DSP_GD_END GCLK falling edge point(SCLK), which count from the falling edge of hsync
15:8	RW	0x64	dsp_sce_width DSP_SCE_WIDTH Source driver length
7:5	RO	0x0	reserved
4:2	RW	0x0	pwr_en Power enable[3:0]
1	RW	0x0	gate_scan_dir Gate scanning direction 1: button to top 0: top to button
0	RW	0x0	source_scan_dir Source scanning direction 1: right to left 0: left to right

EBC_DSP_CTRL

Address: Operational Base + offset (0x0008)
Display control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dsp_swap Display swap 11: SDO[7:0] = P0,P1,P2,P3 10: SDO[7:0] = P2,P3,P0,P1 01: SDO[7:0] = P1,P0,P3,P2 00: SDO[7:0] = P3,P2,P1,P0
29	RW	0x0	diff_update_mode_en Diff update Mode enable 1: Diff update Mode 0: Normal update Mode
28	RW	0x0	dsp_mode Display Mode 1: LUT mode 0: Direct mode
27	RW	0x0	vcom_mode VCOM mode 1: Multi phase scanning mode 0: one phase scanning mode
26	RW	0x0	goe_pol_inv DSP OUTPUT GOE polarity invert 1: invert; 0: default;
25	RW	0x0	gsp_pol_inv DSP OUTPUT GSP polarity invert 1: invert; 0: default;
24	RW	0x0	gclk_pol_inv DSP OUTPUT GCLK polarity invert 1: invert; 0: default;
23	RW	0x0	sce_pol_inv DSP OUTPUT SCE polarity invert 1: invert; 0: default;
22	RW	0x0	soe_pol_inv DSP OUTPUT SOE polarity invert 1: invert; 0: default;
21	RW	0x0	sle_pol_inv DSP OUTPUT SLE polarity invert 1: invert; 0: default;
20	RW	0x0	sclk_pol_inv DSP OUTPUT SCLK polarity invert 1: invert; 0: default;
19:16	RW	0x3	sclk_div SCLK divide rate $SCLK = (DSP_CLK_DIV+1) * DCLK$
15:0	RW	0x0000	sdo_default_val SDO default value

EBC_DSP_HTIMING0

Address: Operational Base + offset (0x000c)

Display horizontal timing setting0

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x13b	dsp_h_period Panel display scanning horizontal period.
15:8	RO	0x0	reserved
7:0	RW	0xa	dsp_hsync_width Panel display scanning hsync(SLE) pulse width.

EBC_DSP_HTIMING1

Address: Operational Base + offset (0x0010)

Display horizontal timing setting1

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x139	dsp_hact_end Panel display scanning horizontal active end point
15:8	RO	0x0	reserved
7:0	RW	0xd	dsp_hact_st Panel display scanning horizontal active start point

EBC_DSP_VTIMING0

Address: Operational Base + offset (0x0014)

Display vertical timing setting0

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x26b	dsp_v_period Panel display scanning vertical period.
15:8	RO	0x0	reserved
7:0	RW	0x04	dsp_vsync_width Panel display scanning vsync(GOE) pulse width.

EBC_DSP_VTIMING1

Address: Operational Base + offset (0x0018)

Display vertical timing setting0

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x261	dsp_vact_end Panel display scanning vertical active end point
15:8	RO	0x0	reserved
7:0	RW	0x08	dsp_vact_st Panel display scanning vertical active start point

EBC_DSP_ACT_INFO

Address: Operational Base + offset (0x001c)

Display active width/height

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x258	dsp_height Display height
15:11	RO	0x0	reserved
10:0	RW	0x0c8	dsp_width Display width

EBC_WIN_CTRL

Address: Operational Base + offset (0x0020)

Window control register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RW	0x0	win_en Win enable
16:12	RW	0x00	ahb_incr_num Ahb master Incrnum
11:9	RW	0x7	ahb_trans_type Ahb master burst type
8:2	RW	0x70	win_fifo_almost_full_level Win fifo almost full level
1	RO	0x0	reserved
0	RW	0x0	win_fmt win_fmt [1:0] win source data format: 11: RGB888(24bpp) 10: RGB565(16bpp) 01: Y-data(8bpp) 00: Y-data(4bpp)

EBC_WIN_MST0

Address: Operational Base + offset (0x0024)

Old window layer memory start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win_mst0 Start address of current image data in memory

EBC_WIN_MST1

Address: Operational Base + offset (0x0028)

New window layer memory start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win_mst1 Start address of next image data in memory

EBC_WIN_VIR

Address: Operational Base + offset (0x002c)
 Window layer virtual width

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0320	win_vir_width Win virtual width

EBC_WIN_ACT

Address: Operational Base + offset (0x0030)
 Window active width/height

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x258	win_act_height Win active height
15:11	RO	0x0	reserved
10:0	RW	0x320	win_act_width Win active width

EBC_WIN_DSP

Address: Operational Base + offset (0x0034)
 Window display width/height

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x258	win_dsp_height Win display height
15:11	RO	0x0	reserved
10:0	RW	0x320	win_dsp_width Win display width

EBC_WIN_DSP_ST

Address: Operational Base + offset (0x0038)
 Window display start position

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x008	win_dsp_y_st Win display y start point
15:11	RO	0x0	reserved
10:0	RW	0x00d	win_dsp_x_st Win display x start point

EBC_INT_CTRL

Address: Operational Base + offset (0x003c)
 Interrupt control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:9	RW	0x00	frm_flag_num Frame number of the scanning flag interrupt The display frame number when the flag interrupt occur, the range is (0~DSP_FRM_TOTAL-1).
8	W1C	0x0	frm_flag_int_clr Frame flag Interrupt clear After be set to 1, this bit will clear by itself 1 cycle later.
7	W1C	0x0	dsp_end_int_clr Display end interrupt clear After be set to 1, this bit will clear by itself 1 cycle later.
6	W1C	0x0	frm_end_int_clr Frame end interrupt clear After be set to 1, this bit will clear by itself 1 cycle later.
5	RW	0x1	frm_flag_int_msk Frame flag Interrupt Mask 0: unmask; 1: mask;
4	RW	0x1	dsp_end_int_msk Display end interrupt mask 0: unmask; 1: mask;
3	RW	0x1	frm_end_int_msk Frame end interrupt mask 0: unmask; 1: mask;
2	RO	0x0	frm_flag_int_sta Frame flag Interrupt status
1	RO	0x0	dsp_end_int_sta Display end interrupt status
0	RO	0x0	frm_end_int_sta Frame end interrupt status

EBC_VCOM0

Address: Operational Base + offset (0x0040)

VCOM0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	vcom0 VCOM [1:0] values in 0-16 frame. The VCOM [1:0] indicate the COM voltage in one frame. The VCOM can be different in different frame. So we can set the VCOM [1:0] sequence using display frames before start the display.

EBC_VCOM1

Address: Operational Base + offset (0x0044)

VCOM1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	vcom1 VCOM [1:0] values in 16-32 frame. The VCOM [1:0] indicate the COM voltage in one frame. The VCOM can be different in different frame. So we can set the VCOM [1:0] sequence using display frames before start the display.

EBC_VCOM2

Address: Operational Base + offset (0x0048)

VCOM2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	vcom2 VCOM [1:0] values in 32-48 frame. The VCOM [1:0] indicate the COM voltage in one frame. The VCOM can be different in different frame. So we can set the VCOM [1:0] sequence using display frames before start the display.

EBC_VCOM3

Address: Operational Base + offset (0x004c)

VCOM3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	vcom3 VCOM [1:0] values in 48-64 frame. The VCOM [1:0] indicate the COM voltage in one frame. The VCOM can be different in different frame. So we can set the VCOM [1:0] sequence using display frames before start the display.

EBC_CONFIG_DONE

Address: Operational Base + offset (0x0050)

CONFIG finish register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1C	0x0	config_done config finish In the first setting of the register, the new value was saved into the mirror register. When all the register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.

EBC_LUT_ADDR_MAP

Address: Operational Base + offset (0x1000)

LUT address map

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	lut_addr_map LUT memory address map (8k Byte)

Chapter 23 LVDS

23.1 Overview

LVDS IP is integrated into MIPI D-PHY. LVDS transmitter converts a CMOS signal into a low-voltage differential signal. Using a differential signal reduces the system's susceptibility to noise and EMI emissions. In addition, using a differential signal can deliver high speeds. This results in a very cost-effective solution to some of the greatest bandwidth bottlenecks in many transmission applications.

23.1.1 Features

- 150MHz clock support
- LVDS 24bits or 18bits color data output
- PLL requires no external components
- Combine LVTTL IO, support LVDS/LVTTL data output
- Comply with the Standard TIA/EIA-644-A LVDS standard
- Support 8bit format-1, format-2, format-3 display mode, Support 6bit display mode.
- Display mode can be selected by input MUX
- Consumes Less Than 1mW When Disabled

23.2 Block Diagram

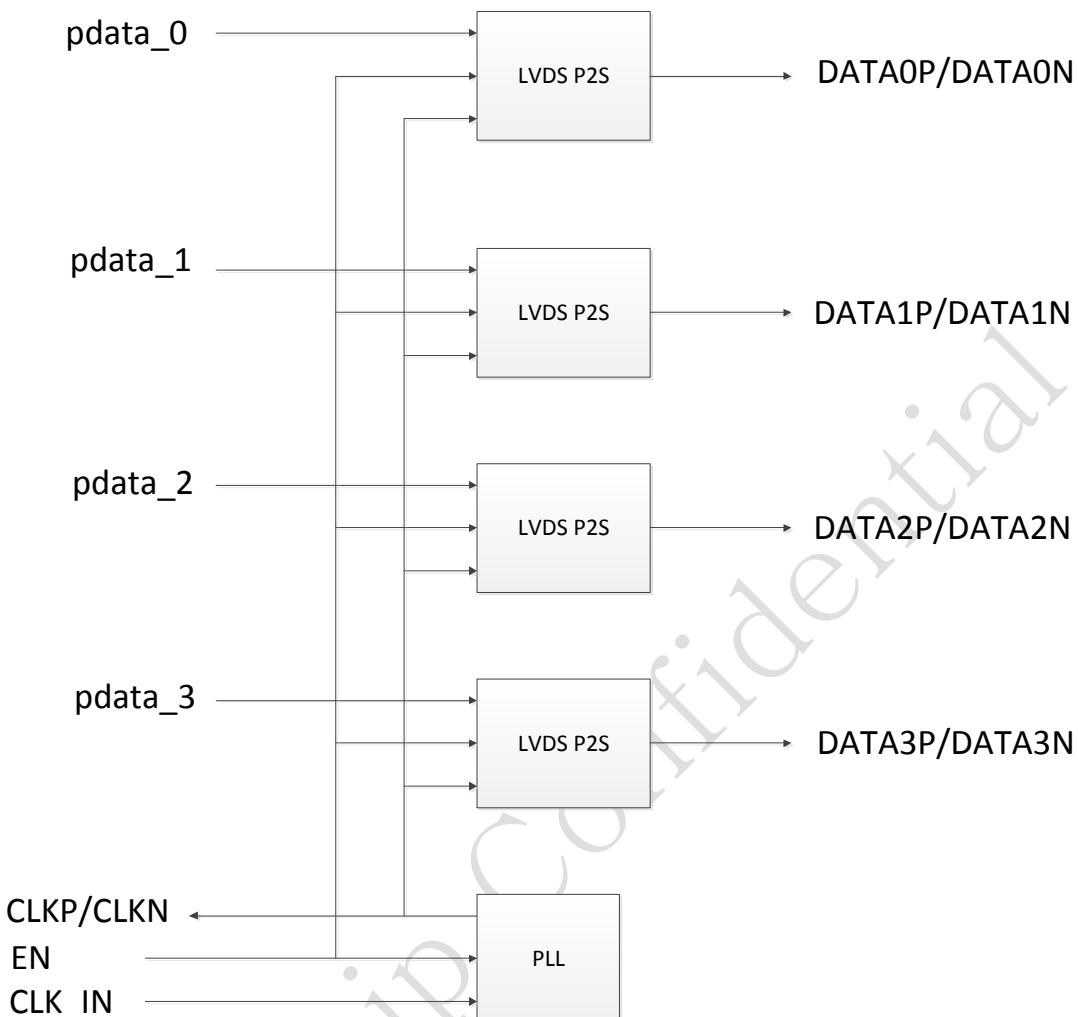


Fig.23-1LVDS TX Block Diagram

23.3 Function description

23.3.1 Video Data Processing

LVDS PHY implements LVDS TIA/EIA protocol. LVDS PHY contains four 7-bit parallel-load serial-out shift registers, a 7X clock PLL, and five LowVoltage Differential Signaling (LVDS) line drivers in a single integrated circuit. These functions allow 28 bits of single-ended LVTTLdata to be synchronously transmitted over five balanced-pair conductors for receipt by a compatible receiver.

When transmitting, parallel data, pdata0/1/2/3 are each loaded into registers upon the edge of the input clock signal (CLKIN). The frequency of CLKIN is multiplied seven times, and then used to unload the data registers in 7-bit slices and serially. The four serial streams and a phase-locked clock (CLKOUT) are then output to LVDS output drivers.

The frequency of CLKOUT is the same as the input clock, CLKIN.

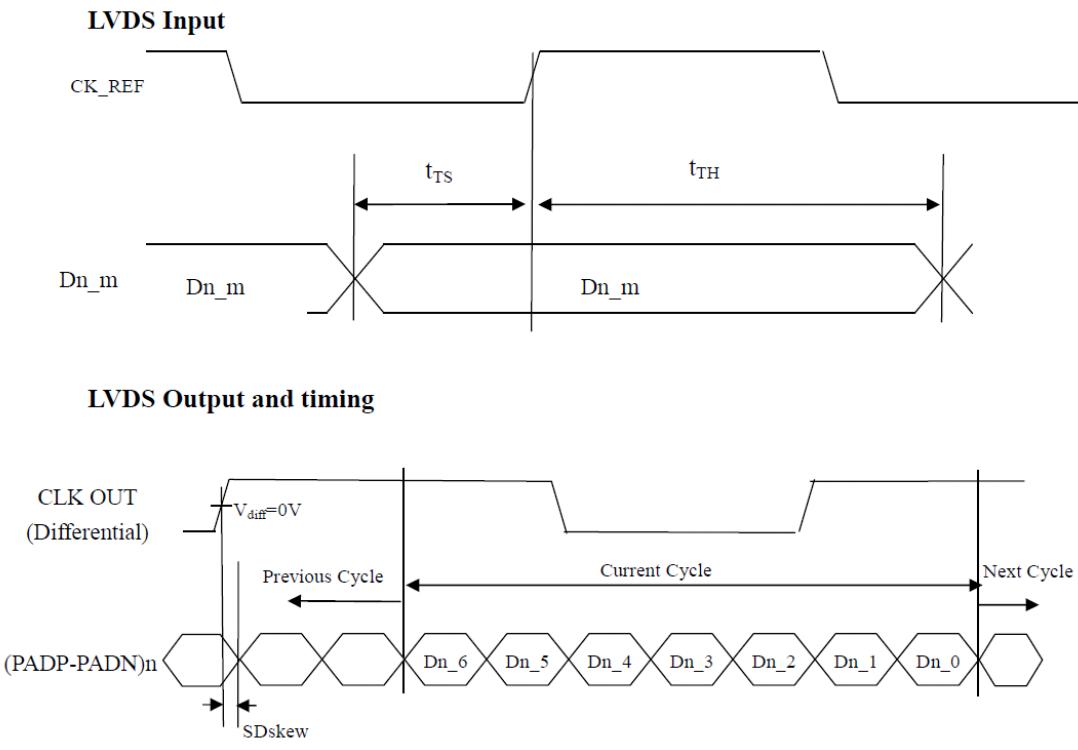


Fig.23-2LVDS TX InterfaceTiming

23.3.2 LVDSFORMAT

LVDSFORMAT converts LCDC RGB interface to LVDS format data, only support 8-bit/6-bit mode. Table 23-4 is LVDSFORMAT input data format.

Table 23-1 is LVDSFORMAT Input Data Format

Serial Channel	Data Bits	8-Bit			6-Bit	4-Bit	
		Format-1 ⁽¹⁾	Format-2 ⁽²⁾	Format-3 ⁽³⁾		Non-Linear Step Size ⁽⁴⁾	Linear Step Size ⁽⁵⁾
Y0	D0	R0	R2	R2	R0	R2	VCC
	D1	R1	R3	R3	R1	R3	GND
	D2	R2	R4	R4	R2	R0	R0
	D3	R3	R5	R5	R3	R1	R1
	D4	R4	R6	R6	R4	R2	R2
	D6	R5	R7	R7	R5	R3	R3
	D7	G0	G2	G2	G0	G2	VCC
Y1	D8	G1	G3	G3	G1	G3	GND
	D9	G2	G4	G4	G2	G0	G0
	D12	G3	G5	G5	G3	G1	G1
	D13	G4	G6	G6	G4	G2	G2
	D14	G5	G7	G7	G5	G3	G3
	D15	B0	B2	B2	B0	B2	VCC
	D18	B1	B3	B3	B1	B3	GND
Y2	D19	B2	B4	B4	B2	B0	B0
	D20	B3	B5	B5	B3	B1	B1
	D21	B4	B6	B6	B4	B2	B2
	D22	B5	B7	B7	B5	B3	B3
	D24	H SYNC	H SYNC	H SYNC	H SYNC	H SYNC	H SYNC
	D25	V SYNC	V SYNC	V SYNC	V SYNC	V SYNC	V SYNC
	D26	ENABLE	ENABLE	ENABLE	ENABLE	ENABLE	ENABLE
Y3	D27	R6	R0	GND	GND	GND	GND
	D5	R7	R1	GND	GND	GND	GND
	D10	G6	G0	GND	GND	GND	GND
	D11	G7	G1	GND	GND	GND	GND
	D16	B6	B0	GND	GND	GND	GND
	D17	B7	B1	GND	GND	GND	GND
	D23	RSVD	RSVD	GND	GND	GND	GND
CLKOUT	CLKIN	CLK	CLK	CLK	CLK	CLK	CLK

⁽¹⁾ Format-1: 2-MSBs of each color transmitted over 4th serial data channel (Y3). Dominant data format for LCD panel.

⁽²⁾ Format-2: 2-LSBs of each color transmitted over 4th serial data channel. System designer needs to verify the data format by checking with the LCD display data sheet.

⁽³⁾ Format-3: 24-bit color host to 18-bit color LCD panel display application.

⁽⁴⁾ Increased dynamic range of the entire color space at the expense of non-linear step sizes between each step.

⁽⁵⁾ Linear step size with less dynamic range.

23.4 Register Description

23.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
MIPI_reg03	0x000c	B	0x03	Register03
MIPI_reg04	0x0010	B	0x7d	Register04
MIPI_rege0	0x0380	B	0x45	Registere0
MIPI_rege1	0x0384	B	0x12	Registere1
MIPI_rege3	0x038c	B	0x01	Registere3
GRF_LVDS_CON0	0x0150	W	0x0100	GRF_LVDS_CON0

Notes: **S**- Byte (8 bits) access, **H**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

23.4.2 Detail Register Description

MIPI_reg03

Address: Operational Base + offset (0x000c)

Register03

Bit	Attr	Reset Value	Description
7:6	RW	0x0	reserved
5	RW	0x0	Reg_fbdv[8] PLL input reference clock divider
4:0	RW	0x3	Reg_prediv[4:0] Integer value programmed into feedback divider

MIPI_reg04

Address: Operational Base + offset (0x0010)

Register04

Bit	Attr	Reset Value	Description
7:0	RW	0x7d	Reg_fbdv[7:0] PLL input reference clock divider

MIPI_rege0

Address: Operational Base + offset (0x0380)

Registere0

Bit	Attr	Reset Value	Description
7:3	RW	0x00	reserved
2	RW	0x1	Digital internal reset Active low, default high
1	RW	0x0	reserved
0	RW	0x1	Selection for MSB and LSB 1'b1: MSB 1'b0: LSB

MIPI_rege1

Address: Operational Base + offset (0x0384)

Registere1

Bit	Attr	Reset Value	Description
7	RW	0x0	Digital internal enable Active high, default low.
6:0	RW	0x12	reserved

MIPI_rege3

Address: Operational Base + offset (0x038c)

Registere3

Bit	Attr	Reset Value	Description
7:3	RW	0x0	reserved
2	RW	0x0	TTL mode enable 1'b1: enable TTL mode 1'b0: disable TTL mode
1	RW	0x0	LVDS mode enable 1'b1: enable LVDS mode 1'b0: disable LVDS mode
0	RW	0x1	Mipi mode enable 1'b1: enable mipi mode 1'b0: disable mipi mode

GRF_LVDS_CON0

Address: Operational Base + offset (0x0150)

GRF_LVDS_CON0

Bit	Attr	Reset Value	Description
31:16	RW	0x0	Write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
13	RW	0x0	Mipiphylane3forceemode 1'b1: enable 1'b0: disable
12	RW	0x0	Mipiphylane2 forceemode 1'b1: enable 1'b0: disable
11	RW	0x0	Mipiphylane1 forceemode 1'b1: enable 1'b0: disable
10	RW	0x0	Mipiphylane0 forceemode 1'b1: enable 1'b0: disable
9	RW	0x0	Mipidsiforceemode 1'b1: enable 1'b0: disable

Bit	Attr	Reset Value	Description
8	RW	0x0	Mipiphylane0turndisable 1'b1: enable 1'b0: disable
7	RW	0x0	Mipiphyttlmode 1'b1: enable 1'b0: disable
6	RW	0x0	Lvds_mode 1'b1: enable 1'b0: disable
5	RW	0x0	Mipictrlldpicoloorm 1'b1: enable 1'b0: disable
4	RW	0x0	Mipictrlldpishutdown 1'b1: enable 1'b0: disable
3	RW	0x0	Lvdsmsbsel 1'b0: MSB is on D0 1'b1: MSB is on D7
2:1	RW	0x0	Lvdsselect 2'b00: 8bit mode format-1 2'b01: 8bit mode format-2 2'b10: 8bit mode format-3 2'b11: 6bit mode
0	RW	0x0	Ebc and Lcdc_datasel 1'b0 : Lcdc_data[9:0] 1'b1 : ebc

23.5 Interface Description

23.5.1 Video Input Source

In RKaudi, the LVDS TX video source comes from VOP.

23.6 ApplicationNotes

23.6.1 LVDS mode

When used in lvds mode, LVDS transmitter source from VOP, vop_dclk need get invert, then input to LVDS.

When lvds panel is LSB receive mode, lvds_msbsel =1, otherwise lvds_msbsel =0.

When LVDS output format is 8bit mode format-1/8bit mode format-2, configure grf_lvds_con0 in 24-bit color mode.

When LVDS output format is 8bit mode format-3/6bit mode, configure grf_lvds_con0 in 18-bit color mode.

Step1: configure GRF_LVDS_CON0

configure MIPI:

Step2: configure PLL

```
    MIPI_reg03 = 0x01;
```

```
    MIPI_reg04 = 0x07;
```

Step3: MIPI_rege0 = 0x45;

```
    MIPI_rege1 = 0x92;
```

```
    MIPI_rege3 = 0x02;
```

23.6.2 TTL mode

When used in lvttl mode, lvds_dclk and rgb_dclk need get invert.

ebc_sel = 1'b0: LVDS transmitter source from lcdc_data[9:0].

ebc_sel = 1'b1: LVDS transmitter source from EBC.

Step1: configure GRF_LVDS_CON0

Step2: configure PLL

```
    MIPI_reg03 = 0x01;
```

```
    MIPI_reg04 = 0x07;
```

Step3: MIPI_rege0 = 0x45;

```
    MIPI_rege1 = 0x92;
```

```
    MIPI_rege3 = 0x04;
```

Chapter 24 HDMI Tx

24.1 Overview

HDMI TX is fully compliant with HDMI 1.4a specification. It offers a simple implementation for consumer electronics like DVD/player/recorder and camcorder. HDMI TX consists of one HDMI transmitter controller and one HDMI transmitter PHY.

24.1.1 Features

- Very low power operation, less than 60mW in PHY during 1080P HD display
- HDMI 1.4a/b/1.3/1.2/1.1, HDCP 1.2 and DVI 1.0 standard compliant transmitter
- Supports data rate from 25MHz, 1.65bps up to 3.4Gbps over a Single channel HDMI
- Support 3D function defined in HDMI 1.4 a/b spec
- TMDS Tx Drivers with programmable output swing, register values and pre-emphasis
- Supports all DTV resolutions including 480i/576i/480p/576p/720p/1080i/1080p
- Digital video interface supports a pixel size of 24bits color depth in RGB
- S/PDIF output supports PCM, Dolby Digital, DTS digital audio transmission (32-192kHz Fs) using IEC60958 and IEC 61937
- Multiphase 4MHz fixed bandwidth PLL with low jitter
- DDC Bus I2C master interface at 3.3V
- HDCP encryption and decryption engine contains all the necessary logic to encrypt the incoming audio and video data
- Support HDMI LipSync if needed as additional feature
- Lower power operation with optimal power management feature
- Embedded ESD, scan support logic.
- Library delivered to support all major EDA tools and detailed guide to integrate into pad ring/BGA bumps
- The EDID and CEC function are also supported by HDMI Transmitter Controller
- Optional Monitor Detection supported through Hot Plug
- 3.3V high speed I/O and 1.2V/1.0V core power supply

24.2 Block Diagram

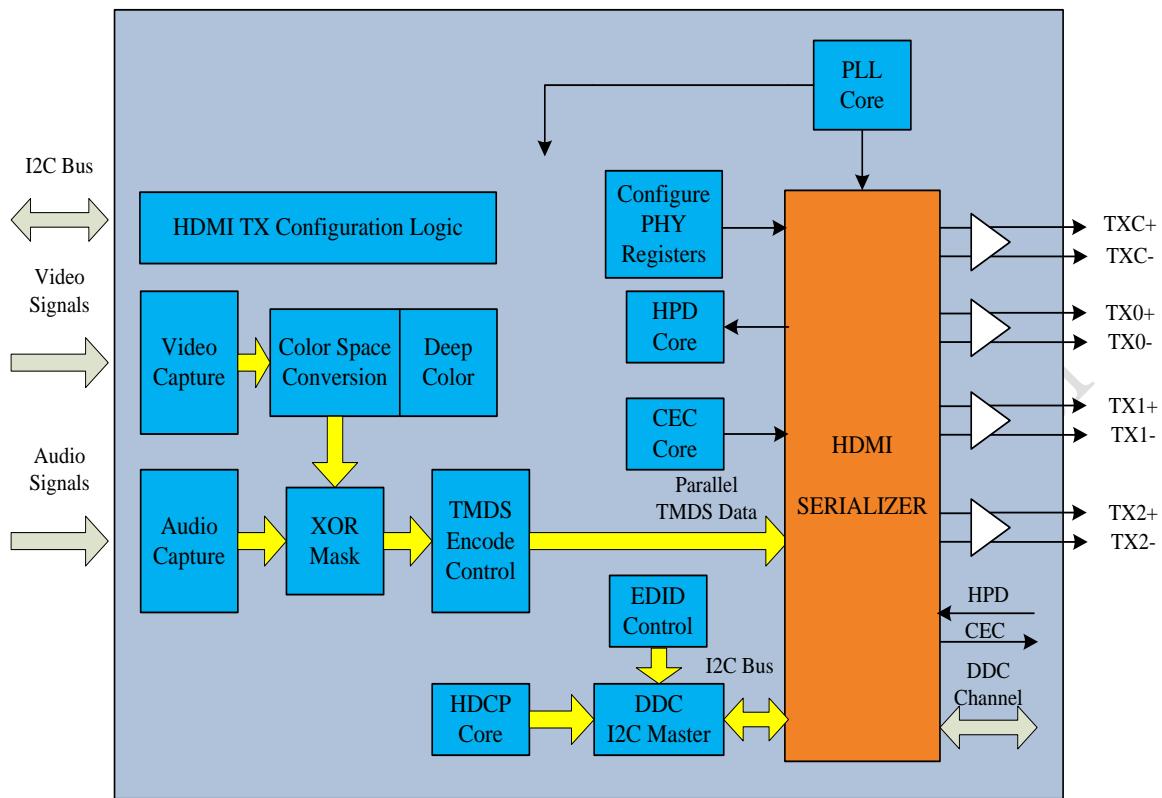


Fig. 24-1 HDMI TX Block Diagram

24.3 Function description

24.3.1 Video Data Processing

The video processing contain video format timings, pixel encodings(RGB to YCbCr, or YCbCr to RGB), colorimetry and corresponding requirements. This function is implemented by some functional blocks, Video Capture block, Color Space Conversion block, and Deep Color block.

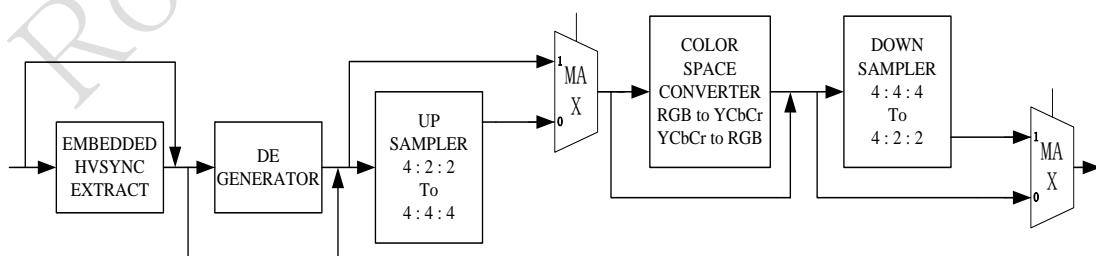


Fig. 24-2 HDMI Video Data Processing

The input video pixels can be encoded in either RGB, YCBCR 4:4:4 or YCBCR 4:2:2 formats by Color Space Conversion block.

The input Video data can have a pixel size of 24, 30, 36 or 48 bits. The deep

color block is used to deal with different pixel size. Video at the default 24-bit color depth is carried at a TMDS clock rate equal to the pixel clock rate. Higher color depths are carried using a correspondingly higher TMDS clock rate. HDMI Transmitter support video formats with TMDS rates below 25MHz (e.g. 13.5MHz for 480i/NTSC) that can be transmitted using a pixel-repetition scheme by setting relative registers.

The following interface timing diagram outlines the Video interface signal format. 24 bit data (we also support 36 bit data for deep color) in RGB can be captured by the rising edge of VCLK with 1ns setup time and 1ns hold time requirements. Control signals such DE and VSync/HSync/FSync going with the same timing relationship.

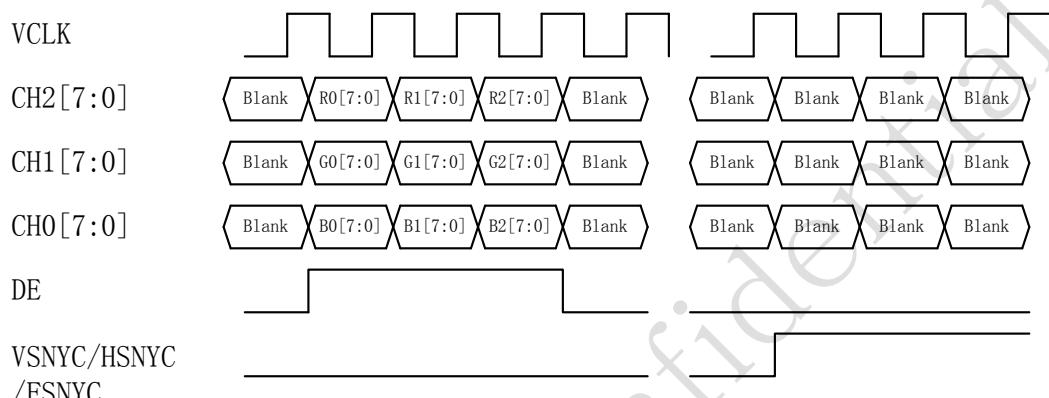


Fig. 24-3HDMI Video Processing Timing

6. Video Data Capture Logic

HDMI TX support input video data related format table is listed below.

Table 24-1HDMI Supported Input Video Formats

Color Space	Pixel Encoding	Sync	Channel Width	Pin Nums
RGB	4:4:4	Separate	8	24
RGB	4:4:4	Separate	10	30
RGB	4:4:4	Separate	12	36
YCbCr	4:4:4	Separate	8	24
YCbCr	4:4:4	Separate	10	30
YCbCr	4:4:4	Separate	12	36
YCbCr	4:2:2	Separate	8	16
YCbCr	4:2:2	Separate	10	20
YCbCr	4:2:2	Separate	12	24
YCbCr	4:4:4	Embedded	8	24
YCbCr	4:4:4	Embedded	10	30
YCbCr	4:4:4	Embedded	12	36
YCbCr	4:2:2	Embedded	8	16
YCbCr	4:2:2	Embedded	10	20

YCbCr	4:2:2	Embedded	12	24
-------	-------	----------	----	----

7. Embedded Sync Extraction Module

The module is used to extract Vsync and Hsync signals from input video data stream such as ITU656 format. With setting the relative registers, this functional module can extract correct video sync signals for later processblock using.

8. Data Enable (DE) Generator

HDMI Transmitter has DE signal generator by incoming HSYNCs, VSYNCs and Video clock. External DE is optional and selected by appropriate register settings. This feature is particularly useful when interfacing to MPEG decoders that do not provide a specific DE output signal.

9. Color Space Conversion

HDMI Transmitter Color space conversion (CSC) is available to interface for several MPEG decoders like with YCbCr-only outputs, and to provide full DVI backwards compatibility.

The function of this module is to perform color space conversion functionality as listed below.

- (1). Convert RGB input Video data to YCbCr Video data.
- (2). Convert YCbCr input Video data to RGB Video data.
- (3). upsample for YCbCr 4:2:2 to YCbCr 4:4:4
- (4). downsample for YCbCr 4:4:4 to YCbCr 4:2:2

24.3.2 Audio Data Processing

The HDMI TX audio process contain audio clock regeneration, placement of audio samples within packets, packet timing control, audio sample rates setting, and channel/speaker assignments. This function is implemented by Audio Capture blocks

The Audio Capture support either SPDIF or four channel I2S input. SPDIF input supports audio sampling rates from 32to 192 KHz. The I2S input supports from 2-channel to 8-channel audio up to 192 KHz.

The scheme of audio processing as shown in the figure below:

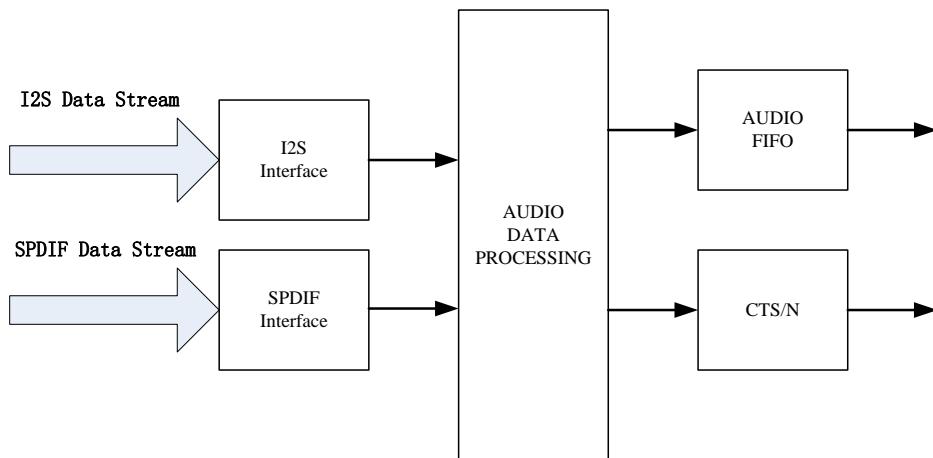


Fig. 24-4HDMI Audio Data Processing Diagram

1. I2S

The function of this module is to implement I2S audio input feature. The incoming audio stream is captured, processed then transmitted into the TMDS link. Four I2S inputs also allow transmission of DVD-Audio and decoded Dolby Digital to A/V Receivers and high-end displays. The interface supports from 2-channel to 8-channel audio up to 192 kHz. The I2S pins must also be coherent with mclk. The appropriate registers must be configured to describe the format of audio being input. This information is passed over the HDMI link in the CEA-861D Audio Info (AI) packets. Table shows the I2S 8 channel audio formats that are supported for each of the video formats.

Table 24-2HDMI TX I2S 2Channel Audio Sampling Frequency at Each Video Format

Video Format	32kH	44.1kH	48kH	88.2kH	96kH	176.4k Hz	192kH
720x480p /720x576p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1440x480i/1440x576i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 24-3HDMI TX I2S 8 Channel Audio Sampling Frequency at Each Video Format

Video Format	32kH	44.1kH	48kH	88.2kH	96kH	176.4k Hz	192kH
720x480p /720x576p	Yes	Yes	Yes	No	No	No	No
1440x480i/1440x576i	Yes	Yes	Yes	Yes	No	No	No
720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

2. SPDIF

The function of this module is to implement SPDIF audio input feature. The incoming audio stream is captured, processed then transmitted into the TMDS link. SPDIF stream can carry 2-channel uncompressed PCM data (IEC 60958) or a compressed bit stream for multi-channel (IEC 61937) formats. The audio data capture logic forms the audio data into packets in accordance with the HDMI specification. SPDIF input supports audio sampling rates from 32 to 192 KHz. The following shows the SPDIF audio formats that are supported for each of the video formats

Table 24-4HDMI SPDIF Sampling Frequency at Each Video Format

Video Format	32kHz	44.1kHz	48kHz	88.2kHz	96kHz	176.4kHz	192kHz
	z	z	z	z	z	Hz	z
720x480p /720x576p	Yes	Yes	Yes	Yes	Yes	No	No
1440x480i/1 440x576i	Yes	Yes	Yes	Yes	Yes	No	No
720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

3. Audio Sample Clock Capture and Regeneration

Audio data being carried across the HDMI link, which is driven by a TMDS clock running at a rate corresponding to the video pixel rate, does not retain the original audio sample clock. The task of recreating this clock at the Sink is called Audio Clock Regeneration.

The HDMI Transmitter determine the fractional relationship between the TMDS clock and an audio reference clock (128 audio sample rate [fs]) and pass the numerator and denominator of that fraction to the HDMI Sink across the HDMI link. The Sink then re-create the audio clock from the TMDS clock by using a clock divider and a clock multiplier.

The exact relationship between the two clocks will be.

$$128 \cdot f_s = f_{TMDS_clock} \cdot N / CTS.$$

The scheme of the Audio Sample Clock Capture and Regeneration as shown below:

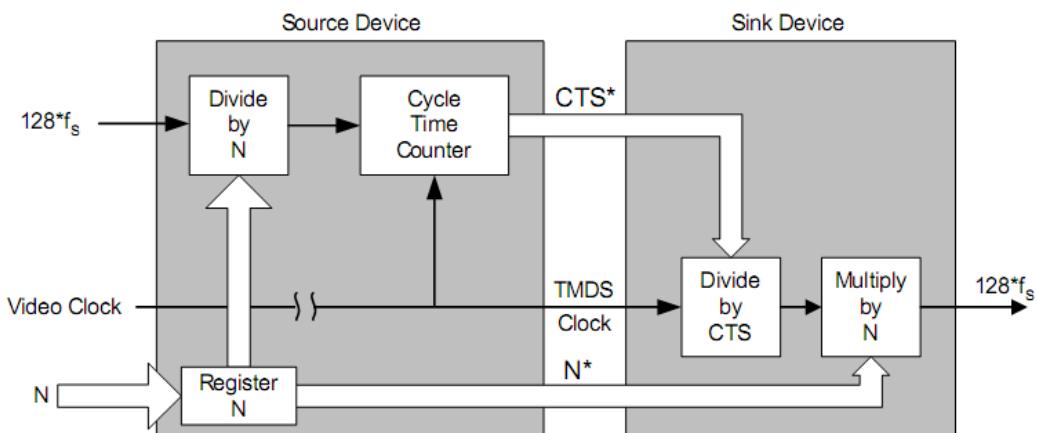


Fig. 24-5HDMI Audio Clock Regeneration Model

24.3.3 DDC

The DDC functional block is used for configuration and status exchange between the HDMI Source and HDMI Sink. HDMI TransmitterController has I2C Master Interface for DDC transactions. It enables for host controller to read EDID, HDCP authentication by issuing simple register access. The I2C bus speed is limited by DDC specification. DDC bus access frequency can be controlled.

24.3.4 EDID

Extended Display Identification Data (EDID) was created by VESA to enable plug and play capabilities of monitors. This data, which is stored in the sink device, describes video formats that the DTV Monitor is capable of receiving and rendering. The information is supplied to the source device, over the interface, upon the request of the source device. The source device then chooses its output format, taking into account the format of the original video stream and the formats supported by the DTV Monitor. The function of this module is to implement EDID feature.

24.3.5 HDCP

HDMI Transmitter has a capability for HDCP authentication by hardware. The function of this module is to implement HDCP encryption feature. This feature can be turned on or off depending on register setting.

24.3.6 Hot Plug Detect

HDMI Transmitter has a capability for detecting the Sink plug in or plug out, and launch an interrupt and registers state indicating for software controlling.

24.3.7 TMDS encoder

The TMDS encoder converts the 2/4/8 bits data into the 10 bit DC-balanced TMDS data.

HDMI TX put the TMDS encoding on the audio /video /aux data received from the HDCP XOR mask. This data is output onto three TMDS differential data lines along with a TMDS differential clock.

24.3.8 CEC

The CEC functional block provides high-level control functions between all of the various audiovisual products in a user's environment through one line.

24.3.9 Optional Lipsync

HDMI sinks and repeaters can now declare audio/video latency information in their EDIDs. HDMI TX can read the audio/video latency information from HDMI sinks, than it may delay audio or video to compensate for latencies in downstream devices.

24.4 Register Description

24.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
HDMI_reg00	0x0000	B	0x67	Register00
HDMI_reg01	0x0004	B	0x01	Register01
HDMI_reg02	0x0008	B	0x30	Register02
HDMI_reg04	0x0010	B	0x08	Register04
HDMI_reg05	0x0014	B	0x00	Register05
HDMI_reg08	0x0020	B	0x00	Register08
HDMI_reg09	0x0024	B	0x00	Register09
HDMI_reg0a	0x0028	B	0x00	Register0a
HDMI_reg0b	0x002c	B	0x00	Register0b
HDMI_reg0c	0x0030	B	0x00	Register0c
HDMI_reg0d	0x0034	B	0x00	Register0d
HDMI_reg0e	0x0038	B	0x00	Register0e
HDMI_reg0f	0x003c	B	0x00	Register0f
HDMI_reg10	0x0040	B	0x00	Register10
HDMI_reg11	0x0044	B	0x00	Register11
HDMI_reg12	0x0048	B	0x00	Register12
HDMI_reg13	0x004c	B	0x00	Register13
HDMI_reg14	0x0050	B	0x00	Register14
HDMI_reg15	0x0054	B	0x00	Register15
HDMI_reg35	0x00d4	B	0x01	Register35
HDMI_reg37	0x00dc	B	0x00	Register37
HDMI_reg38	0x00e0	B	0x3c	Register38
HDMI_reg39	0x00e4	B	0x00	Register39
HDMI_reg3a	0x00e8	B	0x00	Register3a
HDMI_reg3f	0x00fc	B	0x00	Register3f
HDMI_reg40	0x0100	B	0x18	Register40
HDMI_reg41	0x0104	B	0x00	Register41
HDMI_reg45	0x0114	B	0x00	Register45
HDMI_reg46	0x0118	B	0x00	Register46
HDMI_reg47	0x011c	B	0x00	Register47
HDMI_reg4a	0x0128	B	0x00	Register4a
HDMI_reg4b	0x012c	B	0x40	Register4b
HDMI_reg4c	0x0130	B	0x00	Register4c
HDMI_reg4d	0x0134	B	0x00	Register4d
HDMI_reg4e	0x0138	B	0x00	Register4e
HDMI_reg4f	0x013c	B	0x00	Register4f
HDMI_reg50	0x0140	B	0x00	Register50
HDMI_reg52	0x0148	B	0x12	Register52
HDMI_reg53	0x014c	B	0x04	Register53
HDMI_reg57	0x015c	B	0x20	Register57
HDMI_reg58	0x0160	B	0x00	Register58
HDMI_reg63	0x018c	B	0x26	Register63
HDMI_reg65	0x0194	B	0x00	Register65
HDMI_reg66	0x0198	B	0x00	Register66
HDMI_reg67	0x019c	B	0x00	Register67
HDMI_reg68	0x01a0	B	0x00	Register68

Name	Offset	Size	Reset Value	Description
HDMI_reg69	0x01a4	B	0x00	Register69
HDMI_reg6a	0x01a8	B	0x00	Register6a
HDMI_reg6c	0x01b0	B	0x00	Register6c
HDMI_reg95	0x0254	B	0x00	Register95
HDMI_reg96	0x0258	B	0x00	Register96
HDMI_reg97	0x025c	B	0x00	Register97
HDMI_reg98	0x0260	B	0x03	Register98
HDMI_reg9c	0x0270	B	0x00	Register9c
HDMI_reg9e	0x0278	B	0x01	Register9e
HDMI_reg9f	0x027c	B	0x00	Register9f
HDMI_rega0	0x0280	B	0x00	Registera0
HDMI_rega1	0x0284	B	0x00	Registera1
HDMI_rega2	0x0288	B	0x00	Registera2
HDMI_rega3	0x028c	B	0x00	Registera3
HDMI_rega4	0x0290	B	0x00	Registera4
HDMI_rega5	0x0294	B	0x00	Registera5
HDMI_rega6	0x0298	B	0x00	Registera6
HDMI_rega7	0x029c	B	0x00	Registera7
HDMI_rega8	0x02a0	B	0x00	Registera8
HDMI_rega9	0x02a4	B	0x00	Registera9
HDMI_regaa	0x02a8	B	0x00	Registeraa
HDMI_regab	0x02ac	B	0x00	Registerab
HDMI_regac	0x02b0	B	0x00	Registerac
HDMI_regad	0x02b4	B	0x00	Registerad
HDMI_regae	0x02b8	B	0x00	Registerae
HDMI_regaf	0x02bc	B	0x00	Registeraf
HDMI_regb0	0x02c0	B	0x00	Registerb0
HDMI_regb1	0x02c4	B	0x00	Registerb1
HDMI_regb2	0x02c8	B	0x00	Registerb2
HDMI_regb3	0x02cc	B	0x00	Registerb3
HDMI_regb4	0x02d0	B	0x00	Registerb4
HDMI_regb5	0x02d4	B	0x00	Registerb5
HDMI_regb6	0x02d8	B	0x00	Registerb6
HDMI_regb7	0x02dc	B	0x00	Registerb7
HDMI_regb8	0x02e0	B	0x00	Registerb8
HDMI_regb9	0x02e4	B	0x00	Registerb9
HDMI_regba	0x02e8	B	0x00	Registerba
HDMI_regbb	0x02ec	B	0x00	Registerbb
HDMI_regbc	0x02f0	B	0x00	Registerbc
HDMI_regbd	0x02f4	B	0x00	Registerbd
HDMI_refbe	0x02f8	B	0x00	Registerbe
HDMI_regc0	0x0300	B	0xc0	Registerc0
HDMI_regc1	0x0304	B	0x00	Registerc1
HDMI_regc2	0x0308	B	0x78	Registerc2
HDMI_regc3	0x030c	B	0x00	Registerc3
HDMI_regc4	0x0310	B	0x00	Registerc4
HDMI_regc5	0x0314	B	0x00	Registerc5
HDMI_regc8	0x0320	B	0x00	Registerc8
HDMI_regc9	0x0324	B	0x50	Registerc9

Name	Offset	Size	Reset Value	Description
HDMI_regce	0x0338	B	0x01	Registerce
HDMI_regd0	0x0340	B	0x00	Registerd0
HDMI_regd1	0x0344	B	0x00	Registerd1
HDMI_regd2	0x0348	B	0x00	Registerd2
HDMI_regd3	0x034c	B	0x00	Registerd3
HDMI_regd4	0x0350	B	0x03	Registerd4
HDMI_regd5	0x0354	B	0x09	Registerd5
HDMI_regd6	0x0358	B	0x03	Registerd6
HDMI_regd7	0x035c	B	0x00	Registered7
HDMI_regd8	0x0360	B	0xff	Registerd8
HDMI_regd9	0x0364	B	0xff	Registerd9
HDMI_regda	0x0368	B	0x00	Registerda
HDMI_regdb	0x036c	B	0x00	Registerdb
HDMI_regdc	0x0370	B	0xd0	Registerdc
HDMI_regdd	0x0374	B	0x20	Registerdd
HDMI_regede	0x0378	B	0x00	Registerde
HDMI_rege0	0x0380	B	0x23	Registere0
HDMI_rege1	0x0384	B	0x0f	Registere1
HDMI_rege2	0x0388	B	0xaa	Registere2
HDMI_rege3	0x038c	B	0x0f	Registere3
HDMI_rege7	0x039c	B	0x1e	Registere7
HDMI_rege8	0x03a0	B	0x00	Registere8
HDMI_reged	0x03b4	B	0x03	Registered

24.4.2 Detail Register Description

HDMI_reg00

Address: Operational Base + offset (0x0000)

Register00

Bit	Attr	Reset Value	Description
7	RW	0x0	reserved
6	RW	0x1	sw_reset_ana Soft reset for analog part 1'b0: reset 1'b1: not
5	RW	0x1	sw_reset_dig Soft reset for digital part 1'b0: reset 1'b1: not
4	RW	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	vclk_inv Vclk invert select 1'b0: not invert 1'b1: invert
2	RW	0x0	reserved
1	RW	0x1	pd_dig System power down for digital function 1'b0: not 1'b1: power down
0	RW	0x1	interrupt_polarity Interrupt polarity 1'b1: Active High 1'b0: Active low

HDMI_reg01

Address: Operational Base + offset (0x0004)

Register01

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved
3:1	RW	0x0	input_video_fmt Input video format 3'b000: RGB and YCbCr 4:4:4 3'b101: DDR RGB 4:4:4 or YCbCr 4:4:4 3'b110: DDR YCbCr 4:2:2
0	RW	0x1	de_sel DE select 1'b0: internal DE 1'b1: external DE

HDMI_reg02

Address: Operational Base + offset (0x0008)

Register02

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
7:6	RW	0x0	video_output_fmt Video output format 2'b00: RGB 4:4:4 2'b01: YCbCr 4:4:4 2'b10: YCbCr 4:2:2
5:4	RW	0x3	data_width_422 Data width for 4:2:2 input 2'b00: 12 bits 2'b01: 10 bits 2'b11: 8 bits
3:1	RW	0x0	reserved
0	RW	0x0	Video_in_color_space Video input color space 1'b0: RGB 1'b1: YCbCr

HDMI_reg04

Address: Operational Base + offset (0x0010)

Register04

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved
3	RW	0x1	sof_sel After 1st SOF(the first edge of the Vsync) for external DE sample 1'b0: after SOF 1'b1: Not
2:1	RW	0x0	reserved
0	RW	0x0	csc_en CSC (Color Space Convert) enable. 1'b0: no CSC 1'b1: enable CSC

HDMI_reg05

Address: Operational Base + offset (0x0014)

Register05

Bit	Attr	Reset Value	Description
7	RW	0x0	clear_avmute Clear avmute

Bit	Attr	Reset Value	Description
6	RW	0x0	set_avmute Set avmute
5:2	RW	0x0	reserved
1	RW	0x0	audio_mute Audio mute
0	RW	0x0	video_black Video black

HDMI_reg08

Address: Operational Base + offset (0x0020)

Register08

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved
3	RW	0x0	vs_polarity VSYNC polarity 1'b0: Negative 1'b1: Positive
2	RW	0x0	hs_polarity HSYNC polarity 1'b0: Negative 1'b1: Positive
1	RW	0x0	interlace_progressiv_sel Interlace/Progressive 1'b0: Progressive 1'b1: Interlace
0	RW	0x0	ext_video_para_set_en External video parameter setting enable 1'b0: disable 1'b1: enable

HDMI_reg09

Address: Operational Base + offset (0x0024)

Register09

Bit	Attr	Reset Value	Description
7:0	RW	0x00	ext_hs_total_part1 External horizontal total part1. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg0a

Address: Operational Base + offset (0x0028)

Register0a

Bit	Attr	Reset Value	Description
7:5	RW	0x0	reserved
4:0	RW	0x00	ext_hs_total_part2 External horizontal total part2. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg0b

Address: Operational Base + offset (0x002c)

Register0b

Bit	Attr	Reset Value	Description
7:0	RW	0x00	ext_hs_blank_part1 External horizontal blank part1. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg0c

Address: Operational Base + offset (0x0030)

Register0c

Bit	Attr	Reset Value	Description
7:3	RW	0x0	reserved
2:0	RW	0x0	ext_hs_blank_part2 External horizontal blank part2. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg0d

Address: Operational Base + offset (0x0034)

Register0d

Bit	Attr	Reset Value	Description
7:0	RW	0x00	Ext_hsync_part1 External horizontal hsync part1. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg0e

Address: Operational Base + offset (0x0038)

Register0e

Bit	Attr	Reset Value	Description
7:2	RW	0x0	reserved
1:0	RW	0x0	Ext_hsync_part2 External horizontal hsync part2. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg0f

Address: Operational Base + offset (0x003c)

Register0f

Bit	Attr	Reset Value	Description
7:0	RW	0x00	ext_hs_sync_part1 External horizontal duration part1. It is not need to be set at normal resolution. Just for special resolution or test purpose

HDMI_reg10

Address: Operational Base + offset (0x0040)

Register10

Bit	Attr	Reset Value	Description
7:2	RW	0x0	reserved
1:0	RW	0x0	ext_hs_sync_part2 External horizontal duration part2. It is not need to be set at normal resolution. Just for special resolution or test purpose

HDMI_reg11

Address: Operational Base + offset (0x0044)

Register11

Bit	Attr	Reset Value	Description
7:0	RW	0x00	ext_vertical_total_part1 External vertical total part1. It is not need to be set at normal resolution. Just for special resolution or test purpose

HDMI_reg12

Address: Operational Base + offset (0x0048)

Register12

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	ext_vertical_total_part2 External vertical total part2. It is not need to be set at normal resolution. Just for special resolution or test purpose

HDMI_reg13

Address: Operational Base + offset (0x004c)

Register13

Bit	Attr	Reset Value	Description
7	RW	0x0	reserved
6:0	RW	0x00	ext_vertical_blank External vertical blank. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg14

Address: Operational Base + offset (0x0050)

Register14

Bit	Attr	Reset Value	Description
7	RW	0x0	reserved
6:0	RW	0x00	ext_vertical_vsync External vertical vsync. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg15

Address: Operational Base + offset (0x0054)

Register15

Bit	Attr	Reset Value	Description
7:6	RW	0x0	reserved
5:0	RW	0x00	ext_vertical_duration External vertical duration. It is not need to be set at normal resolution. Just for special resolution or test purpose.

HDMI_reg35

Address: Operational Base + offset (0x00d4)

Register35

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
7	RW	0x0	cts_sel CTS source select 1'b0: internal CTS 1'b1: external CTS
6:5	RW	0x0	reserved
4:3	RW	0x0	audio_type_sel Audio type select 2'b00: I2S 2'b01: S/PDIF
2	RW	0x0	mclk_sel MCLK select
1:0	RW	0x1	mclk_ratio MCLK ratio 2'b00: 128 fs 2'b01: 256 fs 2'b10: 384 fs 2'b11: 512 fs

HDMI_reg37

Address: Operational Base + offset (0x00dc)

Register37

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved
3:0	RW	0x0	sample_freq_sel Sampling frequency for I2S audio 4'b0011: 32 kHz 4'b0000: 44.1 kHz 4'b0010: 48 kHz 4'b1000: 88.2 kHz 4'b1010: 96 kHz 4'b1100: 176.4 kHz 4'b1110: 192 kHz

HDMI_reg38

Address: Operational Base + offset (0x00e0)

Register38

Bit	Attr	Reset Value	Description
7:6	RW	0x0	reserved

Bit	Attr	Reset Value	Description
5:2	RW	0xf	i2s_sel I2S enable for the four I2S pins 4'b0001: I2S0 4'b0010: I2S1 4'b0100: I2S2 4'b1000: I2S3
1:0	RW	0x0	i2s_fmt I2S format 2'b00: standard I2S mode 2'b01: right-justified I2S mode 2'b10: left-justified I2S mode

HDMI_reg39

Address: Operational Base + offset (0x00e4)

Register39

Bit	Attr	Reset Value	Description
7:6	RW	0x0	audio_channel3_input_sel Audio channel3 input 2'b00: I2S3 2'b01: I2S2 2'b10: I2S1 2'b11: I2S0
5:4	RW	0x0	audio_channel2_input_sel Audio channel2 input 2'b00: I2S2 2'b01: I2S1 2'b10: I2S0 2'b11: I2S3
3:2	RW	0x0	audio_channel1_input_sel Audio channel1 input 2'b00: I2S1 2'b01: I2S0 2'b10: I2S3 2'b11: I2S2
1:0	RW	0x0	audio_channel0_input_sel Audio channel0 input 2'b00: I2S0 2'b01: I2S3 2'b10: I2S2 2'b11: I2S1

HDMI_reg3a

Address: Operational Base + offset (0x00e8)

Register3a

Bit	Attr	Reset Value	Description
7	RW	0x0	lr_swap_ch7_8 Left/Right data swap for ch7/8
6	RW	0x0	lr_swap_ch5_6 Left/Right data swap for ch5/6
5	RW	0x0	lr_swap_ch3_4 Left/Right data swap for ch3/4
4	RW	0x0	lr_swap_ch1_2 Left/Right data swap for ch1/2
3:0	RW	0x0	spdif_sample_freq S/PDIF sampling frequency 4'b0011: 32 kHz. 4'b0000: 44.1 kHz. 4'b0010: 48 kHz. 4'b1000: 88.2 kHz. 4'b1010: 96 kHz. 4'b1100: 176.4 kHz. 4'b1110: 192 kHz.

HDMI_reg3f

Address: Operational Base + offset (0x00fc)

Register3f

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved
3:0	RW	0x0	ncts_part1 20-bit N-CTS Value for audio clock generation at Sink end. N-CTS=reg3f[3:0],reg40[7:0],reg41[7:0] Use this function please refer to the HDMI specification at "Recommended N and Expected CTS Values"

HDMI_reg40

Address: Operational Base + offset (0x0100)

Register40

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
7:0	RW	0x18	ncts_part2 20-bit N-CTS Value for audio clock generation at Sink end. N-CTS=reg3f[3:0],reg40[7:0],reg41[7:0] Use this function please refer to the HDMI specification at "Recommended N and Expected CTS Values"

HDMI_reg41

Address: Operational Base + offset (0x0104)

Register41

Bit	Attr	Reset Value	Description
7:0	RW	0x00	ncts_part3 20-bit N-CTS Value for audio clock generation at Sink end. N-CTS=reg3f[3:0],reg40[7:0],reg41[7:0] Use this function please refer to the HDMI specification at "Recommended N and Expected CTS Values"

HDMI_reg45

Address: Operational Base + offset (0x0114)

Register45

Bit	Attr	Reset Value	Description
7:4	RW	0x00	reserved
3:0	RW	0x00	cts_part3 When use external CTS (reg0x35[7]=1),then set these three regs. CTS = {reg45[3:0], reg46[7:0], reg47[7:0]}

HDMI_reg46

Address: Operational Base + offset (0x0118)

Register46

Bit	Attr	Reset Value	Description
7:0	RW	0x00	cts_part3 When use external CTS (reg0x35[7]=1),then set these three regs. CTS = {reg45[3:0], reg46[7:0], reg47[7:0]}

HDMI_reg47

Address: Operational Base + offset (0x011c)

Register47

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>cts_part3</p> <p>When use external CTS (reg0x35[7]=1),then set these three regs.</p> <p>CTS = {reg45[3:0], reg46[7:0], reg47[7:0]}</p>

HDMI_reg4a

Address: Operational Base + offset (0x0128)

Register4a

Bit	Attr	Reset Value	Description
7:2	RW	0x0	reserved
1	RO	0x0	<p>ddc_sda_bus_statue</p> <p>DDC SDA bus status</p>
0	RO	0x0	<p>ddc_scl_bus_statue</p> <p>DDC SCL bus status</p>

HDMI_reg4b

Address: Operational Base + offset (0x012c)

Register4b

Bit	Attr	Reset Value	Description
7:0	RW	0x40	<p>ddc_bus_access_freq_lsb</p> <p>DDC bus access frequency (LSB)</p>

HDMI_reg4c

Address: Operational Base + offset (0x0130)

Register4c

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>ddc_bus_access_freq_msb</p> <p>DDC bus access frequency (MSB)</p>

HDMI_reg4d

Address: Operational Base + offset (0x0134)

Register4d

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>edid_segment_pointer</p> <p>EDID segment pointer</p>

HDMI_reg4e

Address: Operational Base + offset (0x0138)

Register4e

Bit	Attr	Reset Value	Description
7:0	RW	0x00	edid_word_addr EDID word address

HDMI_reg4f

Address: Operational Base + offset (0x013c)

Register4f

Bit	Attr	Reset Value	Description
7:0	RW	0x00	edid_fifo_rd_addr EDID FIFO read address

HDMI_reg50

Address: Operational Base + offset (0x0140)

Register50

Bit	Attr	Reset Value	Description
7:0	RO	0x00	edid_rd_access_window EDID reading access window

HDMI_reg52

Address: Operational Base + offset (0x0148)

Register52

Bit	Attr	Reset Value	Description
7	RW	0x0	authentication_start Authentication start
6	RW	0x0	bksv_not_in_blacklist BKSV is not in the blacklist
5	RW	0x0	bksv_in_blacklist BKSV is in the blacklist
4	RW	0x1	encryte_en current frame encrypted en 1'b0: current frame should not be encrypted 1'b1: current frame should be encrypted
3	RW	0x0	authentication_stop Authentication Stop
2	RW	0x0	advanced_mode_sel 1'b0: do not use advanced cipher mode 1'b1: use advanced cipher mode

Bit	Attr	Reset Value	Description
1	RW	0x1	mode_sel mode selection 1'b0: DVI mode 1'b1: HDMI mode
0	RW	0x0	hdcp_reset HDCP reset

HDMI_reg53

Address: Operational Base + offset (0x014c)

Register53

Bit	Attr	Reset Value	Description
7	RW	0x0	disable_127_chk Disable 127 check
6	RW	0x0	skip_bksv_blacklist_chk Skip BKSV blacklist check
5	RW	0x0	pj_chk_en Enable Pj check
4	RW	0x0	disable_dev_num_check Disable device number check
3	RW	0x1	dly_ri_chk Delay Ri check by one clock
2	RW	0x0	use_preset Use preset An value
1:0	RW	0x0	key_comb Key combination

HDMI_reg57

Address: Operational Base + offset (0x015c)

Register57

Bit	Attr	Reset Value	Description
7	RW	0x0	Authenticated_en 1'b0: Authenticated 1'b0: Not authenticated
6	RW	0x0	av_encrypte_en AV content is encrypted enable 1'b0: AV content is not encrypted 1'b1: AV content is encrypted

Bit	Attr	Reset Value	Description
5	RW	0x1	reserved
4	RW	0x0	HDCP_work 1'b0: HDCP is working 1'b1: HDCP is not working
3	RW	0x0	advanced_cipher_en advanced cipher enable 1'b0: Advanced cipher is not enabled 1'b1: Advanced cipher is enabled
2:0	RW	0x0	reserved

HDMI_reg58

Address: Operational Base + offset (0x0160)

Register58

Bit	Attr	Reset Value	Description
7:0	RW	0x00	rx_bcaps_value RX Bcaps value

HDMI_reg63

Address: Operational Base + offset (0x018c)

Register63

Bit	Attr	Reset Value	Description
7:5	RW	0x00	reserved
4:0	RW	0x06	timer_100ms Registers for timer 100ms

HDMI_reg65

Address: Operational Base + offset (0x0194)

Register65

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved
3	RW	0x0	ddc_channel_no_ack DDC channels no acknowledge
2	RW	0x0	pj_mismatch Pj mismatch
1	RW	0x0	ri_mismatch Ri mismatch
0	RW	0x0	bksv_wrong Bksv is wrong

HDMI_reg66

Address: Operational Base + offset (0x0198)

Register66

Bit	Attr	Reset Value	Description
7:0	RW	0x00	Bksv value with least significant byte first

HDMI_reg67

Address: Operational Base + offset (0x019c)

Register67

Bit	Attr	Reset Value	Description
7:0	RW	0x00	Bksv value with least significant byte first

HDMI_reg68

Address: Operational Base + offset (0x01a0)

Register68

Bit	Attr	Reset Value	Description
7:0	RW	0x00	Bksv value with least significant byte first

HDMI_reg69

Address: Operational Base + offset (0x01a4)

Register69

Bit	Attr	Reset Value	Description
7:0	RW	0x00	Bksv value with least significant byte first

HDMI_reg6a

Address: Operational Base + offset (0x01a8)

Register6a

Bit	Attr	Reset Value	Description
7:0	RW	0x00	Bksv value with least significant byte first

HDMI_reg6c

Address: Operational Base + offset (0x01b0)

Register6c

Bit	Attr	Reset Value	Description
7:0	RW	0x00	seed_an An seed for generate An

HDMI_reg95

Address: Operational Base + offset (0x0254)

Register95

Bit	Attr	Reset Value	Description
7:0	RW	0x00	hdcp_fifo_first_wr_byte_addr HDCP KEY FIFO first write byte num address

HDMI_reg96

Address: Operational Base + offset (0x0258)

Register96

Bit	Attr	Reset Value	Description
7:0	RW	0x00	hdcp_fifo_first_rd_byte_addr HDCP KEY FIFO first read byte num address

HDMI_reg97

Address: Operational Base + offset (0x025c)

Register97

Bit	Attr	Reset Value	Description
7:2	RO	0x0	reserved
1	RW	0x0	hdcp_fifo_rd_addr8 HDCP KEY FIFO first read byte num address[8]
0	RW	0x0	hdcp_fifo_wr_addr8 HDCP KEY FIFO first write byte num address[8]

HDMI_reg98

Address: Operational Base + offset (0x0260)

Register98

Bit	Attr	Reset Value	Description
7:0	RW	0x03	hdcp_fifo_wr_rd_addr HDCP KEY FIFO write or read point address

HDMI_reg9c

Address: Operational Base + offset (0x0270)

Register9c

Bit	Attr	Reset Value	Description
7	RW	0x0	general_control_packet General control packet 1'b1:enable 1'b0:disable
6	RW	0x0	mpeg_source_frame_packet MPEG source InfoFrame packet 1'b1:enable 1'b0:disable

Bit	Attr	Reset Value	Description
5	RW	0x0	source_descriptor_frame_packet Source product descriptor InfoFrame packet 1'b1:enable 1'b0:disable
4	RW	0x0	vendor_specific_frame_packet Vendor specific InfoFrame packet 1'b1:enable 1'b0:disable
3	RW	0x0	gamut_metadata_packet Gamut metadata packet 1'b1:enable 1'b0:disable
2	RW	0x0	isrc1_packet ISRC1 packet 1'b1:enable 1'b0:disable
1	RW	0x0	acp_packet ACP packet 1'b1:enable 1'b0:disable
0	RW	0x0	generic_packet Generic packet 1'b1:enable 1'b0:disable

HDMI_reg9e

Address: Operational Base + offset (0x0278)

Register9e

Bit	Attr	Reset Value	Description
7:1	RW	0x0	reserved
0	RW	0x1	auto_checksum_frame Auto checksum for InfoFrame 1'b1:enable 1'b0:disable It enables checksum calculation for any InfoFrame packets by hardware.

HDMI_reg9f

Address: Operational Base + offset (0x027c)

Register9f

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>packet_offset</p> <p>packet offset register for 0xa0~0xbe</p> <p>8'h0: Generic packet</p> <p>8'h1: ACP packet</p> <p>8'h2: ISRC1 packet</p> <p>8'h3: ISRC2 packet</p> <p>8'h4: Gamut metadata packet</p> <p>8'h5: Vendor specific InfoFrame</p> <p>8'h6: AVI InfoFrame</p> <p>8'h7: Source product descriptor InfoFrame packet</p> <p>8'h8: Audio InfoFrame packet</p> <p>8'h9: MPEG source InfoFrame</p>

HDMI_rega0

Address: Operational Base + offset (0x0280)

Registera0

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet</p> <p>ACP packet</p> <p>ISRC1 packet</p> <p>ISRC2 packet</p> <p>Gamut metadata packet</p> <p>Vendor specific InfoFrame packet</p> <p>AVI InfoFrame packet</p> <p>Source product descriptor InfoFrame packet</p> <p>Audio InfoFrame packet</p> <p>MPEG source InfoFrame packet</p>

HDMI_rega1

Address: Operational Base + offset (0x0284)

Registera1

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_rega2

Address: Operational Base + offset (0x0288)

Registera2

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_rega3

Address: Operational Base + offset (0x028c)

Registera3

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_rega4

Address: Operational Base + offset (0x0290)

Registera4

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_rega5

Address: Operational Base + offset (0x0294)

Registera5

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_rega6

Address: Operational Base + offset (0x0298)

Registera6

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_rega7

Address: Operational Base + offset (0x029c)

Registera7

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_rega8

Address: Operational Base + offset (0x02a0)

Registera8

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_rega9

Address: Operational Base + offset (0x02a4)

Registera9

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regaa

Address: Operational Base + offset (0x02a8)

Registeraa

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regab

Address: Operational Base + offset (0x02ac)

Registerab

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regac

Address: Operational Base + offset (0x02b0)

Registerac

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regad

Address: Operational Base + offset (0x02b4)

Registerad

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regae

Address: Operational Base + offset (0x02b8)

Registerae

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regaf

Address: Operational Base + offset (0x02bc)

Registeraf

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb0

Address: Operational Base + offset (0x02c0)

Registerb0

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb1

Address: Operational Base + offset (0x02c4)

Registerb1

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb2

Address: Operational Base + offset (0x02c8)

Registerb2

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb3

Address: Operational Base + offset (0x02cc)

Registerb3

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb4

Address: Operational Base + offset (0x02d0)

Registerb4

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb5

Address: Operational Base + offset (0x02d4)

Registerb5

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb6

Address: Operational Base + offset (0x02d8)

Registerb6

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb7

Address: Operational Base + offset (0x02dc)

Registerb7

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb8

Address: Operational Base + offset (0x02e0)

Registerb8

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb9

Address: Operational Base + offset (0x02e4)

Registerb9

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regba

Address: Operational Base + offset (0x02e8)

Registerba

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regb

Address: Operational Base + offset (0x02ec)

Registerbb

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regbc

Address: Operational Base + offset (0x02f0)

Registerbc

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regbd

Address: Operational Base + offset (0x02f4)

Registerbd

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regbe

Address: Operational Base + offset (0x02f8)

Registerbe

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>After configure the packet offset register, these registers can configure any packet shown following.</p> <p>Use this function please refer to the HDMI specification at "Data Island Packet Definitions"</p> <p>Generic packet ACP packet ISRC1 packet ISRC2 packet Gamut metadata packet Vendor specific InfoFrame packet AVI InfoFrame packet Source product descriptor InfoFrame packet Audio InfoFrame packet MPEG source InfoFrame packet</p>

HDMI_regc0

Address: Operational Base + offset (0x0300)

Registerc0

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>mask_int</p> <p>Mask for Interrupt</p> <p>1'b0: masked 1'b1: not mask</p> <p>[5] Interrupt for active Vsync edge [2] Interrupt for EDID Ready</p>

HDMI_regc1

Address: Operational Base + offset (0x0304)

Registerc1

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>int_statu</p> <p>Interrupt status</p> <p>[5] Interrupt for active Vsync edge</p> <p>This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1.</p> <p>Clear interrupt: write 1 to this bit.</p> <p>[2] Interrupt for EDID Ready</p> <p>This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1.</p> <p>Clear interrupt: write 1 to this bit.</p>

HDMI_regc2

Address: Operational Base + offset (0x0308)

Registerc2

Bit	Attr	Reset Value	Description
7:3	RW	0x0f	<p>int_mask</p> <p>Mask for interrupt:</p> <p>1'b0:masked 1'b1:not masked</p> <p>[7] for HDCP error [6] for bksv fifo ready [5] for bksv update [4] for authentication success [3] for ready to start authentication</p>
2:0	RW	0x0	reserved

HDMI_regc3

Address: Operational Base + offset (0x030c)

Registerc3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
7:3	RW	0x00	<p>int_status Interrupt status [7] HDCP error Interrupt This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[6] bksv fifo ready This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[5] bksv update This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[4] authentication success This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[3] ready to start authentication This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p>
2:0	RW	0x0	reserved

HDMI_regc4

Address: Operational Base + offset (0x0310)

Registerc4

Bit	Attr	Reset Value	Description
7	RW	0x00	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x00	<p>int_mask</p> <p>Mask for interrupt:</p> <p>1'b0:masked</p> <p>1'b1:not masked</p> <p>[6] for HDCP authentication M0 ready</p> <p>[5] for first frame arrive</p> <p>[4] for HDCP An ready</p> <p>[3]</p> <p>[2] for HDCP encrypted</p> <p>[1] for HDCP not encrypted (av mute)</p> <p>[0] for HDCP not encrypted (no av mute)</p>

HDMI_regc5

Address: Operational Base + offset (0x0314)

Registersc5

Bit	Attr	Reset Value	Description
7	RW	0x00	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x00	<p>int_status_group3 Interrupt status Group3 This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[6] HDCP authentication M0 ready This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[5] irst frame arrive This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[4] HDCP An ready This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[2] HDCP encrypted This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[1] HDCP not encrypted (av mute) This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p> <p>[0] HDCP not encrypted (no av mute) This bit will active when the mask for this bit to be set 1. When Interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.</p>

HDMI_regc8

Address: Operational Base + offset (0x0320)

Registersc8

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
7	RO	0x0	hot_plug_pin_status Hot plug pin status 1'b1: Plug in 1'b0: Plug out
6	RO	0x0	reserved
5	RW	0x1	Mask_hpd_int Mask for hot plug detect(HPG) interrupt 1'b0: masked 1'b1: not mask
4:2	RW	0x4	reserved
1	RW	0x0	Int_hpd Interrupt for hot plug detect(HPD) This bit will active when the mask for this bit to be set 1. When interrupt occurs, this bit will be equal to 1. Clear interrupt: write 1 to this bit.
0	RW	0x0	reserved

HDMI_regc9

Address: Operational Base + offset (0x0324)

Registerc9

Bit	Attr	Reset Value	Description
7:6	RW	0x1	video_bist_mode Video BIST mode 2'b00: normal color bar. 2'b01: special color bar. 2'b10: black
5	RW	0x0	flush_en Flush enable. After enable it the screen will flush from color bar to black again and again. 1'b1: enable 1'b0: disable
4	RW	0x1	disable_colorbar_bist_test Disable color bar BIST test 1'b1: disable 1'b0: enable
3:0	RW	0x0	reserved

HDMI_regce

Address: Operational Base + offset (0x0338)

Registerce

Bit	Attr	Reset Value	Description
7:1	RW	0x0	reserved
0	RW	0x1	<p>tmds_channel_sync</p> <p>The synchronization for TMDS channels</p> <p>The synchronization for TMDS channels is automatic after Reset the HDMI. But you can synchronization for TMDS channels manually.</p> <p>The operation for this function is:</p> <p>first send 0 to this bit, then send 1, and keep 1 into this bit.</p>

HDMI_regd0

Address: Operational Base + offset (0x0340)

Registerd0

Bit	Attr	Reset Value	Description
7	RW	0x0	reserved
6	RW	0x0	<p>modify_start_bit_timing_hisense_tv</p> <p>Modify the start bit timing for Hisense TV</p> <p>1'b1: Modify the start bit timing for Hisense TV</p> <p>1'b0: Not modify</p>
5	RW	0x0	<p>reject_rec_cec_broadcast_message</p> <p>Reject receive CEC broadcast message</p> <p>1'b1: Reject receive CEC broadcast message</p> <p>1'b0: Not reject</p>
4:3	RW	0x0	reserved
2	RW	0x0	<p>CEC_bus_free_time_cnt_en</p> <p>Enable count CEC bus free time</p> <p>1'b1: Enable count CEC bus free time</p> <p>1'b0: Not enable</p>
1	RW	0x0	<p>forbid_receive_cec_message</p> <p>Forbid receive CEC message</p> <p>1'b1: Forbid receive CEC message</p> <p>1'b0: Not forbid</p>
0	RW	0x0	<p>start_transmit_cec_message</p> <p>Start transmit CEC message</p> <p>1'b1: Start transmit CEC message</p> <p>1'b0: Not start</p>

HDMI_regd1

Address: Operational Base + offset (0x0344)

Registerd1

Bit	Attr	Reset Value	Description
7:0	RW	0x00	cec_rx_tx_data_in_fifo CEC RX/TX data in FIFO

HDMI_regd2

Address: Operational Base + offset (0x0348)

Registerd2

Bit	Attr	Reset Value	Description
7:5	RW	0x0	reserved
4:0	RW	0x00	wr_cec_tx_data_addr Point address for write CEC TX data

HDMI_regd3

Address: Operational Base + offset (0x034c)

Registerd3

Bit	Attr	Reset Value	Description
7:5	RW	0x0	reserved
4:0	RW	0x00	cec_rx_data_addr Point address for CEC RX data

HDMI_regd4

Address: Operational Base + offset (0x0350)

Registerd4

Bit	Attr	Reset Value	Description
7:0	RW	0x03	cec_clk_freq CEC working clock frequency register1

HDMI_regd5

Address: Operational Base + offset (0x0354)

Registerd5

Bit	Attr	Reset Value	Description
7:0	RW	0x09	cec_work_clk_freq CEC working clock frequency register1

HDMI_regd6

Address: Operational Base + offset (0x0358)

Registerd6

Bit	Attr	Reset Value	Description
7:5	RW	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x03	tx_block_length Length of TX blocks

HDMI_regd7

Address: Operational Base + offset (0x035c)

Registerd7

Bit	Attr	Reset Value	Description
7:5	RW	0x0	reserved
4:0	RW	0x00	rx_block_length Length of RX blocks

HDMI_regd8

Address: Operational Base + offset (0x0360)

Registerd8

Bit	Attr	Reset Value	Description
7:0	RW	0xff	cec_tx_int_mask Mask for CEC TX Interrupts 1'b0: masked 1'b1: not mask [3] Interrupt for TX done [2] Interrupt for TX no ACK from follower [1] Interrupt for TX broadcast rejected [0] Interrupt for CEC line free time not satisfied

HDMI_regd9

Address: Operational Base + offset (0x0364)

Registerd9

Bit	Attr	Reset Value	Description
7:0	RW	0xff	cec_rx_int_mask Mask for CEC RX Interrupts 1'b0: masked 1'b1: not mask [4] Interrupt for RX receive logic address error [3] Interrupt for RX receive glitch error [2] Interrupt for RX ACK detect overtime [1] Interrupt for RX sending broadcast reject [0] Interrupt for RX done

HDMI_rega

Address: Operational Base + offset (0x0368)

Registerda

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>cec_tx_int</p> <p>CEC TX Interrupts</p> <ul style="list-style-type: none"> [3] Interrupt for TX done [2] Interrupt for TX no ACK from follower [1] Interrupt for TX broadcast rejected [0] Interrupt for CEC line free time not satisfied

HDMI_regdb

Address: Operational Base + offset (0x036c)

Registerdb

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>cec_tx_int</p> <p>CEC RX Interrupts</p> <ul style="list-style-type: none"> [4] Interrupt for RX receive logic address error [3] Interrupt for RX receive glitch error [2] Interrupt for RX ACK detect overtime [1] Interrupt for RX sending broadcast reject [0] Interrupt for RX done

HDMI_regdc

Address: Operational Base + offset (0x0370)

Registerdc

Bit	Attr	Reset Value	Description
7:0	RW	0xd0	<p>signal_free_time_l</p> <p>Signal free time length_L</p>

HDMI_regdd

Address: Operational Base + offset (0x0374)

Registerdd

Bit	Attr	Reset Value	Description
7:0	RW	0x20	<p>signal_free_time_h</p> <p>Signal free time length_H</p>

HDMI_Regde

Address: Operational Base + offset (0x0378)

Registerde

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved
3:0	RW	0x0	<p>dev_addr</p> <p>Device logic address</p>

HDMI_rege0

Address: Operational Base + offset (0x0380)

Registere0

Bit	Attr	Reset Value	Description
7:5	RW	0x0	reserved
4	RW	0x1	TMDS_clk_sel 1'b0: select TMDS clock from PLL; 1'b1: select TMDS clock generated by serializer
3	RW	0x0	TMDS_phase_sel 1'b0: select default phase for TMDS clock; 1'b1: select synchronized phase for TMDS clock with regard to TMDS data
2	RW	0x0	hdmi_band_gap_pd HDMI Band gap power down. 1:Active
1	RW	0x0	hdmi_pll_pd HDMI PLL power down. 1:Active
0	RW	0x0	tmds_channel_pd TMDS channel power down. 1:Active

HDMI_rege1

Address: Operational Base + offset (0x0384)

Registere1

Bit	Attr	Reset Value	Description
7:4	RW	0x0	reserved
3	RW	0x1	clk_channel_driver_en Clock channel driver enable 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
2:0	RW	0x7	<p>data_channels_driver_en</p> <p>Three data channels driver enable [2]-ch2 [1]-ch1 [0]-ch0 1'b0: disable 1'b1: enable</p>

HDMI_rege2

Address: Operational Base + offset (0x0388)

Registers2

Bit	Attr	Reset Value	Description
7:4	RW	0xa	<p>clk_driver_strength</p> <p>Clock channel main-driver strength 0000~1111: the strength from low to high</p>
3:0	RW	0xa	<p>data_driver_strength</p> <p>Three data channels main-driver strength 0000~1111: the strength from low to high</p>

HDMI_rege3

Address: Operational Base + offset (0x038c)

Registers3

Bit	Attr	Reset Value	Description
7	RW	0x0	reserved
6:4	RW	0x0	<p>pre_emphasis_strength</p> <p>Pre-emphasis strength 000~111: the strength from 0 to high</p>
3:2	RW	0x3	<p>clk_driver_strength</p> <p>Clock channel pre-driver strength Slew rate from low to high 00~11: the strength from low to high</p>
1:0	RW	0x3	<p>data_driver_strength</p> <p>Three data channels pre-driver strength Slew rate from low to high 00~11: the strength from low to high</p>

HDMI_rege7

Address: Operational Base + offset (0x039c)
 Registere7

Bit	Attr	Reset Value	Description
7:0	RW	0x78	feedback_dividing_ratio_part1 The value of feedback dividing ratio

HDMI_rege8

Address: Operational Base + offset (0x03a0)
 Registere8

Bit	Attr	Reset Value	Description
7:1	RW	0x0	reserved
0	RW	0x0	feedback_dividing_ratio_part2 The value of feedback dividing ratio

HDMI_reged

Address: Operational Base + offset (0x03b4)
 Registered

Bit	Attr	Reset Value	Description
7:5	RW	0x0	reserved
4:0	RW	0x0c	pre_dividing_ratio The value of pre-dividing ratio

Note1: There are some bits of registers have not description in the registers table above, are reserved for other using, please keep it into default value.

24.5 Interface Description

24.5.1 Video Input Source

In RKaudi, the HDMI TX video source comes from VOP.

24.5.2 Audio Input Source

In RKaudi, the HDMI TX audio source comes from I2S_8CH.

24.6 ApplicationNotes

This chapter describes how to bring up HDMI transmitter in your system. As shown few examples below, these introduce the basically HDMI transmitter application, likes, the Hot Plug Detect, EDID read back, multiple audio format input and different video resolution displaying.

You can easily configure these functions with proper registers value setting by HDMI TX APB BUS.

24.6.1 Initial Operation

The default HDMI transmitter is configured to 24bit RGB 1080P resolution video with 8 channel 48K sample I2S format audio input. It is easily for customer to turn on HDMI transmitter without doing more complex operation. Just do the step, reset the HDMI TX.

24.6.2 Hot Plug Detection

Hot Plug Detect is a special feature for HDMI transmitter spying the state on the HDMI port.

You can control this function by using the interrupt signal and proper registers from the HDMI transmitter with few operations. The following is a step by step instruction for detecting the hot plug in and out.

Hot Plug in Steps:

Step1: Send 0 x63 to register 0x00.

Step2: Plug HDMI receiver in.

Step3: Check the interrupt from signal pin_int.

If the pin_int shows high, that means the HDMI transmitter interrupt have generated.

Step4: Check the interrupt.

Read the bit1 o f the register 0 xc8. If bit1=1'b1 that means the hot plug interrupt valid, otherwise there is other interrupt. Then send the 1'b1 to bit1 of register 0xc8to clean up the hot plug interrupt.

Step5: Check if really HDMI receiver plug in.

Read the bit7 of the register 0xc8. If bit7=1'b1 that means hot plug in was really happen, otherwise there is ma y be a glitch o n the HPD port.

Hot Plug out Steps:

Step1: HDMI transmitter at working state.

Step2: Plug HDMI receiver out.

Step3: Check the interrupt from signal pin_int.

If the pin_int shows high, that means the HDMI transmitter interrupt have generated.

Step4: Check the interrupt.

Read the bit1 of the register 0xc8. If bit1=1'b1 that means the hot plug interrupt valid, otherwise there is other interrupt. Then send the 1'b1 to bit1 of register 0xc8to clean up the hot plug interrupt.

Step5: Check if really HDMI receiver plug out.

Read the bit7 of the register 0xc8. If bit7=1'b0 that means hot plug out was really happen, otherwise there is ma y be a glitch o n the HPD port.

24.6.3 Reading EDID

Read EDID is a function that can make the HDMI transmitter to read the HDMI receiver's Extended Display Identification Data (EDID) in order to discover the HDMI receiver's configuration and capabilities. HDMI transmitter can choose the appropriate audio and video format for playing and displaying by the HDMI receiver through the use of the EDID. Besides, HDMI transmitter support the reading Enhanced Extended Display Identification Data (E-EDID) if HDMI receiver have this enhanced structure.

The following describes how to read E-EDID through HDMI transmitter. The total

E-EDID is 512bytes data, which is divided into 2 segments. Each segment has 256bytes data. The Read E-EDID function is only read 128bytes data from HDMI receiver at each time. So, you must read 4 times that can read total 512bytes data back.

Prepare read E-EDID512bytes Steps:

Step1: Send 0 x63 to register 0x00.

System power down mode.

Step2: HDMI transmitter detected Hot Plug in.

Step3: Define DDC clock (SCL) frequency $F_{SCL} = 100K$.

$F_{SCL} = F_{reg_clk}/(4*X)$. $X=\{register\ 0x4c,\ register\ 0x4b\}$.

Note F_{reg_clk} is the frequency of the tmds clock.

Assume the tmds clock is 148.5MHz.

Send 0 x73 to register 0x4b, Send 0x01 to register 0x4c.

Step4: Enable EDID Ready interrupt.

Send 0 x04 to register 0xc0

Read E-EDIDsegment 0x00 256bytes Steps:

Step1: Set EDID FIFO initial address.

Send 0x00 to register 0x4f

Step2: Set EDID first word address, read first 128bytes.

Send 0x00 to register 0x4e

Step3: Set EDID segment address.

Send 0x00 to register 0x4d

Step4: Waiting and check EDID interrupt from pin_int.

Check the EDID Ready state from bit2 of register 0xc1. If bit2 = 1'b1 that means EDID data ready. Then send the 1'b1 to bit2 of register 0xc1 to clean up the EDID Ready interrupt.

Step5: Read first 128bytes EDID data from EDID FIFO.

for ($i = 0, i < 128, i = i + 1$)

 Read data from register 0x50.

Step6: Set EDID FIFO initial address again.

Send 0x00 to register 0x4f

Step7: Set EDID first word address, read last 128bytes.

Send 0x80 to register 0x4e

Step8: Set EDID segment address.

Send 0x00 to register 0x4d

Step9: Waiting and check EDID interrupt from pin_int.

Check the EDID Ready state from bit2 of register 0xc1. If bit2 = 1'b1 that means EDID data ready. Then send the 1'b1 to bit2 of register 0xc1 to clean up the EDID Ready interrupt.

Step10: Read last 128bytes EDID data from EDID FIFO.

 for ($i = 0, i < 128, i = i + 1$)

 Read data from register 0x50.

Read E-EDIDsegment 0x01 256bytes Steps:

Step1: Set EDID FIFO initial address.

Send 0x00 to register 0x4f

Step2: Set EDID first word address, read first 128bytes.

Send 0x00 to register 0x4e

Step3: Set EDID segment address.

Send 0x01 to register 0x4d

Step4: Waiting and check EDID interrupt from pin_int.

Check the EDID Ready state from bit2 of register 0xc1. If bit2 = 1'b1 that means EDID data ready. Then send the 1'b1 to bit2 of register 0xc1 to clean up the EDID Ready interrupt.

Step5: Read first 128bytes EDID data from EDID FIFO.

for (i = 0, i < 128, i = i + 1)

Read data from register 0x50.

Step6: Set EDID FIFO initial address again.

Send 0x00 to register 0x4f

Step7: Set EDID first word address, read last 128bytes.

Send 0x80 to register 0x4e

Step8: Set EDID segment address.

Send 0x01 to register 0x4d

Step9: Waiting and check EDID interrupt from pin_int.

Check the EDID Ready state from bit2 of register 0xc1. If bit2 = 1'b1 that means EDID data ready. Then send the 1'b1 to bit2 of register 0xc1 to clean up the EDID Ready interrupt.

Step10: Read last 128bytes EDID data from EDID FIFO.

for (i = 0, i < 128, i = i + 1)

Read data from register 0x50.

Note: The segment address must be sent at each time when read 128bytes E-EDID data.

24.6.4 Audio input configuration

HDMI transmitteraudio support either SPDIF or four channel I2S input. SPDIF input supports audio sampling rates from 32to 192 KHz. The I2S input supports from 2-channel to 8-channel audio up to 192 KHz. The default audio format is I2S input with 8 channels. The audio sample rate is 48K.

The following describes how to configure audio input format into I2S input with 2 channels and the sample rate is 44.1K.

Audio input requirement:

SD0 input (2 Channels I2S input)

WS (fs) = 44.1 KHz

SCK = 64fs

MCLK = 256fs

Configure Audio Input Format Steps:

Step1: Select I2S input.

Send 0x01 to register 0x35

Step2: Select SD0 input.

Send 0x04 to register 0x38

Step3: Select N/CTS for sample rate = 44.1 KHz.

Send 0x18 to register 0x40

Send 0x80 to register 0x41

The following describes how to configure audio input format into I2S input with 8 channels and the sample rate is 44.1K.

Audio input requirement:

SD[3 :0] input (8 Channels I2S input)

WS (fs) = 44.1 KHz

SCK = 64fs

MCLK = 256fs

Configure Audio Input Format Steps :

Step1: Select I2S input.
Send 0x01 to register 0x35
Step2: Select SD[3:0] input.
Send 0x3c to register 0x38
Send 0x00 to register 0x39
Step3: Select N/CTS for sample rate = 44.1 KHz.
Send 0x18 to register 0x40
Send 0x80 to register 0x41

24.6.5 Video input configuration

HDMI transmitter support RGB/YCbCr 24/30/36bit video input with different resolution. The default video format is RGB24bit input at resolution of 1080P@60. The following describes how to configure video input format into RGB24bit input at resolution of 480P@60, 720P@60 or 1080P@60.

VOP dclk cannot get invert.

Video input requirement:

24bit RGB 4:4:4 Source.
Resolution is 480P@60, 720P@60 or 1080P@60.

Configure Video Input Format Steps:

Step1: Select configure for video format.
Send 0x06 to register 0x9f
Step2: Configure the AVI info to HDMI RX.
Send 0x82 to register 0xa0 //HB0
Send 0x02 to register 0xa1 //HB1
Send 0x0d to register 0xa2 //HB3
Send 0x00 to register 0xa3 //PB0: checksum is auto set, needn't set this register
Send 0x00 to register 0xa4 //PB1
Send 0x08 to register 0xa5 //PB2
Send 0x70 to register 0xa6 //PB3
Send 0x10 to register 0xa7 //PB4: VID 1920*1080P@60
Send 0x40 to register 0xa8 //PB5
Note: Select correct VID for displaying.
Send 0x02 to register 0xa7 for 720*480P@60.
Send 0x04 to register 0xa7 for 1280*720P@60.
Send 0x05 to register 0xa7 for 1920*1080I@60.
Send 0x10 to register 0xa7 for 1920*1080P@60.
Send 0x11 to register 0xa7 for 720*576P@50.
Send 0x13 to register 0xa7 for 1280*720P@50.
Send 0x14 to register 0xa7 for 1920*1080I@50.
Send 0x1f to register 0xa7 for 1920*1080P@50.

The detail configuration for AVI information, please refer to the HDMI specification (8.2.1) and CEA-861-D (6.3).

24.6.6 Low Power Mode

HDMI Transmitter can be configured into Low Power Mode at special customer works mode. The following is a step by step instruction to describe how to configure our HDMI Transmitter into this mode.

Low Power Enter Steps:

- Step1:**HDMI Transmitter working.
Step2:Low Power analog module.
 Send 0x00 to register 0xe1.//Drive disable
 Send 0x17 to register 0xe0.
Step3:Assign pin_vclk = 1'b0 or 1'b1.

Low Power Quit Steps:

- Step1: Reset system by pin_rst_n.
 Step2: Wait Hot Plug.
 Step3: Read EDID.
 Step4: Active vclk through p_in_vclk
 Step5: Bring up analog module.
 Send 0x15 to register 0xe0. //PLL power on
 Send 0x14 to register 0xe0. //TX power on
 Send 0x10 to register 0xe0. //BG power on
 Send 0x0f to register 0xe1. //driver enable
 Step6: Configuration mode reg at addr 0x00 if needed.
 Step7: Configuration Video format if needed.
 Step8: Configuration Audio format if needed.
 Step9: Configuration mode reg, power on digital part and select tmds_clk for configuration.
 Send 0x61 to register 0x00.
 Step10: Synchronize analog module.
 Send 0x00 to register 0xce.
 Send 0x01 to register 0xce.

24.6.7 CEC OPERATION

The CEC line is used for high-level user control of HDMI-connected devices.

You can control this function by using the interrupt signal and proper registers from the HDMI transmitter with few operations.

The following is a step by step instruction for CEC TX operations and CEC RXoperations.

CEC TX steps

- (Send the Command : Standby (0x36) for example)
- Step1: Set logic address for CEC (logic address = 1).
 Send 0x01 to register 0xde.
- Step2: Configure the divide numbers for internal reference clock.
 Send 0x03 to register 0xd4 and 0x09 to register 0xd5. See Note1 for details.
- Step3: Configure the value for signal free time counter.
 Send 0xd0 to register 0xdc and 0x20 to register 0xdd. See Note2 for details.
- Step4: Configure point address for write CEC TX data.
 Send 0x00 to register 0xd2.
- Step5: Configure TX data FIFO (the Command o f Standby).
 Send 0x01 to register 0xd1, then, send 0x36 to register 0xd1.
 The data 0x01 is for the Header Block, in which the high 4bits is for the initiator, the low 4bits is for the destination. The data 0x36 is for the Data Block, it indicate the command for standby.
- Step6: Configure TX data length of this packet. The length is 2.

Send 0x02 to register 0xd6.

Step7: Enable count CEC bus free time.

Send 0x04 to register 0xd0.

Step8: Waiting 16.8ms+4 us for CEC bus free.

If Interrupt happened for CEC line free time not satisfied. You should change to receive coming data from opposite. After this you can do CEC TX steps again.

(Note that the time for 16.8ms+4us should be compute by MCU itself)

Step9: Begin TX.

Send 0x05 to register 0xd0.

Step10: Waiting and check interrupt of CEC TX done.

Waiting the interrupt, then, read the value of register 0xda, if the value is 0x08 that means CEC command was transmitted successfully. At last, send 0x08 to register 0xda to clear up this interrupt.

CEC RX steps:

Step1: Set logic address for CEC (logic address = 1).

Send 0x01 to register 0xde.

Step2: Configure the divide numbers for internal reference clock. See Note1 for details.

Send 0x03 to register 0xd4 and 0x09 to register 0xd5.

Step3: Configure the point address for CEC RX data.

Send 0x00 to register 0xd3.

Step4: Waiting CEC TX send CEC packet to our HDMI Transmitter, then check interrupt of CEC. Read the value of register 0xdb, if the value is 0x80, that means CEC command was received successfully. Then send 0x80 to register 0xdb to clear up this interrupt.

Step5: Read the RX packet length.

Read the value of the register 0xd7.

Step6: Read the received CEC packet.

Read the value of the register 0xd1 according to the RX packet length. If the length is 2, please read twice from the register 0xd1.

Note1: The system clock (Default value Fsys = 20MHz from pin_sys_clk), register 0xd4 and 0xd5 are used to configure the CEC logic to generate a reference clock (Fref=0.5 MHz, Tref = 2us), which is used to control the CEC signal's generation. The following is the formula for generating reference clock 0.5MHz.

$Fref = Fsys / ((register 0xd4 + 1)*(register 0xd5 + 1))$. Under the default 20MHz system clock, the values of register 0xd4 and 0xd5 are 0x03 and 0x09.

Note2: Before attempting to transmit or re-transmit a frame, a device shall ensure that the CEC line has been inactive for a number of bit periods. For example, the signal free time is $7*2.4ms=16.8ms$ (the bit time is 2.4 ms). According to the $16.8ms/2us = \{register 0xdd, register 0xdc\}$, so the register 0xdd=0x20 and register 0xdc=0xd0.

24.6.8 HDCP OPERATION

HDCP is designed to protect the transmission of Audiovisual Content between an HDCP Transmitter and an HDCP Receiver. You can control this function by using the interrupt signal and proper registers from the HDMI transmitter with few operations.

The following is a step by step instruction for HDCP operation.

HDCP operation steps (skip BKSV black list check) :

Example Condition: Video Format : 1920x1080p@59.94/60Hz

Step1: Select the TMDS clock to configure registers.

Send 0x61 to register 0x00.

Step2: Write HDCP transmitter's Akey to the chip. **See Note1 for details.**

```
for(i =0; i <= 39 ; i = i + 1 )  
begin  
    hdcp_wdata_temp = akey[i];  
    for ( j=0 ; j < 7 ; j = j + 1 )  
        begin  
            Send hdcp_wdata_temp[7:0] to register 0x98;  
            hdcp_wdata_temp = hdcp_wdata_temp >>8;  
        end  
    end
```

Step3: Write HDCP transmitter's AKSV1 to the chip. **See Note1 for details.**

```
hdcp_wdata_temp = pre_ksv1  
for (i = 0, i<5, i = i + 1)  
begin  
    Send hdcp_wdata_temp[7:0] to register 0x98;  
    hdcp_wdata_temp = hdcp_wdata_temp >>8;  
end
```

Step4: Write HDCP transmitter's AKSV2 to the chip. **See Note1 for details.**

```
hdcp_wdata_temp = pre_ksv2  
for (i = 0, i<5, i = i + 1)  
begin  
    Send hdcp_wdata_temp[7:0] to register 0x98;  
    hdcp_wdata_temp = hdcp_wdata_temp >>8;  
end
```

Step5: Check the key_ready signal.

Read the value of register 0x54, if the value is 0x01, that means the Akey and AKSV were load into the chip successfully.

Step6: Configure the DDC frequency. **See Note2 for details.**

Send 0x73 to register 0x4b.
Send 0x01 to register 0x4c.

Step7: Configure the HDCP function.

Send 0x77 to register 0x53.

Step8: Open the mask for HDCP interrupt.

Send 0xff to register 0xc2.
Send 0xff to register 0xc4.

Step9: Start the HDCP authentication.

Send 0x96 to register 0x52.

Step10: Checking HDCP interrupt.

Read the value of register 0xc3 and register 0xc5, if the value of register 0xc3 is 0x10 and the value of register 0xc5 is 0x65, that means the HDCP authentication was successfully finished.

HDCP operation steps (**not skip BKSV black list check**) :

Example Condition: Video Format : 1920x1080p@59.94/60Hz

Step1: Select the TMDS clock to configure registers.

Send 0x01 to register 0x00.

Step2: Write HDCP transmitter's Akey to the chip. **See Note1 for details.**

```
for(i =0; i <= 39 ; i = i + 1 )  
begin  
    hdcp_wdata_temp = akey[i];  
    for ( j=0 ; j < 7 ; j = j + 1 )  
        begin  
            Send hdcp_wdata_temp[7:0] to register 0x98;
```

```
    hdcp_wdata_temp = hdcp_wdata_temp >>8;  
    end  
end
```

Step3: Write HDCP transmitter's AKSV1 to the chip. **See Note1 for details.**

```
hdcp_wdata_temp = pre_ksv1  
for (i = 0, i<5, i = i + 1)  
begin  
    Send hdcp_wdata_temp[7:0] to register 0x98;  
    hdcp_wdata_temp = hdcp_wdata_temp >>8;  
end
```

Step4: Write HDCP transmitter's AKSV2 to the chip. **See Note1 for details.**

```
hdcp_wdata_temp = pre_ksv2  
for (i = 0, i<5, i = i + 1)  
begin  
    Send hdcp_wdata_temp[7:0] to register 0x98;  
    hdcp_wdata_temp = hdcp_wdata_temp >>8;  
end
```

Step5: Check the key_ready signal.

Read the value of register 0x54, if the value is 0x01, that means the Akey and AKSV were load into the chip successfully.

Step6: Configure the DDC frequency. **See Note2 for details.**

Send 0x73 to register 0x4b.
Send 0x01 to register 0x4c.

Step7: Configure the HDCP function.

Send 0x37 to register 0x53(do not skip the BKSV blacklist check).

Step8: Open the mask for HDCP interrupt.

Send 0xff to register 0xc2.
Send 0xff to register 0xc4.

Step9: Start the HDCP authentication.

Send 0x96 to register 0x52.

Step10: Check the BKSV.

Wait for INT from registerc3 bit5(bksv_update), than read register66~register6a for BKSV values to check whether the BKSV is in the revocation list. If the BKSV is not in the revocation list, write 1 to register52 bit6(bksv_pass) , and authentication will continue. Else if the BKSV is in the revocation list, write 1 to register52 bit5(bksv_fail), and authentication will stop.

Step11: Checking HDCP interrupt.

Read the value of register 0xc3 and register 0xc5, if the value of register 0xc3 is 0x10 and the value of register 0xc5 is 0x65, that is means the HDCP authentication was successfully finished.

Note1: For HDCP's akey and KSV write, use the TEST KEYS in HDCP specification for example

```
akey[0]=56'h4da4588f131e69;  
akey[1]=56'h1f823558e65009;  
akey[2]=56'h8a6a47abb9980d;  
akey[3]=56'hf3181b52cbc5ca;  
akey[4]=56'fb147f6896d8b4;  
akey[5]=56'he08bc978488f81;  
akey[6]=56'ha0d064c8112c41;  
akey[7]=56'hb39d5a28242044;  
akey[8]=56'hb928b2bdad566b;  
akey[9]=56'h91a47b4a6ce4f6;
```

```
akey[10] =56'h5600f8205e9d58;  
akey[11] =56'h8c7fb706ee3fa0;  
akey[12] =56'hc02d8c9d7cbc28;  
akey[13] =56'h561261e54b9f05;  
akey[14] =56'h74f0de8ccac1cb;  
akey[15] =56'h3bb8f60efcdb6a;  
akey[16] =56'ha02bbb16b22fd7;  
akey[17] =56'h482f8e46785498;  
akey[18] =56'h66ae2562274738;  
akey[19] =56'h3d4952a323ddf2;  
akey[20] =56'he2d231767b3a54;  
akey[21] =56'h4d581aede66125;  
akey[22] =56'h326082bf7b22f7;  
akey[23] =56'hf61b463530ce6b;  
akey[24] =56'h360409f0d7976b;  
akey[25] =56'ha1e105618d49f9;  
akey[26] =56'hc98e9dd1053406;  
akey[27] =56'h20c36794426190;  
akey[28] =56'h964451ceac4fc3;  
akey[29] =56'h3e904504e18c8a;  
akey[30] =56'h290010579c2dfc;  
akey[31] =56'hd7943b69e5b180;  
akey[32] =56'h54c7ea5bdd7b43;  
akey[33] =56'h74fb5887c790ba;  
akey[34] =56'h935cfa364e1de0;  
akey[35] =56'h03075e159a11ae;  
akey[36] =56'h05d3408a78fb01;  
akey[37] =56'h0059a5d7a04db3;  
akey[38] =56'h373b634a2c9e40;  
akey[39] =56'h2573bbb4562041;  
  
pre_ksv1 = 40'hb70361f714;  
pre_ksv2 = 40'hb70361f714;
```

Note2: For DDC frequency configuration

In the HDCP specification, the frequency value of DDC(F_{ddc}) support only up to 100Kbps, and now we select the TMDS clock as the reference clock of DDC , and under the condition of 1920x1080p@59.94/60Hz, the frequency of TMDS(F_{TMDS}) is 148.5MHz. Pay attention, internally we use the 4 times of DDC frequency to generate the SCL. So

$X = F_{TMDS}/F_{ddc}/4 = 'h173$, So the values of register 0x4b and 0x4c are 0x73 and 0x01.

24.6.9 NOMAL OPERATION EXAMPLE AT 1080P

After the description of HDMI Transmitter configuration above, the following example shows an entire configuration for HDMI Transmitter works on the 1080P mode.

Audio input requirement:

SD0 input (2 Channels I2S input)

WS (fs) = 48 KHz

SCK = 64fs

MCLK = 256fs

Video input requirement:

24bit RGB 4:4:4 Source.

Resolution is 1080P@60.

Setup Steps:

Step1:Power on HDMI Transmitter.

Step2: Configure the input signals.

Assign pin_test_en = 1'b0.

Step3:Reset HDMI Transmitter.

Assign 0 to the signal pin_rst_n and then assign 1.

Step4:System power down.

Send 0x63 to register 0x00.

Step5:Detect Hot Plug In.

Step6:Read EDID.

Step7: Configure video input format.

Step8:Configure audio input format.

Step9:Assign video and audio source to HDMI Transmitter.

Step10:System power on.

Send 0x61 to register 0x00.

Step11: Now, HDMI Transmitter is ready to go. Start your operation.

24.7 Electrical Specification

HDMI Transmitter contains tunable source termination and pre-emphasis to enable high speed operation. The Transmitter meets the AC specifications below across all operating conditions specified. Rise and fall times are defined as the signal transition time between 20% and 80% of the nominal swing voltage (V_{swing}) of the device under test. The Transmitter intra-pair skew is the maximum allowable time difference (on both low-to-high and high-to-low transitions) as measured at TP1, between the true and complement signals of a given differential pair. This time difference is measured at the midpoint on the single-ended signal swing of the true and complement signals. The Transmitter inter-pair skew is the maximum allowable time difference (on both low-to-high and high-to-low transitions) as measured at TP1.

Table 24-5Normal Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
V_{CC}	Supply Voltage	3.0	3.3	3.6	V
V_{REF}	Reference Voltage	3.0	3.3	3.6	V
V_{CCN}	Supply Voltage Noise			100	mV _{p. p}
T_A	Ambient Temperature (with power applied)	0	25	70	°C

Table 24-6Absolute Maximum Conditions

Symbol	Parameter	Min	Typ	Max	Units
$V_{CC}^{1.2}$	Supply Voltage 3.3V	-0.3		4.0	V
$V_1^{1.2}$	Input Voltage	-0.3		$V_{CC}+0.3$	V
$V_0^{1.2}$	Output Voltage	-0.3		$V_{CC}+0.3$	V
$V_J^{1.2}$	Junction Temperature (with power applied)			125	°C

Table 24-7Transmitter DC Characteristics at TP1

Item	Value
Single-ended high level output voltage, V_H	$AV_{CC}-10mV < V_H < AV_{CC}+10mV$ when

	sink <= 165Mhz AV _{cc} -200mV < V _H < AV _{cc} +10mV when sink >165Mhz
Single-ended low level output voltage, V _L	(AV _{cc} - 600mV) < V _L < (AV _{cc} - 400mV) when sink <=165Mhz
	(AV _{cc} - 700mV) < V _L < (AV _{cc} - 400mV) when sink > 165Mhz
Single-ended output swing voltage, V _{swing}	400mV < V _{swing} < 600mV
Single-ended standby (off) output voltage, V _{OFF}	AV _{cc} +10mV
Single-ended standby (off) output current, I _{OFF}	I _{OFF} < 10uA

Table 24-8 Transmitter AC Characteristics at TP1

Item	Value
Risetime/falltime (20%-80%)	75psec < Risetime/falltime < 0.4 T _{bit}
Overshoot, max	15% of full differential amplitude (V _{swing} *2)
Undershoot, max	25% of full differential amplitudee (V _{swing} *2)
Intra-Pair Skew at Transmitter Connector, max	0.15 T _{bit}
Inter-Pair Skew at Transmitter Connector, max	0.20 T _{pixel}
TMDS Differential Clock Jitter, max	0.25 T _{bit}
Clock duty cycle	40% to 60%

Table 24-9 Programmable Output Resistance and Output Equalization Level

Item	Value
Output Equalization level	10% to 60%
Termination Resistance	50ohm and 75 ohms selectable with fine steps
Output Swing Ranges	4 different levels

24.7.1 CONTROL SIGNAL - DDC

DDC (Display Data Channel) control signals follow the I2C Bus specifications. More details to be filled out from I2C specs.

Item	Value
High Voltage Level (Vih)	-3.0V < Vih < 3.8V
Low Voltage Level (Vil)	-0.5V < Vil < 1.5V
SCL clock frequency	100KHz (max)
Rise time	< 1000ns
Fall time	< 300ns
Capacitive load on bus line	400pF

24.7.2 CONTROL SIGNAL - HPD

HPD (Hot Plug Detect) signal is used by the source to read the sink's E-DID. The sink needs to meet the following requirements for reliable detection.

For Sink

Item	Value

High Voltage Level (Vih)	$2.4V < Vih < 5.3V$
Low Voltage Level (Vil)	$0V < Vil < 0.4V$

For Source

Item	Value
High Voltage Level (Voh)	$2.0V < Voh < 5.3V$
Low Voltage Level (Vol)	$0V < Vol < 0.8V$

Rockchip Confidential

Chapter 25 USB Host 2.0

25.1 Overview

USB HOST2.0 supports Non_OTG Host functions and is fully compliant with USB2.0 specification, and support high-speed (480Mbps), full-speed (12Mbps), low-speed (1.5Mbps) transfer. It is optimized for point-to-point applications (no hub, direct connection to device).

25.1.1 Features

- ◆ Compliant with the USB2.0 Specification
- ◆ Operates in Non_OTG Host mode
- ◆ Operates in High-Speed, Full-Speed, Low-speed mode
- ◆ Support 16 channels in host mode
- ◆ Built-in one 840x35 bits FIFO
- ◆ Internal DMA with no scatter/gather function

25.2 Block Diagram

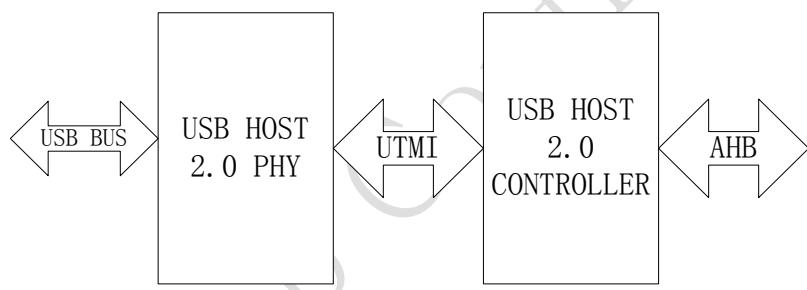


Fig. 25-1 USB HOST 2.0 Architecture

Fig.25-1 shows the architecture of USB HOST 2.0. It is broken up into two separate units: USB HOST 2.0 controller and USB HOST 2.0 PHY. The two units are interconnected with 16-bits UTMI interface.

25.3 USB Host2.0 Controller

Much the same as USB OTG with no Device Mode supported. See Chapter OTG for more information.

Note: There is another option for host controller: EHCI-OHCI Controller. Please refer to <DesignWare Cores USB 2.0 Host-AHB Controller > for details

25.4 USB Host2.0 PHY

Much the same as USB OTG with no Device Mode supported. See Chapter OTG for more information.

USB Host2.0 PHY doesn't support UART-DEBUG function.

25.5 Register Description

The Registers are much the same as OTG with no Device-Mode supported. The

registers of Device are not available. See Chapter OTG for more information.

25.6 Interface description

Table 25-1 USB HOST 2.0 Interface Description

Module Pin	Direction	Pad Name	pinmux
VSSA	AG	VSSA	-
VCCA3P3	AP	VCCA3P3	-
VCCCORE1P1	AP	VCCCORE1P1	-
USB1PN	A	USB0PN	-
USBRBIAS	A	USBRBIAS	-
USB1PP	A	USB0PP	-

Note: **A**—Analog pad ; **AP**—Analog power; **AG**—Analog ground ;**DP**—Digital power ;**DG**—Digital ground;

25.7 Application Note

See Chapter OTG for more information.

25.7.1 Reset a port

CRU_SOFRST4_CON contains HOST reset signal description. Please refer to "Chapter CRU" for more details.

25.7.2 Relative GRF Registers

GRF_UOC0_CON0 ~ GRF_UOC0_CON2 is OTG PHY register.

GRF_UOC0_CON3 is OTG Controller register.

Please refer to "Chapter GRF" for more details.

Chapter 26 USB OTG 2.0

26.1 Overview

USB OTG 2.0 is a Dual-Role Device controller, which supports both device and host functions and is fully compliant with OTG Supplement to USB2.0 specification, and support high-speed (480Mbps), full-speed (12Mbps), low-speed (1.5Mbps) transfer.

USB OTG 2.0 is optimized for portable electronic devices, point-to-point applications (no hub, direct connection to device) and multi-point applications to devices. USB OTG 2.0 interface supports both device and host functions and is fully compliant with OTG Supplement to USB2.0 specification, and support high-speed (480Mbps), full-speed (12Mbps), low-speed (1.5Mbps) transfer. It is optimized for portable electronic device, point-to-point applications (no hub, direct connection to device) and multi-point applications to devices.

26.1.1 Features

- ◆ Compliant with the OTG Supplement to the USB2.0 Specification
- ◆ Operates in High-Speed and Full-Speed mode
- ◆ Support 9 channels in host mode
- ◆ 9 Device mode endpoints in addition to control endpoint 0, 4 in, 3 out and 2 IN/OUT
- ◆ Built-in one 1024x35 bits FIFO
- ◆ Internal DMA with scatter/gather function
- ◆ Supports packet-based, dynamic FIFO memory allocation for endpoints for flexible, efficient use of RAM
- ◆ Support dynamic FIFO sizing

26.2 Block Diagram

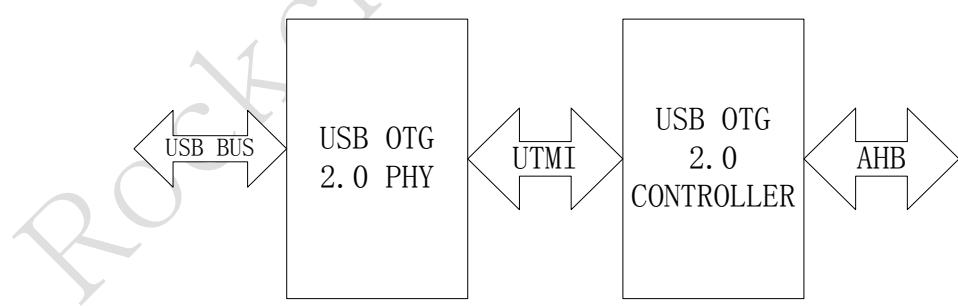


Fig. 26-1 USB OTG 2.0 Architecture

Fig.26-1 shows the architecture of USB OTG 2.0. It is broken up into two separate units: USB OTG 2.0 controller and USB OTG 2.0 PHY. The two units are interconnected with UTMI interface.

26.2.1 USB OTG 2.0 Controller Function

The USB OTG 2.0 Controller controls SIE (Serial Interface Engine) logic, the endpoint logic, the channel logic and the internal DMA logic.

The SIE logic contains the USB PID and address recognition logic, and other sequencing and state machine logic to handle USB packets and transactions.

Generally the SIE Logic is required for any USB implementation while the number and types of endpoints will vary as function of application and performance requirements.

The endpoint logic contains the endpoint specific logic: endpoint number recognition, FIFOs and FIFO control, etc.

The channel Logic contains the channel tasks schedule, FIFOs and FIFO control, etc.

The internal DMA logic controls data transaction between system memory and USB FIFOs.

26.2.2 USB OTG 2.0 PHY Function

The USB OTG 2.0 PHY handles the low level USB protocol and signaling. This includes features such as; data serialization and deserialization, bit stuffing and clock recovery and synchronization. The primary focus of this block is to shift the clock domain of the data from the USB 2.0 rate to the frequency of UTMI clock which is 30MHz.

26.2.3 UTMI Interface

- Transmit

Transmit must be asserted to enable any transmissions.

The USB OTG2.0 CONTROLLER asserts TXValid to begin a transmission and negates TXValid to end a transmission. After the USB OTG2.0 CONTROLLER asserts TXValid it can assume that the transmission has started when it detects TXReady asserted.

The USB OTG2.0 CONTROLLER assumes that the USB OTG2.0 PHY has consumed a data byte if TXReady and TXValid are asserted.

The USB OTG2.0 CONTROLLER must have valid packet information (PID) asserted on the Data In bus coincident with the assertion of TXValid. Depending on the USB OTG2.0 PHY implementation, TXReady may be asserted by the Transmit State Machine as soon as one CLK after the assertion of TXValid. TXValid and TXReady are sampled on the rising edge of CLK.

The Transmit State Machine does NOT automatically generate Packet ID's (PIDs) or CRC. When transmitting, the USB OTG2.0 CONTROLLER is always expected to present a PID as the first byte of the data stream and if appropriate, CRC as the last bytes of the data stream.

The USB OTG2.0 CONTROLLER must use LineState to verify a Bus Idle condition before asserting TXValid in the TX Wait state.

The state of TXReady in the TX Wait and Send SYNC states is undefined. An MTU implementation may prepare for the next transmission immediately after the Send EOP state and assert TXReady in the TX Wait state. An MTU implementation may also assert TXReady in the Send SYNC state. The first assertion of TXReady is Macrocell implementation dependent. The USB OTG2.0 CONTROLLER must prepare DataIn for the first byte to be transmitted before asserting TXValid.

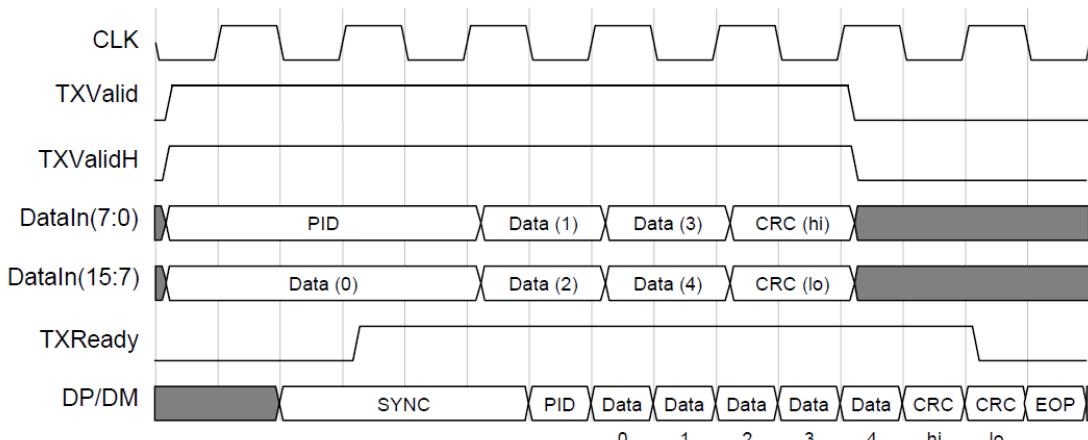


Fig. 26-2 UTMI interface – Transmit timing for a data packet

- Receive

RXActive and RXValid are sampled on the rising edge of CLK.

In the RX Wait state the receiver is always looking for SYNC.

The USB OTG 2.0 PHY asserts RXActive when SYNC is detected (Strip SYNC state).

The USB OTG 2.0 PHY negates RXActive when an EOP is detected (Strip EOP state).

When RxActive is asserted, RXValid will be asserted if the RX Holding Register is full.

RXValid will be negated if the RX Holding Register was not loaded during the previous byte time.

This will occur if 8 stuffed bits have been accumulated.

The USB OTG2.0 Controller must be ready to consume a data byte if RXActive and RXValid are asserted (RX Data state).

In FS mode, if a bit stuff error is detected then the Receive State Machine will negate RXActive and RXValid, and return to the RX Wait state.

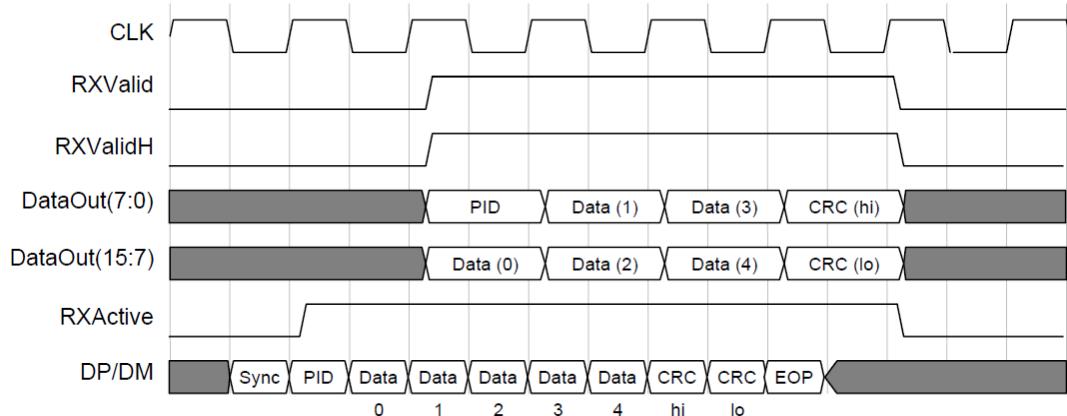


Fig. 26-3 UTMI interface – Receive timing for a data packet

26.3 USB OTG2.0 Controller

Fig.26-4 shows the main components and flow of the USB OTG 2.0 controller

system.

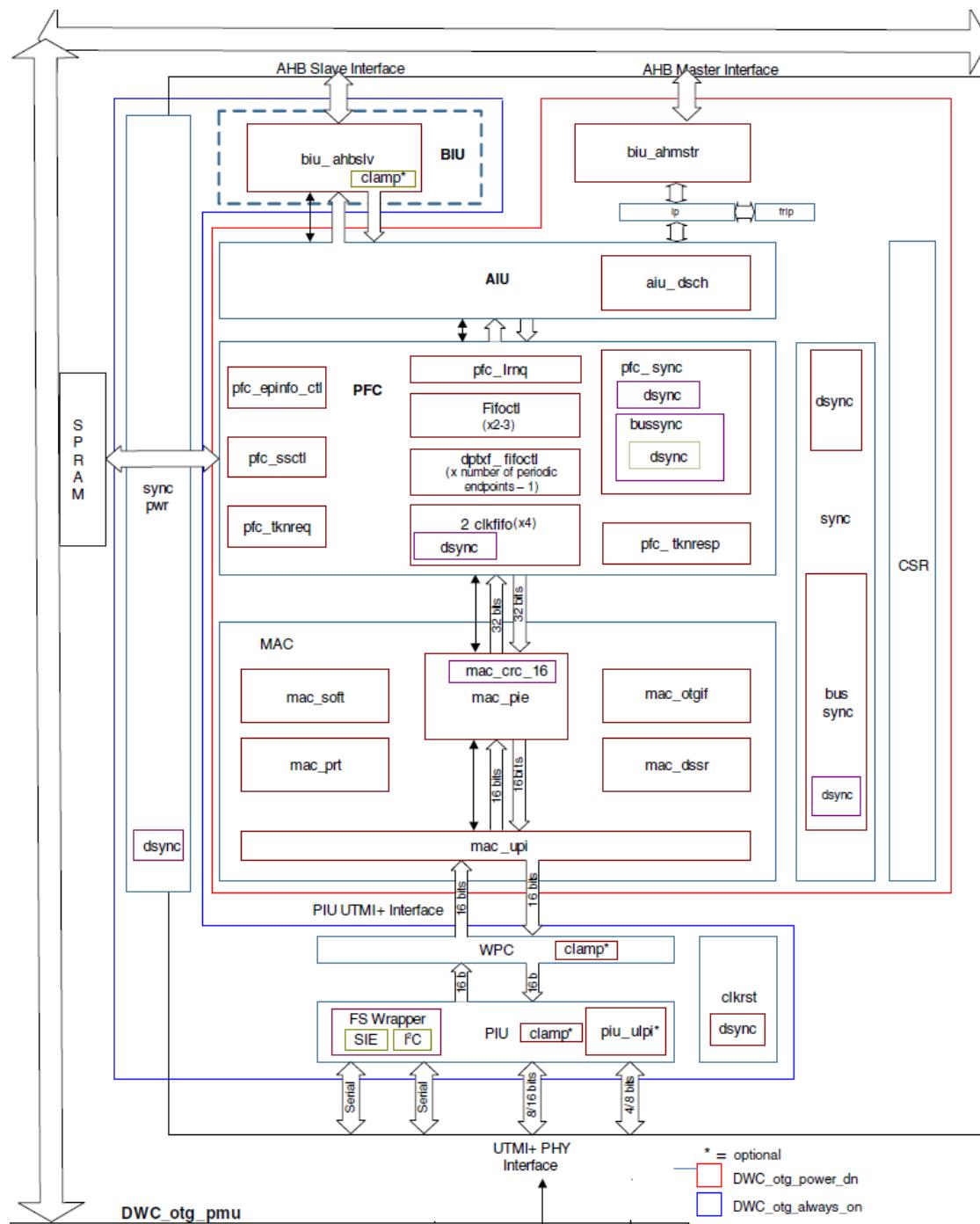


Fig. 26-4 USB OTG2.0 Controller Architecture

1). AHB Slave Bus Interface Unit (BIUS)

The AHB Slave interface unit converts AHB cycles to CSR write/read, Data-FIFO read/write, and DFIFO push/pop signals.

2) Control and Status Registers (CSR)

The CSR block resides in the AHB clock domain, and contains all registers except the Power and Clock Gating Control Register (PCGCCTL) and bits 31:29 of the Core Interrupt register (GINTSTS).

3) Application Interface Unit (AIU)

The application Interface Unit (AIU) consists of the following interfaces:

- AHB Master
- AHB Slave
- Packet FIFO Controller
- Control and Status registers

4). DMA Scheduler (DSCH)

This block is used only in DMA mode. It controls the transfer of data packets between the system memory and the USB OTG 2.0 Controller for both Internal and External DMA.

5). Packet FIFO Controller (PFC)

Several FIFOs are used in Device and Host modes to store data inside the core before transmitting it on either the AHB or the USB. PFC connect the Data FIFO interface to an industry-standard, single-port synchronous SRAM. Address, write data, and control outputs are driven late by the USB OTG 2.0 Controller, but in time to meet the SRAM setup requirements. Input read data is expected late from the SRAM and registered inside the core before being used.

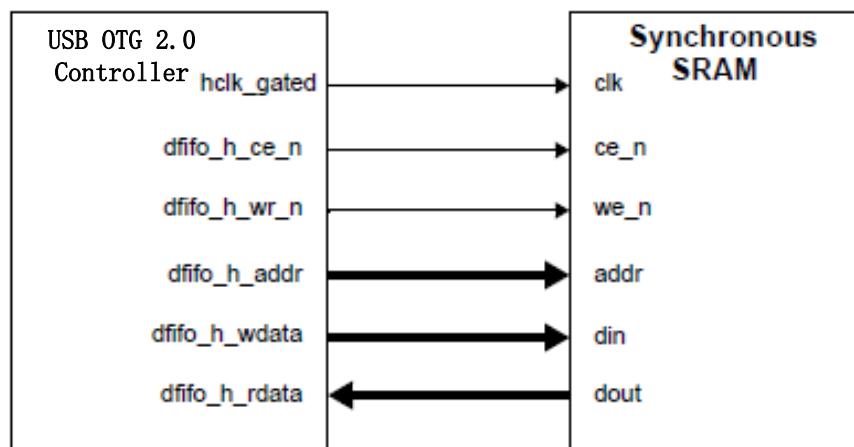


Fig. 26-5 DFIFO single-port synchronous SRAM interface

6). Media Access Controller (MAC)

The Media Access Controller (MAC) module handles USB transactions, and device, host, and OTG protocols.

7) PHY Interface Unit (PIU)

The core uses 16-bit UTMI+ Interface.

8) Wakeup and Power Controller (WPC)

When the USB is suspended or the session is not valid, the PHY is driven into Suspend mode and the PHY clock is stopped to reduce PHY and the core power consumption. To reduce power consumption further, the core also supports AHB clock gating and partial power-down.

26.3.1 Host Architecture

The host uses one transmit FIFO for all non-periodic OUT transactions and one transmit FIFO for all periodic OUT transactions. These transmit FIFOs are used as transmit buffers to hold the data (payload of the transmit packet) to be transmitted over USB.

The host pipes the USB transactions through Request queues (one for periodic and one for non-periodic). Each entry in the Request - queue holds the IN or OUT channel number along with other information to perform a transaction on the USB. The order in which the requests are written into the queue determines the sequence of transactions on the USB. The host processes the periodic Request queue first, followed by the non-periodic Request queue, at the beginning of each (micro) frame.

The host uses one Receive-FIFO for all periodic and non-periodic transactions. The FIFO is used as a Receive-buffer to hold the received data (payload of the received packet) from the USB until it is transferred to the system memory. The status of each packet received also goes into the FIFO. The status entry holds the IN channel number along with other information, such as received byte count and validity status, to perform a transaction on the AHB.

26.3.2 Device Architecture

The core uses Dedicated Transmit FIFO Operation. In this mode, there are individual transmit FIFOs for each IN endpoint.

The OTG device uses a single receive FIFO to receive the data for all the OUT endpoints. The receive FIFO holds the status of the received data packet, such as byte count, data PID and the validity of the received data. The DMA or the application reads the data out of the receive FIFO as it is received.

26.3.3 FIFO Mapping

- Fig.26-6 shows FIFO mapping in Host mode.

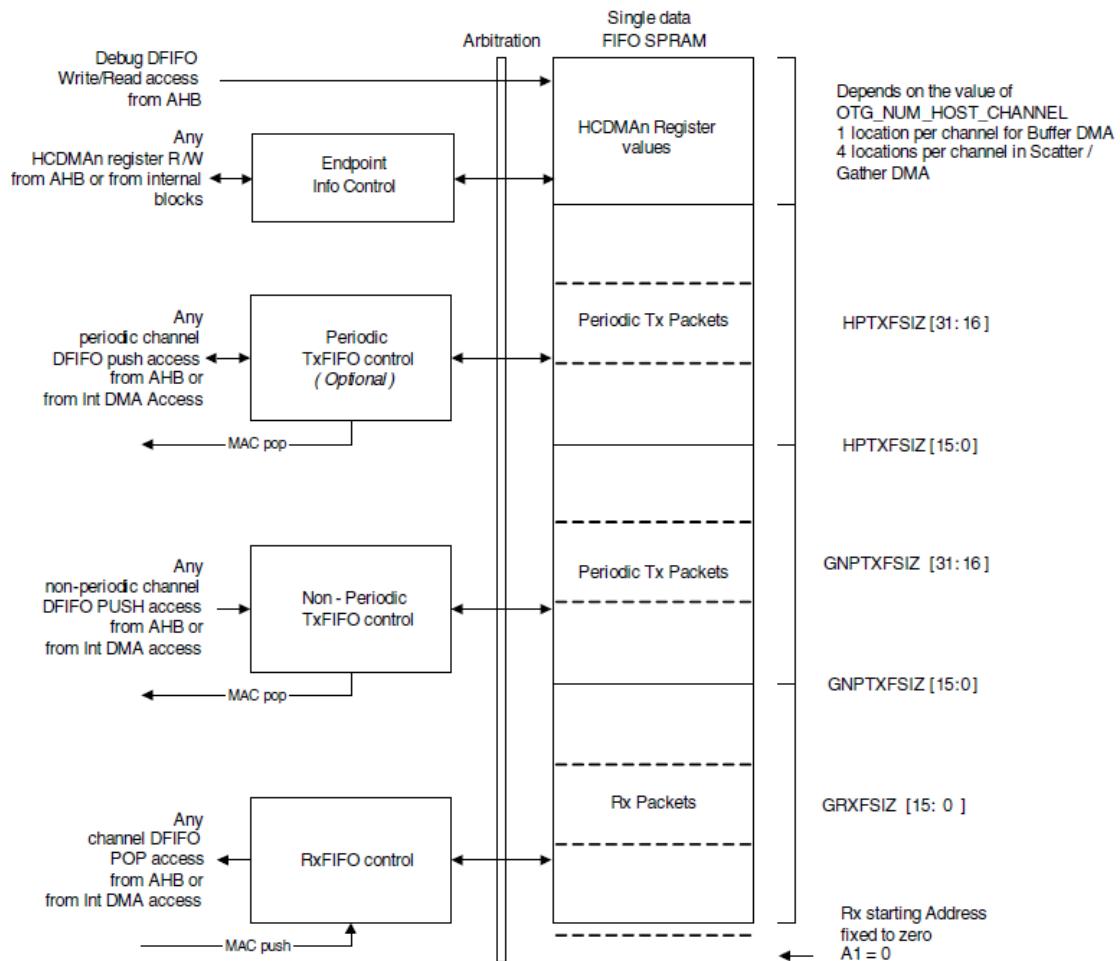


Fig. 26-6 USB OTG 2.0 Controller host mode FIFO address mapping

Note: When the device is operating in Internal DMA mode, the last locations of the SPRAM are used to store the DMAADDR values for each channel.

- Fig.26-7 shows FIFO mapping in Device mode.

When the device is operating in non-Descriptor Internal DMA mode, the last locations of the SPRAM are used to store the DMAADDR values for each channel. When the device is operating in Descriptor mode, then the last locations of the SPRAM store the Base Descriptor address, Current Descriptor address, Current Buffer address, and status quad let information for each endpoint direction.

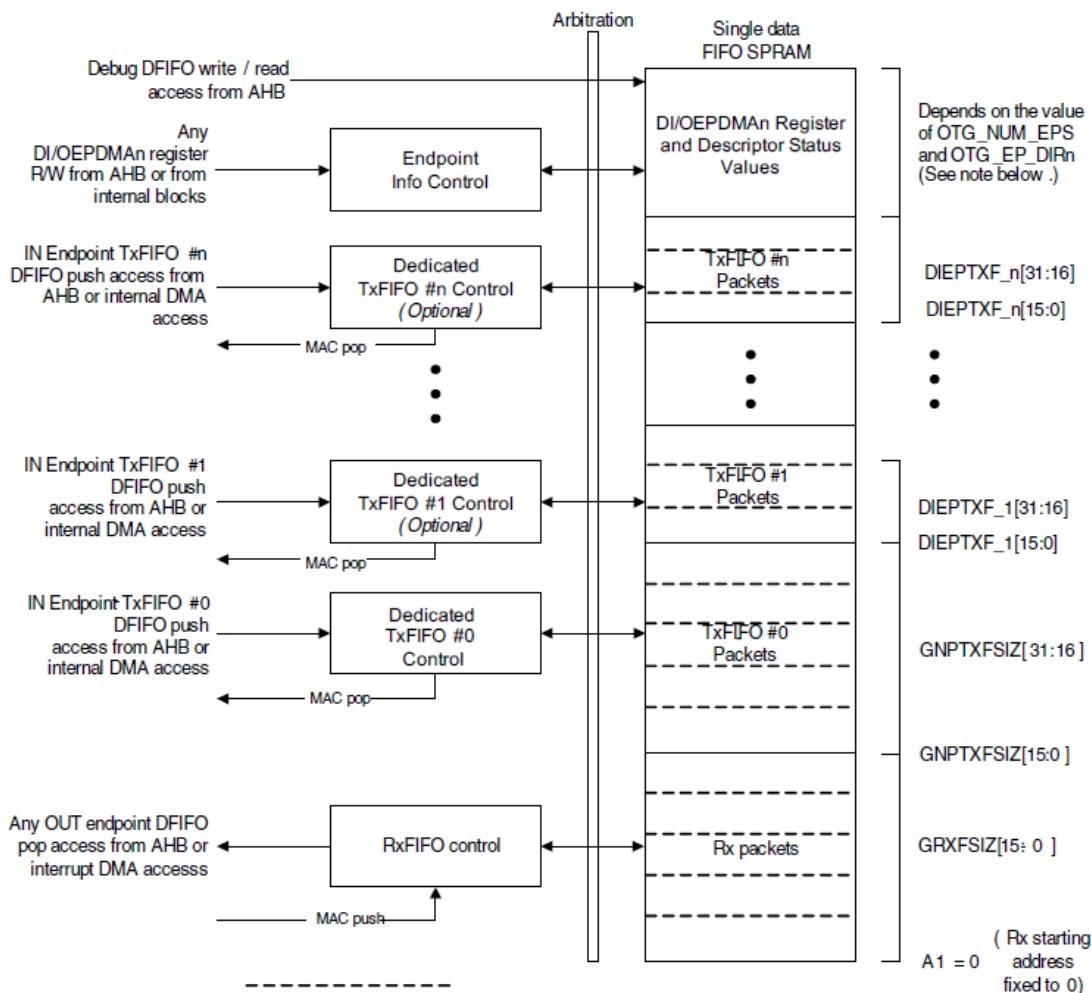


Fig. 26-7 USB OTG 2.0 Controller device mode FIFO address mapping

Note: When the device is operating in non-Scatter Gather Internal DMA mode, the last locations of the SPRAM are used to store the DMAADDR values for each Endpoint (1 location per endpoint). When the device is operating in Scatter Gather mode, then the last locations of the SPRAM store the Base Descriptor address, Current Descriptor address, Current Buffer address, and status quadlet information for each endpoint direction (4 locations per Endpoint). If an Endpoint is bidirectional, then 4 locations will be used for IN, and another 4 for OUT).

26.4 USB OTG2.0 PHY

USB PHY supports dual OTG ports' functions and is fully compliant with USB2.0 specification, and support High-speed (480Mbps), full-speed (12Mbps), low-speed (1.5Mbps) transfer. It provides a complete on-chip transceiver physical solution with ESD protection. A minimum number of external components are needed, which include a 45 ohm resistor for resistance calibration purpose. Its feature contains:

- provide dual UTMI ports
- OTG0 Support UART Bypass Function
- Fully compliant with USB specifications Rev 2.0, 1.1 HOST/Device and OTG V1.2.
- Supports 480Mbps (HS), 12Mbps (FS) & 1.5Mbps(LS) serial data transmission
- Supports low latency hub mode with 40 bit time round trip delay
- 8 bit or 16 bit UTMI interface compliant with UTMI+ specification level 3 Rev 1.
- Loop back BIST mode supported
- Built-in I/O and ESD structure

- On-die self-calibrated HS/FS/LS termination
- 12MHz crystal oscillator with integrated phase-locked loop (PLL) oscillator
- Manufactured in SMIC 65/55nm LL process
- Dual 3.3V / 1.2V supply

26.4.1 Block Diagram

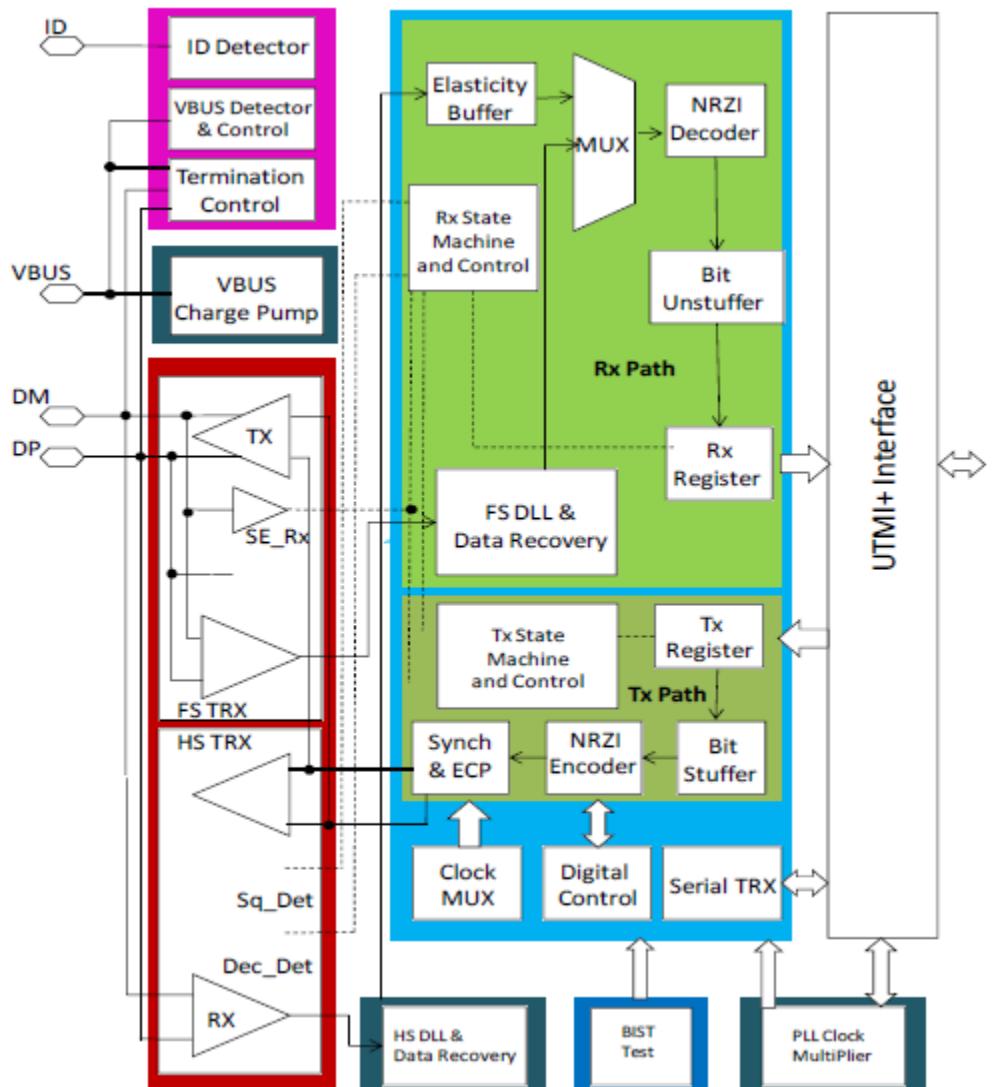


Fig. 26-8 usb phy architecture

HS AFE

The HS AFE contains the low-level analog circuitry, and also the HS differential data transmitter and receiver, to perform HS transmission envelope detection and host disconnection detection. It works in HS mode only.

HS Transmit driver

The HS transmit driver is active only when transmit is asserted. In HS transceiver enabled mode and the transmit state machine has data to send, the XCVR selects input. Data from transmit data path will be driven onto the DP/DM signal lines when enabled.

HS Differential Receiver

When enabled, received HS data will be multiplexed through the receive data path to the receive shift and hold registers. It is active only in HS mode.

transmission envelope detector (Squelch detector)

When the amplitude of the differential signal at a receiver's inputs falls below the squelch threshold, the envelope detector will indicate the invalid data. It must indicate squelch when the signal drops below 100mV differential amplitude, and also, it must indicate that the line is not in the squelch state when the signal exceeds 150mV differential amplitude.

Disconnection envelope detector

In host mode, this envelope detector is active to detect the high speed disconnect state on the line. Disconnection must be indicated when the amplitude of the differential signal at the downstream facing driver's connector is more than 625 mV, and it must not be indicated when the signal amplitude is less than 525 mV.

FS/LS AFE

In FS or LS mode, the FS/LS AFE is active to send and receive the FS or LS data on the USB bus. Also it supports the reset, suspend and resume detection through the data line single ended receivers.

FS/LS Transmitter

The FS/LHS transmitter is active only when transmit is asserted. In FS or LS transceiver enabled mode and the transmit state machine has data to send, the XCVR selects input. Data from transmit data path will be driven onto the DP/DM signal lines when enabled.

FS/LS Differential Receiver

When enabled, received FS or LS data will be multiplexed through the receive data path to the receive shift and hold registers. It is active only in FS or LS modes.

Single ended receivers

The single ended receivers are used for low-speed and full-speed signaling detection.

Digital Core TX Path

The digital core TX path has some blocks responsible for SYNC and EOP generation, data encoding, bit stuffing and data serialization. And meanwhile, also a TX state machine is involved to manage the communication with the controller.

TX Shift/Hold Register

The TX shift/Hold register module consists of an 8-bit primary shift register for parallel/serial conversion and 8-bit hold register used to buffer the next data to serialize. This module is responsible for reading parallel data from the parallel application bus interface upon command and serializing for transmission over USB.

Bit stuffer

To ensure adequate signal transitions, when sending a packet on USB, a bit stuffer is employed by the transmitter. A '0' has to be inserted after every six consecutive ones in the data stream before the data is NRZI encoded, to force a transition in the NRZI data stream.

NRZI Encoder

The High speed, Full speed or low speed serial transmitted data are encoded by The NRZI encoder. As a state transition, a '0' is encoded, and as no state transition, a '1' is encoded.

Transmit state machine

The communication between the controller and the PHY in TX path is controlled by the transmit state machine, which synchronizes the Data with the Sync and the EOP, and also supports the LS, FS and HS Modes.

Digital Core RX Path

The digital core RX path includes blocks responsible for SYNC and EOP detection and stripping, data decoding, bit un-stuffing and data de-serialization. Also a RX state machine is involved to manage the communication with the controller. FS/LS data and clock is recovered in this section.

Elasticity buffer

To compensate for differences between transmitting and receiving clocks, the Elasticity Buffer is used to synchronize the HS extracted data with the PLL internal clock.

Mux

The Mux block allows the data from the HS or FS/LS receivers to be routed to the shared receive logic. The state of the Mux is determined by the Xcvr Select input.

NRZI Decoder

The NRZI is responsible for decoding the High speed or Full speed received NRZI encoded data. A change in level is decoded as '0' and no change in level is decoded as '1'.

Bit Un-stuffer

The Bit Un-stuffer not only recognizes the stuffed bits from the data stream, but also discards them. Also it detects bit stuff error, which is interpreted as HS EOP.

RX Shift/Hold Register

This module de-serializes received data and transmits 8-bit parallel data to the application bus interface. It consists of an 8-bit primary shift register for serial to parallel conversion and an 8-bit hold register for buffering the last de-serialized data byte.

Receiver state machine

The receiver state machine controls the communication between the controller and the PHY in the RX path, strips the SYNC and the EOP from the Data and supports the LS, FS and HS Modes.

PLL Clock Multiplier

This module is composed of the off-chip crystal and the on-chip clock multiplier. It generates the appropriate internal clocks for the UTM and the CLK output signal. All data transfer signals are synchronized with the CLK signal.

External Crystal

The external crystal is composed of a precise resonance frequency crystal and a crystal oscillator. It is optional to have this crystal oscillator integrated on-chip

or have it off-chip. This crystal/crystal oscillator provides a very precise clock of 12 MHz with deviation of ± 100 ppm. The oscillator is not a part of the PHY, but external.

Clock Multiplier

The UTM interface is described as an un-directional/bi-directional 8-bit/16-bit parallel interface and the CLK signal is a 60/30 MHz signal. All data transfer signals should be synchronized with the CLK signal. CLK usable signal is internally implemented which blocks any transitions of CLK until it is usable. Meanwhile, the clock multiplier provides another three clocks in addition to the CLK signal. That is a 480 MHz and 7.5 MHz clock signals.

Clock MUX

The Clock Multiplexer supplies both the transmitter and receiver paths with the adequate bit clock depending on the XcvrSelect signal and to ensure smooth clock switching. It also includes clock gating and power-down features.

Control Logic Block

This block is responsible for controlling, enabling and disabling the different blocks in the system.

OTG Circuitry (optional)

With the OTG circuitry, the system has the capability to dynamically switch between host and peripheral, enable dual role device behavior and point-to-point communication. The OTG circuitry functions as VBUS generation and detection. Both ID detection and terminations control are implemented in it.

ID Detector (optional)

To provide the ID signal that is used to indicate the state of the ID pin on the USB mini receptacle. This pin makes it able to determine which kind of plug is connected and to confirm if the device default state is A device or B device.

VBUS Detector and termination control

The VBUS detector is a set of comparators, functions to monitor and sense the voltage on USB bus power line. For VBUS signaling and discharging, VBUS pull up and pull-down resistors are also implemented.

Automatic Test Functions

- Loop-back test to address all IP components.

In loop-back test mode, all transmitted data packets are received back in an internal loop to check IP functional integrity. There are some digital components that cannot be tested with the scan technique due to the high-speed nature of the digital part. To be regarded as a good idea, Loop-back allows testing full design paths at speed. It should complement the testing suite for digital core to achieve the highest coverage possible. According to the UTMI specification Section 5.18, version 1.05 Page 34, the 8 bits un-directional data bus can be implemented as 8 bits bi-directional one. This implementation will hinder the loop-back test functionality.

26.5 UART BYPASS FUNCITON

When in UART bypass mode, UART2 is connect to USB interface; Otherwise, UART2 use normal UART interface.

Signal	CONNECT	I/O	Description
BYPASSDMDATA0	uart2_sout	I	Data for DM0 Transmitter Digital Bypass
BYPASSDMENO	uoc0_con0[8]	I	DM0 Transmitter Digital Bypass Enable
BYPASSSEL0	uoc0_con0[9]	I	Transmitter Digital Bypass mode Enable
FSVPLUS0	uart2_sin	O	Single-Ended D- Indicator The controller signal indicates the state of the DP during normal operation or UART data reception
OTGDISABLE0	uoc0_con0[4]	I	1'b1: OTG0 disable; 1'b0: OTG0 normal mode
COMMONONNN	uoc0_con0[0]	I	Common Block Power-Down Control This signal controls the power-down signals in PLL blocks when the USB PHY is in Suspend Mode. 1: PLL blocks are powered down. 0: PLL blocks remain powered This signal is a strapping option that must be set prior to a power-on reset and remain static during normal operation.

Note: USB OTG2.0 PHY support UART Bypass Function.

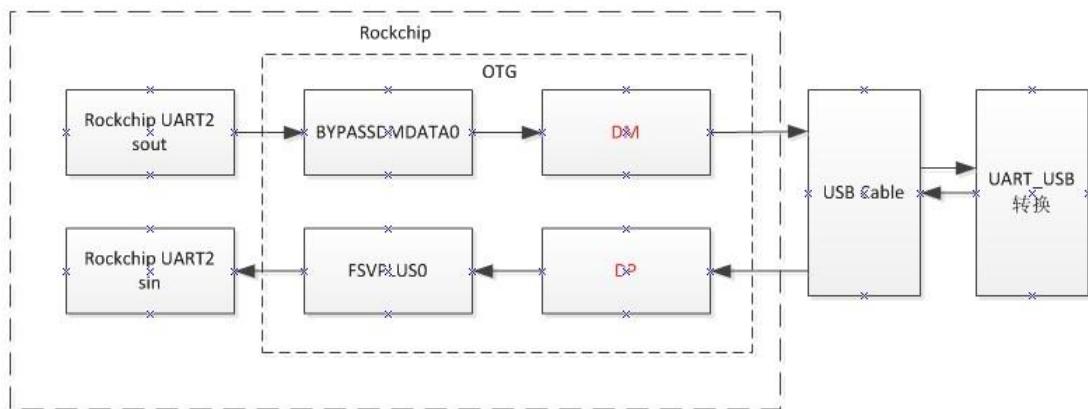


Fig. 26-9 UART Application

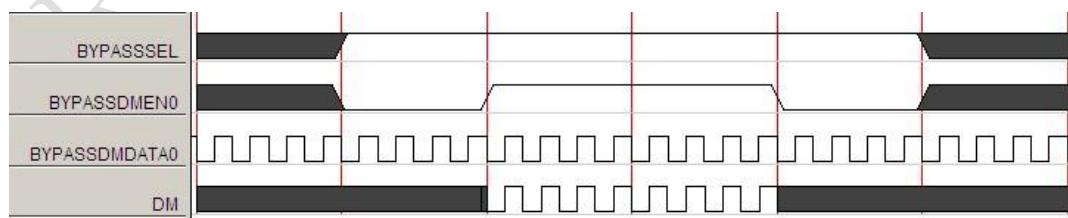


Fig. 26-10 UART Timing Sequence

To use UART and Auto resume functions:

1. Disable the OTG block by setting OTGDISABLE0 to 1'b1.
2. Disable the pull-up resistance on the D+ line by setting OPMODE0[1:0] to 2'b01.

3. To ensure that the XO, Bias, and PLL blocks are powered down in Suspend mode, set COMMONONN to 1'b1.
 4. Place the USB PHY in Suspend mode by setting SUSPENDM0 to 1'b0.
 5. Set BYPASSSEL0 to 1'b1.
 6. To transmit data, controls BYPASSDMEN0, and BYPASSDMDATA0.
- To receive data, monitor FSVPLUS0.

To return to normal operating mode:

1. Ensure that there is no activity on the USB.
2. Set BYPASSSEL0 to 1'b0.
3. Set SUSPENDM0 to 1'b1. Resume the USB PHY.
4. Set COMMONONN to 1'b0.
5. set OTGDISABLE0 to 1'b0.

26.6 Register Description

26.6.1 Register Summary

Name	Offset	Size	Reset Value	Description
USBOTG_GOTGCTL	0x0000	W	0x00000000	Control and Status Register
USBOTG_GOTGINT	0x0004	W	0x00000000	Interrupt Register
USBOTG_GAHBCFG	0x0008	W	0x00000000	AHB Configuration Register
USBOTG_GUSBCFG	0x000c	W	0x00001400	USB Configuration Register
USBOTG_GRSTCTL	0x0010	W	0x80000000	Reset Register
USBOTG_GINTSTS	0x0014	W	0x00000000	Interrupt Register
USBOTG_GINTMSK	0x0018	W	0x00000000	Interrupt Mask Register
USBOTG_GRXSTSR	0x001c	W	0x00000000	Receive Status Debug Read Register
USBOTG_GRXSTSP	0x0020	W	0x00000000	Receive Status Read and Pop Register
USBOTG_GRXFSIZ	0x0024	W	0x00000000	Receive FIFO Size Register
USBOTG_GNPTXFSIZ	0x0028	W	0x00000000	Non-Periodic Transmit FIFO Size Register
USBOTG_GNPTXSTS	0x002c	W	0x00000000	Non-Periodic Transmit FIFO/Queue Status Register
USBOTG_GI2CCTL	0x0030	W	0x11000000	I2C Address Register
USBOTG_GPVNDCTL	0x0034	W	0x00000000	PHY Vendor Control Register
USBOTG_GGPIO	0x0038	W	0x00000000	General Purpose Input / Output Register
USBOTG_GUID	0x003c	W	0x00000000	User ID Register
USBOTG_GSNPSID	0x0040	W	0x00004f54	Core ID Register
USBOTG_GHWCFG1	0x0044	W	0x00000000	User HW Config1 Register
USBOTG_GHWCFG2	0x0048	W	0x00000000	User HW Config2 Register
USBOTG_GHWCFG3	0x004c	W	0x00000000	User HW Config3 Register
USBOTG_GHWCFG4	0x0050	W	0x00000000	User HW Config4 Register

Name	Offset	Size	Reset Value	Description
USBOTG_GLPMCFG	0x0054	W	0x00000000	Core LPM Configuration Register
USBOTG_GPWRDN	0x0058	W	0x00000000	Global Power Down Register
USBOTG_GDFIFOFG	0x005c	W	0x00000000	Global DFIFO Software Configuration Register
USBOTG_GADPCTL	0x0060	W	0x00000000	ADP Timer, Control and Status Register
USBOTG_HPTXFSIZ	0x0100	W	0x00000000	Host Periodic Transmit FIFO Size Register
USBOTG_DIEPTXF _n	0x0104 +4*(n-1)	W	0x00000000	Device Periodic Transmit FIFO-n Size Register n = 1 - 15
USBOTG_HCFG	0x0400	W	0x00000000	Host Configuration Register
USBOTG_HFIR	0x0404	W	0x00000000	Host Frame Interval Register
USBOTG_HFNUM	0x0408	W	0x0000ffff	Host Frame Number/Frame Time Remaining Register
USBOTG_HPTXSTS	0x0410	W	0x00000000	Host Periodic Transmit FIFO/Queue Status Register
USBOTG_HAINT	0x0414	W	0x00000000	Host All Channels Interrupt Register
USBOTG_HAINTMSK	0x0418	W	0x00000000	Host All Channels Interrupt Mask Register
USBOTG_HPRT	0x0440	W	0x00000000	Host Port Control and Status Register
USBOTG_HCCHAR _n	0x0500 +0x20 *n	W	0x00000000	Host Channel-n Characteristics Register n = 0 - 15
USBOTG_HCSPLT _n	0x0504 +0x20 *n	W	0x00000000	Host Channel-n Split Control Register n = 0 - 15
USBOTG_HCINT _n	0x0508 +0x20 *n	W	0x00000000	Host Channel-n Interrupt Register n = 0 - 15
USBOTG_HCINTMSK _n	0x050c +0x20 *n	W	0x00000000	Host Channel-n Interrupt Mask Register n = 0 - 15
USBOTG_HCTSIZ _n	0x0510 +0x20 *n	W	0x00000000	Host Channel-n Transfer Size Register n = 0 - 15
USBOTG_HCDMAN	0x0514 +0x20 *n	W	0x00000000	Host Channel-n DMA Address Register n = 0 - 15
USBOTG_HCDMAB _n	0x051c +0x20 *n	W	0x00000000	Host Channel-n DMA Buffer Address Register n = 0 - 15
USBOTG_DCFG	0x0800	W	0x08200000	Device Configuration Register
USBOTG_DCTL	0x0804	W	0x00002000	Device Control Register
USBOTG_DSTS	0x0808	W	0x00000000	Device Status Register
USBOTG_DIEPMSK	0x0810	W	0x00000000	Device IN Endpoint common interrupt mask register

Name	Offset	Size	Reset Value	Description
USBOTG_DOEPMASK	0x0814	W	0x00000000	Device OUT Endpoint common interrupt mask register
USBOTG_DAINT	0x0818	W	0x00000000	Device All Endpoints interrupt register
USBOTG_DAINTMSK	0x081c	W	0x00000000	Device All Endpoint interrupt mask register
USBOTG_DTKNQR1	0x0820	W	0x00000000	Device IN token sequence learning queue read register1
USBOTG_DTKNQR2	0x0824	W	0x00000000	Device IN token sequence learning queue read register2
USBOTG_DVBUSDIS	0x0828	W	0x00000b8f	Device VBUS discharge time register
USBOTG_DVBUSPULSE	0x082c	W	0x00000000	Device VBUS Pulsing Timer Register
USBOTG_DTHRCTL	0x0830	W	0x08100020	Device Threshold Control Register
USBOTG_DIEPEMPMSK	0x0834	W	0x00000000	Device IN endpoint FIFO empty interrupt mask register
USBOTG_DEACHINT	0x0838	W	0x00000000	Device each endpoint interrupt register
USBOTG_DEACHINTMSK	0x083c	W	0x00000000	Device each endpoint interrupt register mask
USBOTG_DIEPEACHMSKn	0x0840 +4*n	W	0x00000000	Device each IN endpoint -n interrupt Register n = 0 - 15
USBOTG_DOEPEACHMSKn	0x0880 +4*n	W	0x00000000	Device each out endpoint-n interrupt register n = 0 - 15
USBOTG_DIEPCTL0	0x0900	W	0x00008000	Device control IN endpoint 0 control register
USBOTG_DIEPINTn	0x0908 +0x20 *n	W	0x00000000	Device Endpoint-n Interrupt Register n = 0 - 15
USBOTG_DIEPTSIZn	0x0910 +0x20 *n	W	0x00000000	Device endpoint n transfer size register n = 0 - 15
USBOTG_DIEPDMAAn	0x0914 +0x20 *n	W	0x00000000	Device endpoint-n DMA address register n = 0 - 15
USBOTG_DTXFSTSn	0x0918 +0x20 *n	W	0x00000000	Device IN endpoint transmit FIFO status register n = 0 - 15
USBOTG_DIEPDMAFn	0x091c	W	0x00000000	Device endpoint-n DMA buffer address register
USBOTG_DIEPCTLn	0x0920 +0x20 *(n-1)	W	0x00000000	Device endpoint-n control register n = 1 - 15
USBOTG_DOEPCTL0	0x0b00	W	0x00000000	Device control OUT endpoint 0 control register

Name	Offset	Size	Reset Value	Description
USBOTG_DOEPINTn	0x0b08 +0x20 *n	W	0x00000000	Device endpoint-n control register n = 0 - 15
USBOTG_DOEPTSIZn	0x0b10 +0x20 *n	W	0x00000000	Device endpoint n transfer size register n = 0 - 15
USBOTG_DOEPDMAn	0x0b14 +0x20 *n	W	0x00000000	Device Endpoint-n DMA Address Register n = 0 - 15
USBOTG_DOEPDMABn	0x0b1c +0x20 *n	W	0x00000000	Device endpoint-n DMA buffer address register n = 0 - 15
USBOTG_DOEPCTLn	0x0b20 +0x20 *(n-1)	W	0x00000000	Device endpoint-n control register n = 1 - 15
USBOTG_PCGCR	0x0b24	W	0x200b8000	Power and clock gating control register

Notes: *Size* : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** - WORD (32 bits) access

26.6.2 Detail Register Description

USBOTG_GOTGCTL

Address: Operational Base + offset (0x0000)

Control and Status Register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	ChirpEn Chirp on enable This bit when programmed to 1'b1 results in the core asserting chirp_on before sending an actual Chirp "K" signal on USB. This bit is present only if OTG_BC_SUPPORT = 1. If OTG_BC_SUPPORT != 1, this bit is a reserved bit.
26:22	RO	0x00	MultValidBc Multi Valued ID pin Battery Charger ACA inputs in the following order: Bit 26 - rid_float. Bit 25 - rid_gnd Bit 24 - rid_a Bit 23 - rid_b Bit 22 - rid_c These bits are present only if OTG_BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0.
21	RO	0x0	reserved

Bit	Attr	Reset Value	Description
20	RW	0x0	OTGVer OTG version Indicates the OTG revision. 1'b0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP. 1'b1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.
19	RO	0x0	BSesVld B-session valid Indicates the Device mode transceiver status. 1'b0: B-session is not valid. 1'b1: B-session is valid. In OTG mode, you can use this bit to determine if the device is connected or disconnected. Note: If you do not enable OTG features (such as SRP and HNP), the read reset value will be 1. The vbus assigns the values internally for non-SRP or non-HNP configurations.
18	RO	0x0	ASesVld A-session valid Indicates the Host mode transceiver status. 1'b0: A-session is not valid 1'b1: A-session is valid Note: If you do not enable OTG features (such as SRP and HNP), the read reset value will be 1. The vbus assigns the values internally for non-SRP or non-HNP configurations.
17	RO	0x0	DbnTime Long/short debounce time Indicates the debounce time of a detected connection. 1'b0: Long debounce time, used for physical connections (100 ms + 2.5 us) 1'b1: Short debounce time, used for soft connections (2.5 us)
16	RO	0x0	ConIDSts Connector ID Status Indicates the connector ID status on a connect event. 1'b0: The core is in A-Device mode 1'b1: The core is in B-Device mode
15:12	RO	0x0	reserved
11	RW	0x0	DevHNPEn Device HNP Enable The application sets this bit when it successfully receives a SetFeature. SetHNPEnable command from the connected USB host. 1'b0: HNP is not enabled in the application 1'b1: HNP is enabled in the application

Bit	Attr	Reset Value	Description
10	RW	0x0	<p>HstSetHNPEn Host set HNP enable The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. 1'b0: Host Set HNP is not enabled 1'b1: Host Set HNP is enabled</p>
9	RW	0x0	<p>HNPReq HNP request The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. 1'b0: No HNP request 1'b1: HNP request</p>
8	RO	0x0	<p>HstNegScs Host Negotiation Success The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. 1'b0: Host negotiation failure 1'b1: Host negotiation success</p>
7:2	RO	0x0	reserved
1	RW	0x0	<p>SesReq Session Request The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. 1'b0: No session request 1'b1: Session request</p>

Bit	Attr	Reset Value	Description
0	RO	0x0	SesReqScs Session Request Success The core sets this bit when a session request initiation is successful. 1'b0: Session request failure 1'b1: Session request success

USBOTG_GOTGINT

Address: Operational Base + offset (0x0004)

Interrupt Register

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	W1C	0x0	MultiValueChg Multi-Valued input changed This bit when set indicates that there is a change in the value of at least one ACA pin value. This bit is present only if OTG_BC_SUPPORT = 1, otherwise it is reserved.
19	W1C	0x0	DbnceDone Debounce Done The core sets this bit when the debounce is completed after the device connection. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).
18	W1C	0x0	ADevTOUTChg A-Device Timeout Change The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.
17	W1C	0x0	HstNegDet Host Negotiation Detected The core sets this bit when it detects a host negotiation request on the USB
16:10	RO	0x0	reserved
9	W1C	0x0	HstNegSucStsChng Host Negotiation Success Status Change The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure

Bit	Attr	Reset Value	Description
8	W1C	0x0	SesReqSucStsChng Session Request Success Status Change The core sets this bit on the success or failure of a session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.
7:3	RO	0x0	reserved
2	W1C	0x0	SesEndDet Session End Detected The core sets this bit when the utmisrp_bvalid signal is deasserted
1:0	RO	0x0	reserved

USBOTG_GAHBCFG

Address: Operational Base + offset (0x0008)

AHB Configuration Register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	NotiAllDmaWrit Notify All Dma Write Transactions This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1. GAHBCFG.NotiAllDmaWrit = 1. HSOTG core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint. GAHBCFG.NotiAllDmaWrit = 0. HSOTG core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint.

Bit	Attr	Reset Value	Description
21	RW	0x0	<p>RemMemSupp Remote Memory Support This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write Transfers. GAHBCFG.RemMemSupp=1. The int_dma_req output signal is asserted when HSOTG DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from HSOTG. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint. GAHBCFG.RemMemSupp=0. The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the HSOTG Core Boundary and it does not wait for the sys_dma_done signal to complete the DATA transfers.</p>
20:9	RO	0x0	reserved
8	RW	0x0	<p>PTxFEmpLvl Periodic TxFIFO Empty Level Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. 1'b0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty</p>

Bit	Attr	Reset Value	Description
7	RW	0x0	<p>NPTxFEmpLvl Non-Periodic TxFIFO Empty Level This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered.</p> <p>Host mode and with Shared FIFO with device mode:</p> <p>1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is half empty 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is completely empty</p> <p>Dedicated FIFO in device mode:</p> <p>1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty</p>
6	RO	0x0	reserved
5	RW	0x0	<p>DMAEn DMA Enable 1'b0: Core operates in Slave mode 1'b1: Core operates in a DMA mode This bit is always 0 when Slave-Only mode has been selected.</p>

Bit	Attr	Reset Value	Description
4:1	RW	0x0	<p>HBstLen Burst Length/Type This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports, External DMA Mode defines the DMA burst length in terms of 32-bit words:</p> <ul style="list-style-type: none"> 4'b0000: 1 word 4'b0001: 4 words 4'b0010: 8 words 4'b0011: 16 words 4'b0100: 32 words 4'b0101: 64 words 4'b0110: 128 words 4'b0111: 256 words Others: Reserved <p>Internal DMA Mode AHB Master burst type:</p> <ul style="list-style-type: none"> 4'b0000: Single 4'b0001: INCR 4'b0011: INCR4 4'b0101: INCR8 4'b0111: INCR16 Others: Reserved
0	RW	0x0	<p>GlblIntrMsk Global Interrupt Mask The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core.</p> <p>1'b0: Mask the interrupt assertion to the application. 1'b1: Unmask the interrupt assertion to the application.</p>

USBOTG_GUSBCFG

Address: Operational Base + offset (0x000c)

USB Configuration Register

Bit	Attr	Reset Value	Description
31	RW	0x0	<p>CorruptTxpacket Corrupt Tx packet This bit is for debug purposes only. Never set this bit to 1.</p>

Bit	Attr	Reset Value	Description
30	RW	0x0	<p>ForceDevMode Force Device Mode Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. 1'b0: Normal Mode 1'b1: Force Device Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 us is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.</p>
29	RW	0x0	<p>ForceHstMode Force Host Mode Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. 1'b0: Normal Mode 1'b1: Force Host Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 us is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.</p>
28	RW	0x0	<p>TxEndDelay Tx End Delay Writing a 1 to this bit enables the TxEndDelay timers in the core. 1'b0: Normal mode 1'b1: Introduce Tx end delay timers</p>
27	RW	0x0	<p>IC_USB_TrafficCtrl IC_USB Traffic Pull Remove Control When this bit is set, pullup/pulldown resistors are detached from the USB during traffic signaling. This bit is valid only when configuration parameter OTG_ENABLE_IC_USB = 1 and register field GUSBCFG.IC_USBCap is set to 1.</p>
26	RW	0x0	<p>IC_USBCap IC_USB-Capable The application uses this bit to control the IC_USB capabilities. 1'b0: IC_USB PHY Interface is not selected. 1'b1: IC_USB PHY Interface is selected. This bit is writable only if OTG_ENABLE_IC_USB=1 and OTG_FSPHY_INTERFACE!=0. The reset value depends on the configuration parameter OTG_SELECT_IC_USB when OTG_ENABLE_IC_USB = 1. In all other cases, this bit is set to 1'b0 and the bit is read only.</p>

Bit	Attr	Reset Value	Description
25	RW	0x0	<p>ULPIIfDis ULPI Interface Protect Disable Controls circuitry built into the PHY for protecting the ULPI interface when the link tri-states STP and data. Any pull-ups or pull-downs employed by this feature can be disabled. Please refer to the ULPI Specification for more detail.</p> <p>1'b0: Enables the interface protect circuit 1'b1: Disables the interface protect circuit</p>
24	RW	0x0	<p>IndPassThrough Indicator Pass Through Controls whether the Complement Output is qualified with the Internal Vbus Valid comparator before being used in the Vbus State in the RX CMD. Please refer to the ULPI Specification for more detail.</p> <p>1'b0: Complement Output signal is qualified with the Internal VbusValid comparator. 1'b1: Complement Output signal is not qualified with the Internal VbusValid comparator.</p>
23	RW	0x0	<p>IndComple Indicator Complement Controls the PHY to invert the ExternalVbusIndicator input signal, generating the Complement Output. Please refer to the ULPI Specification for more detail</p> <p>1'b0: PHY does not invert External Vbus Indicator signal 1'b1: PHY does invert External Vbus Indicator signal</p>
22	RW	0x0	<p>TermSelDLPulse TermSel DLine Pulsing Selection This bit selects utmi_termselect to drive data line pulse during SRP.</p> <p>1'b0: Data line pulsing using utmi_txvalid (default). 1'b1: Data line pulsing using utmi_termsel.</p>
21	RW	0x0	<p>ULPIExtVbusIndicator ULPI External VBUS Indicator This bit indicates to the ULPI PHY to use an external VBUS over-current indicator.</p> <p>1'b0: PHY uses internal VBUS valid comparator. 1'b1: PHY uses external VBUS valid comparator. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)</p>

Bit	Attr	Reset Value	Description
20	RW	0x0	ULPIExtVbusDrv ULPI External VBUS Drive This bit selects between internal or external supply to drive 5V on VBUS,in ULPI PHY. 1'b0: PHY drives VBUS using internal charge pump (default). 1'b1: PHY drives VBUS using external supply. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
19	RW	0x0	ULPIClkSusM ULPI Clock SuspendM This bit sets the ClockSuspendM bit in the Interface Control register on the ULPI PHY. This bit applies only in serial or carkit modes. 1'b0: PHY powers down internal clock during suspend. 1'b1: PHY does not power down internal clock. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
18	RW	0x0	ULPIAutoRes ULPI Auto Resume This bit sets the AutoResume bit in the Interface Control register on the ULPI PHY. 1'b0: PHY does not use AutoResume feature. 1'b1: PHY uses AutoResume feature. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
17	RW	0x0	ULPIFsLs ULPI FS/LS Select The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is selected on the ULPI PHY. 1'b0: ULPI interface 1'b1: ULPI FS/LS serial interface (Valid only when RTL parameters OTG_HSPHY_INTERFACE = 2 or 3 and OTG_FSPHY_INTERFACE = 1, 2, or 3)
16	RW	0x0	OtgI2CSel UTMIFS or I2C Interface Select The application uses this bit to select the I2C interface. 1'b0: UTMIFS USB 1.1 Full-Speed interface for OTG signals 1'b1: I2C interface for OTG signals This bit is writable only if I2C and UTMIFS were specified for Enable I2C Interface? (parameter OTG_I2C_INTERFACE = 2). Otherwise, reads return 0.

Bit	Attr	Reset Value	Description
15	RW	0x0	<p>PhyLPwrClkSel PHY Low-Power Clock Select Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power.</p> <p>1'b0: 480-MHz Internal PLL clock 1'b1: 48-MHz External Clock In 480 MHz mode, the UTMI interface operates at either 60 or 30-MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS and LS modes. This bit drives the <code>utmi_fs_ls_low_power</code> core output signal, and is valid only for UTMI+ PHYs.</p>
14	RO	0x0	reserved
13:10	RW	0x5	<p>USBTrdTim USB Turnaround Time Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIF(SPRAM). This must be programmed to</p> <p>4'h5: When the MAC interface is 16-bit UTMI+. 4'h9: When the MAC interface is 8-bit UTMI+. Note: The values above are calculated for the minimum AHB frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so if you need the AHB to run at less than 30 MHz, and if USB turnaround time is not critical, these bits can be programmed to a larger value.</p>
9	RW	0x0	<p>HNPCap HNP-Capable The application uses this bit to control the Otg core's HNP capabilities.</p> <p>1'b0: HNP capability is not enabled. 1'b1: HNP capability is enabled. This bit is writable only if an HNP mode was specified for Mode of Operation (parameter <code>OTG_MODE</code>). Otherwise, reads return 0.</p>

Bit	Attr	Reset Value	Description
8	RW	0x0	<p>SRPCap SRP-Capable The application uses this bit to control the Otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session.</p> <p>0: SRP capability is not enabled. 1: SRP capability is enabled.</p> <p>This bit is writable only if an SRP mode was specified for Mode of Operation (parameter OTG_MODE). Otherwise, reads return 0.</p>
7	RW	0x0	<p>DDRSel ULPI DDR Select The application uses this bit to select a Single Data Rate (SDR) or Double Data Rate (DDR) or ULPI interface.</p> <p>1'b0: Single Data Rate ULPI Interface, with 8-bit-wide data bus 1'b1: Double Data Rate ULPI Interface, with 4-bit-wide data bus</p>
6	RW	0x0	<p>PHYSel USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver.</p> <p>1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY 1'b1: USB 1.1 full-speed serial transceiver If a USB 1.1 Full-Speed Serial Transceiver interface was not selected (parameter OTG_FSPHY_INTERFACE = 0), this bit is always 0, with Write Only access. If a high-speed PHY interface was not selected (parameter OTG_HSPHY_INTERFACE = 0), this bit is always 1, with Write Only access. If both interface types were selected (parameters have non-zero values), the application uses this bit to select which interface is active, and access is Read and Write.</p>

Bit	Attr	Reset Value	Description
5	RW	0x0	<p>FSIntf Full-Speed Serial Interface Select The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. 1'b0: 6-pin unidirectional full-speed serial interface 1'b1: 3-pin bidirectional full-speed serial interface If a USB 1.1 Full-Speed Serial Transceiver interface was not selected (parameter OTG_FSPHY_INTERFACE = 0), this bit is always 0, with Write Only access. If a USB 1.1 FS interface was selected (parameter OTG_FSPHY_INTERFACE! = 0), then the application can set this bit to select between the 3- and 6-pin interfaces, and access is Read and Write.</p>
4	RW	0x0	<p>ULPI_UTMI_Sel ULPI or UTMI+ Select The application uses this bit to select either a UTMI+ interface or ULPI Interface. 1'b0: UTMI+ Interface 1'b1: ULPI Interface This bit is writable only if UTMI+ and ULPI was specified for High-Speed PHY Interface(s) (parameter OTG_HSPHY_INTERFACE = 3). Otherwise, reads return either 0 or 1, depending on the interface selected using the OTG_HSPHY_INTERFACE parameter.</p>
3	RW	0x0	<p>PHYIf PHY Interface The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode. 1'b0: 8 bits 1'b1: 16 bits This bit is writable only if UTMI+ and ULPI were selected (parameter OTG_HSPHY_DWIDTH = 3). Otherwise, this bit returns the value for the power-on interface selected during configuration.</p>

Bit	Attr	Reset Value	Description
2:0	RW	0x0	<p>TOutCal HS/FS Timeout Calibration The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed inter-packet timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the line state condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are:</p> <p>High-speed operation: One 30-MHz PHY clock = 16 bit times One 60-MHz PHY clock = 8 bit times</p> <p>Full-speed operation: One 30-MHz PHY clock = 0.4 bit times One 60-MHz PHY clock = 0.2 bit times One 48-MHz PHY clock = 0.25 bit times</p>

USBOTG_GRSTCTL

Address: Operational Base + offset (0x0010)

Reset Register

Bit	Attr	Reset Value	Description
31	RO	0x1	AHBIdle AHB Master Idle Indicates that the AHB Master State Machine is in the IDLE condition.
30	RO	0x0	DMAReq DMA Request Signal Indicates that the DMA request is in progress. Used for debug.
29:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:6	RW	0x00	<p>TxFNum Tx FIFO Number This is the FIFO number that must be flushed using the Tx FIFO Flush bit. This field must not be changed until the core clears the Tx FIFO Flush bit.</p> <p>5'h0: Non-periodic Tx FIFO flush in Host mode; Non-periodic Tx FIFO flush in device mode when in shared FIFO operation. Tx FIFO 0 flush in device mode when in dedicated FIFO mode.</p> <p>5'h1: Periodic Tx FIFO flush in Host mode: Periodic Tx FIFO 1 flush in Device mode when in shared FIFO operation; TX FIFO 1 flush in device mode when in dedicated FIFO mode.</p> <p>5'h2: Periodic Tx FIFO 2 flush in Device mode when in shared FIFO operation: TX FIFO 2 flush in device mode when in dedicated FIFO mode.</p> <p>...</p> <p>5'hF: Periodic Tx FIFO 15 flush in Device mode when in shared FIFO operation: TX FIFO 15 flush in device mode when in dedicated FIFO mode.</p> <p>5'h10: Flush all the transmit FIFOs in device or host mode.</p>
5	RWSC	0x0	<p>TxFFlsh Tx FIFO Flush This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the Tx FIFO nor reading from the Tx FIFO. Verify using these registers: Read NAK Effective Interrupt ensures the core is not reading from the FIFO. Write GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are re-configured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.</p>

Bit	Attr	Reset Value	Description
4	RWSC	0x0	RxFFlsh Rx FIFO Flush The application can flush the entire Rx FIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the Rx FIFO nor writing to the Rx FIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or AHB clock) to clear.
3	RWSC	0x0	INTknQFlsh IN Token Sequence Learning Queue Flush This bit is valid only if OTG_ENDED_TX_FIFO = 0. The application writes this bit to flush the IN Token Sequence Learning Queue.
2	W1C	0x0	FrmCntrRst Host Frame Counter Reset The application writes this bit to reset the (micro) frame number counter inside the core. When the (micro) frame counter is reset, the subsequent SOF sent out by the core has a (micro) frame number of 0.
1	RWSC	0x0	Reset A write to this bit issues a soft reset to the Otg_power_dn module of the core.

Bit	Attr	Reset Value	Description
0	RWSC	0x0	<p>CSftRst Core Soft Reset Resets the hclk and phy_clock domains as follows: Clears the interrupts and all the CSR registers except the following register bits:</p> <ul style="list-style-type: none"> PCGCCTL.RstPdwnModule PCGCCTL.GateHclk PCGCCTL.PwrClmp PCGCCTL.StopPPhyLPwrClkSelClk GUSBCFG.PhyLPwrClkSel GUSBCFG.DDRSel GUSBCFG.PHYSel GUSBCFG.FSIntf GUSBCFG.ULPI_UTMI_Sel GUSBCFG.PHYIf HCFG.FSLSPclkSel DCFG.DevSpd GPIO GPWRDN GADPCTL <p>All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. When Hibernation or ADP feature is enabled, the PMU module is not reset by the Core Soft Reset. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation.</p>

USBOTG_GINTSTS

Address: Operational Base + offset (0x0014)

Interrupt Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	<p>WkUpInt Resume/Remote Wakeup Detected Interrupt Wakeup Interrupt during Suspend (L2) or LPM(L1) state. During Suspend(L2): Device Mode: This interrupt is asserted only when Host Initiated Resume is detected on USB. Host Mode: This interrupt is asserted only when Device Initiated Remote Wakeup is detected on USB. During LPM(L1): Device Mode: This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB. Host Mode: This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.</p>
30	W1C	0x0	<p>SessReqInt Session Request/New Session Detected Interrupt In Host mode, this interrupt is asserted when a session request is detected from the device. In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmisrp_bvalid signal goes high.</p>
29	W1C	0x0	<p>DisconnInt Disconnect Detected Interrupt This interrupt is asserted when a device disconnect is detected.</p>
28	W1C	0x0	<p>ConIDSstsChng Connector ID Status Change This interrupt is asserted when there is a change in connector ID status.</p>
27	W1C	0x0	<p>LPM_Int LPM Transaction Received Interrupt Device Mode : This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response. Host Mode : This interrupt is asserted when the device responds to an LPM transaction with a non-ERRORed response or when the host core has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt). This field is valid only if the Core LPM Configuration register's LPMCapable (LPMCap) field is set to 1.</p>

Bit	Attr	Reset Value	Description
26	RO	0x0	PTxFEmp Periodic TxFIFO Empty This interrupt is asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).
25	RO	0x0	HChInt Host Channels Interrupt The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.
24	RO	0x0	PrtInt Host Port Interrupt The core sets this bit to indicate a change in port status of one of the OTG core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.
23	RW	0x0	ResetDet Reset Detected Interrupt The core asserts this interrupt in Device mode when it detects a reset on the USB in Partial Power-Down mode when the device is in Suspend. This interrupt is not asserted in Host mode.
22	W1C	0x0	FetSusp Data Fetch Suspended This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm.

Bit	Attr	Reset Value	Description
21	W1C	0x0	<p>incomlP Incomplete Periodic Transfer In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current micro-frame. Incomplete Isochronous OUT Transfer (incompISOOUT) The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current micro-frame. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.</p>
20	W1C	0x0	<p>incompISOIN Incomplete Isochronous IN Transfer The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current micro-frame. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register. Note: This interrupt is not asserted in Scatter/Gather DMA mode.</p>
19	RO	0x0	<p>OEPInt OUT Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.</p>

Bit	Attr	Reset Value	Description
18	RO	0x0	IEPInt IN Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.
17	W1C	0x0	EPMis Endpoint Mismatch Interrupt Note: This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.
16	W1C	0x0	RstrDoneInt Restore Done Interrupt The core sets this bit to indicate that the restore command after Hibernation was completed by the core. The core continues from Suspended state into the mode dictated by PCGCCTL.RestoreMode field. This bit is valid only when Hibernation feature is enabled.
15	W1C	0x0	EOPF End of Periodic Frame Interrupt Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.
14	W1C	0x0	ISOOutDrop Isochronous OUT Packet Dropped Interrupt The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.
13	W1C	0x0	EnumDone Enumeration Done The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.

Bit	Attr	Reset Value	Description
12	W1C	0x0	USBRst USB Reset The core sets this bit to indicate that a reset is detected on the USB.
11	W1C	0x0	USBSusp USB Suspend The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time.
10	W1C	0x0	ErlySusp Early Suspend The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.
9	W1C	0x0	I2CINT I2C Interrupt The core sets this interrupt when I2C access is completed on the I2C interface. This field is used only if the I2C interface was enabled . Otherwise, reads return 0.
8	W1C	0x0	ULPICKINT ULPI Carkit Interrupt This field is used only if the Carkit interface was enabled . Otherwise, reads return 0. The core sets this interrupt when a ULPI Carkit interrupt is received. The core's PHY sets ULPI Carkit interrupt in UART or Audio mode. I2C Carkit Interrupt (I2CCKINT) This field is used only if the I2C interface was enabled . Otherwise, reads return 0.The core sets this interrupt when a Carkit interrupt is received. The core's PHY sets the I2C Carkit interrupt in Audio mode.
7	RO	0x0	GOUTNakEff Global OUT NAK Effective Indicates that the Set Global OUT NAK bit in the Device Control register DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).

Bit	Attr	Reset Value	Description
6	RO	0x0	GINNakEff Global IN Non-Periodic NAK Effective Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak), set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Nonperiodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.
5	RO	0x0	NPTxFEmp Non-Periodic Tx FIFO Empty This interrupt is valid only when OTG_ENDED_TX_FIFO = 0. This interrupt is asserted when the Non-periodic Tx FIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic Tx FIFO Empty Level bit in the Core AHB Configuration register(GAHBCFG.NPTxFEmpLvl).
4	RO	0x0	RxFLvl RxFIFO Non-Empty Indicates that there is at least one packet pending to be read from the RxFIFO.
3	W1C	0x0	Sof Start of (micro)Frame In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF(HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro) frame number. This interrupt is seen only when the core is operating at either HS or FS.
2	RO	0x0	OTGInt OTG Interrupt The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.

Bit	Attr	Reset Value	Description
1	W1C	0x0	ModeMis Mode Mismatch Interrupt The core sets this bit when the application is trying to access: A Host mode register, when the core is operating in Device mode; A Device mode register, when the core is operating in Host mode. The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and does not affect the operation of the core.
0	RO	0x0	CurMod Current Mode of Operation Indicates the current mode. 1'b0: Device mode 1'b1: Host mode

USBOTG_GINTMSK

Address: Operational Base + offset (0x0018)

Interrupt Mask Register

Bit	Attr	Reset Value	Description
31	RW	0x0	WkUpIntMsk Resume/Remote Wakeup Detected Interrupt Mask
30	RW	0x0	SessReqIntMsk Session Request/New Session Detected Interrupt Mask
29	RW	0x0	DisconnIntMsk Disconnect Detected Interrupt Mask
28	RW	0x0	ConIDStsChngMsk Connector ID Status Change Mask
27	RW	0x0	LPM_IntMsk LPM Transaction Received Interrupt Mask
26	RW	0x0	PTxFEmpMsk Periodic TxFIFO Empty Mask
25	RW	0x0	HChIntMsk Host Channels Interrupt Mask
24	RW	0x0	PrtIntMsk Host Port Interrupt Mask
23	RW	0x0	ResetDetMsk Reset Detected Interrupt Mask
22	RW	0x0	FetSuspMsk Data Fetch Suspended Mask
21	RW	0x0	incomIPMsk_incompISOOUTMsk Incomplete Periodic Transfer Mask(Host only) Incomplete Isochronous OUT Transfer Mask(Device only)
20	RW	0x0	incompISOINMsk Incomplete Isochronous IN Transfer Mask
19	RW	0x0	OEPIntMsk OUT Endpoints Interrupt Mask

Bit	Attr	Reset Value	Description
18	RW	0x0	IEPIntMsk IN Endpoints Interrupt Mask
17	RW	0x0	EPMisMsk Endpoint Mismatch Interrupt Mask
16	RW	0x0	RstrDoneIntMsk Restore Done Interrupt Mask This field is valid only when Hibernation feature is enabled.
15	RW	0x0	EOPFMsk End of Periodic Frame Interrupt Mask
14	RW	0x0	ISOOutDropMsk Isochronous OUT Packet Dropped Interrupt Mask
13	RW	0x0	EnumDoneMsk Enumeration Done Mask
12	RW	0x0	USBRstMsk USB Reset Mask
11	RW	0x0	USBSuspMsk USB Suspend Mask
10	RW	0x0	ErlySuspMsk Early Suspend Mask
9	RW	0x0	I2CIntMsk I2C Interrupt Mask
8	RW	0x0	ULPICKINTMsk_I2CCKINTMsk ULPI Carkit Interrupt Mask (ULPICKINTMsk) I2C Carkit Interrupt Mask (I2CCKINTMsk)
7	RW	0x0	GOUTNakEffMsk Global OUT NAK Effective Mask
6	RW	0x0	GINNakEffMsk Global Non-periodic IN NAK Effective Mask
5	RW	0x0	NPTxFEmpMsk Non-periodic TxFIFO Empty Mask
4	RW	0x0	RxFLvIMsk Receive FIFO Non-Empty Mask
3	RW	0x0	SofMsk Start of (micro)Frame Mask
2	RW	0x0	OTGIntMsk OTG Interrupt Mask
1	RW	0x0	ModeMisMsk Mode Mismatch Interrupt Mask
0	RO	0x0	reserved

USBOTG_GRXSTSR

Address: Operational Base + offset (0x001c)

Receive Status Debug Read Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24:21	RO	0x0	<p>FN Frame Number (Device Only) This is the least significant 4 bits of the (micro) frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.</p>
20:17	RO	0x0	<p>PktSts Packet Status Indicates the status of the received packet(Host Only) 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved Indicates the status of the received packet(Device only) 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved</p>
16:15	RO	0x0	<p>DPID Data PID Indicates the Data PID of the received packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA</p>
14:4	RW	0x000	<p>BCnt Byte Count Indicates the byte count of the received data packet.</p>
3:0	RO	0x0	<p>ChNum_EPNum Channel Number(Host) Endpoint Number(Device) (Host Only) Indicates the channel number to which the current received packet belongs. (Device Only) Indicates the endpoint number to which the current received packet belongs.</p>

USBOTG_GRXSTSP

Address: Operational Base + offset (0x0020)

Receive Status Read and Pop Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:21	RO	0x0	<p>FN Frame Number (Device Only) This is the least significant 4 bits of the (micro) frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.</p>
20:17	RO	0x0	<p>PktSts Packet Status Indicates the status of the received packet(Host Only) 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved Indicates the status of the received packet(Device only) 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved</p>
16:15	RO	0x0	<p>DPID Data PID Indicates the Data PID of the received OUT data packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA</p>
14:4	RO	0x000	<p>BCnt Byte Count Indicates the byte count of the received data packet.</p>
3:0	RO	0x0	<p>ChNum_EPNum Channel Number(Host) Endpoint Number(Device) (Host Only) Indicates the channel number to which the current received packet belongs. (Device Only) Indicates the endpoint number to which the current received packet belongs.</p>

USBOTG_GRXFSIZ

Address: Operational Base + offset (0x0024)

Receive FIFO Size Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RxFDep RxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16, Maximum value is 32,768. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. If Enable Dynamic FIFO Sizing? was deselected, these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected , you can write a new value in this field. You can write a new value in this field. Programmed values must not exceed the power-on value.

USBOTG_GNPTXFSIZ

Address: Operational Base + offset (0x0028)

Non-Periodic Transmit FIFO Size Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	NPTxFDep Non-periodic TxFIFO For host mode, this field is always valid. For Device mode, this field is valid only when OTG_ENDED_TX_FIFO==0. This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 This field is determined by Enable Dynamic FIFO Sizing. OTG_DFIFO_DYNAMIC = 0: These flops are optimized, and reads return the Power on value. OTG_DFIFO_DYNAMIC = 1: The application can write a new value in this field. Programmed values must not exceed the power-on value. The power-on reset value of this field is specified by OTG_ENDED_TX_FIFO: OTG_ENDED_TX_FIFO = 0:The reset value is the Largest Non-periodic Tx Data FIFO Depth parameter, OTG_TX_NPERIO_DFIFO_DEPTH. OTG_ENDED_TX_FIFO = 1: The reset value is parameter OTG_TX_HNPERIO_DFIFO_DEPTH.

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>NPTxFStAddr Non-periodic Transmit RAM For host mode, this field is always valid. This field contains the memory start address for Non-periodic Transmit FIFO RAM. This field is determined by Enable Dynamic FIFO Sizing?(OTG_DFIFO_DYNAMIC):</p> <ul style="list-style-type: none"> OTG_DFIFO_DYNAMIC = 0: These flops are optimized, and reads return the power-on value. OTG_DFIFO_DYNAMIC = 1: The application can write a new value in this field. <p>Programmed values must not exceed the power-on value. The power-on reset value of this field is specified by Largest Rx Data FIFO Depth (parameter OTG_RX_DFIFO_DEPTH).</p>

USBOTG_GNPTXSTS

Address: Operational Base + offset (0x002c)

Non-Periodic Transmit FIFO/Queue Status Register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RO	0x00	<p>NPTxQTop Top of the Non-periodic Transmit Request Queue Entry in the Non-periodic Tx Request Queue that is currently being processed by the MAC.</p> <p>Bits [30:27]: Channel/endpoint number Bits [26:25]:</p> <ul style="list-style-type: none"> 2'b00: IN/OUT token 2'b01: Zero-length transmit packet (device IN/host OUT) 2'b10: PING/CSPLIT token 2'b11: Channel halt command <p>Bit [24]: Terminate (last entry for selected channel/endpoint)</p>
23:16	RO	0x00	<p>NPTxQSpcAvail Non-periodic Transmit Request Queue Space Available Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests.</p> <p>8'h0: Non-periodic Transmit Request Queue is full 8'h1: 1 location available 8'h2: 2 locations available n: n locations available (0 <=n <= 8) Others: Reserved</p>

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	<p>NPTxFSpAvail Non-periodic TxFIFO Space Avail Indicates the amount of free space available in the Non-periodic TxFIFO. Values are in terms of 32-bit words.</p> <p>16'h0: Non-periodic TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 <=n <=32,768) 16'h8000: 32,768 words available Others: Reserved</p>

USBOTG_GI2CCTL

Address: Operational Base + offset (0x0030)

I2C Address Register

Bit	Attr	Reset Value	Description
31	RWSC	0x0	<p>BsyDne I2C Busy/Done The application sets this bit to 1'b1 to start a request on the I2C interface. When the transfer is complete, the core de-asserts this bit to 1'b0. As long as the bit is set, indicating that the I2C interface is busy, the application cannot start another request on the interface.</p>
30	RW	0x0	<p>RW Read/Write Indicator Indicates whether a read or write register transfer must be performed on the interface. Read/write bursting is not supported for registers. 1'b1: Read 1'b0: Write</p>
29	RO	0x0	reserved
28	RW	0x1	<p>I2CDatSe0 I2C DatSe0 USB Mode Selects the FS interface USB mode. 1'b1: VP_VM USB mode 1'b0: DAT_SE0 USB mode</p>
27:26	RW	0x0	<p>I2CDevAdr I2C Device Address Selects the address of the I2C Slave on the USB 1.1 full-speed serial transceiver that the core uses for OTG signaling. 2'b00: 7'h2C 2'b01: 7'h2D 2'b10: 7'h2E 2'b11: 7'h2F</p>

Bit	Attr	Reset Value	Description
25	RW	0x0	I2CSuspCtl I2C Suspend Control Selects how Suspend is connected to a full-speed transceiver in I2C mode. 1'b0: Use the dedicated utmi_suspend_n pin 1'b1: Use an I2C write to program the Suspend bit in the PHY register
24	RO	0x1	Ack I2C ACK Indicates whether an ACK response was received from the I2C Slave. This bit is valid when BsyDne is cleared by the core, after application has initiated an I2C access. 1'b0: NAK 1'b1: ACK
23	RW	0x0	I2CEn I2C Enable Enables the I2C Master to initiate I2C transactions on the I2C interface
22:16	RW	0x00	Addr I2C Address This is the 7-bit I2C device address used by software to access any external I2C Slave, including the I2C Slave on a USB 1.1 OTG full-speed serial transceiver. Software can change this address to access different I2C Slaves.
15:8	RW	0x00	RegAddr I2C Register Addr This field programs the address of the register to be read from or written to.
7:0	RW	0x00	RWData I2C Read/Write Data After a register read operation, this field holds the read data for the application. During a write operation, the application can use this register to program the write data to be written to a register. During writes, this field holds the write data.

USBOTG_GPVNDCTL

Address: Operational Base + offset (0x0034)

PHY Vendor Control Register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31	RWSC	0x0	DisUlpiDrvr Disable ULPI Drivers This field is used only if the Carkit interface was enabled (OTG_ULPI_CARKIT = 1). Otherwise, reads return 0. The application sets this bit when it has finished processing the ULPI Carkit Interrupt (GINTSTS.ULPICKINT). When set, the Otg core disables drivers for output signals and masks input signal for the ULPI interface. Otg clears this bit before enabling the ULPI interface.
30:28	RO	0x0	reserved
27	RWSC	0x0	VStsDone VStatus Done The core sets this bit when the vendor control access is done. This bit is cleared by the core when the application sets the New Register Request bit (bit 25).
26	RO	0x0	VStsBsy VStatus Busy The core sets this bit when the vendor control access is in progress and clears this bit when done.
25	RWSC	0x0	NewRegReq New Register Request The application sets this bit for a new vendor control access.
24:23	RO	0x0	reserved
22	RW	0x0	RegWr Register Write Set this bit for register writes, and clear it for register reads.
21:16	RW	0x00	RegAddr Register Address The 6-bit PHY register address for immediate PHY Register Set access. Set to 6'h2F for Extended PHY Register Set access.
15:8	RW	0x00	VCtrl UTMI+ Vendor Control Register Address The 4-bit register address a vendor defined 4-bit parallel output bus. Bits 11:8 of this field are placed on utmi_vcontrol[3:0]. ULPI Extended Register Address (ExtRegAddr) The 6-bit PHY extended register address.
7:0	RW	0x00	RegData Register Data Contains the write data for register write. Read data for register read, valid when VStatus Done is set.

USBOTG_GGPIO

Address: Operational Base + offset (0x0038)

General Purpose Input/Output Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	GPO General Purpose Output This field is driven as an output from the core, gp_o[15:0]. The application can program this field to determine the corresponding value on the gp_o[15:0] output.
15:0	RO	0x0000	GPI General Purpose Input This field's read value reflects the gp_i[15:0] core input value.

USBOTG_GUID

Address: Operational Base + offset (0x003c)

User ID Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	UserID Application-programmable ID field.

USBOTG_GSNPSID

Address: Operational Base + offset (0x0040)

Core ID Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00004f54	CoreID Release number of the core being used

USBOTG_GHWCFG1

Address: Operational Base + offset (0x0044)

User HW Config1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	epdir Endpoint Direction This 32-bit field uses two bits per endpoint to determine the endpoint direction. Endpoint Bits [31:30]: Endpoint 15 direction Bits [29:28]: Endpoint 14 direction ... Bits [3:2]: Endpoint 1 direction Bits[1:0]: Endpoint 0 direction (always BIDIR) Direction 2'b00: BIDIR (IN and OUT) endpoint 2'b01: IN endpoint 2'b10: OUT endpoint 2'b11: Reserved

USBOTG_GHWCFG2

Address: Operational Base + offset (0x0048)

User HW Config2 Register

Bit	Attr	Reset Value	Description
31	RO	0x0	OTG_ENABLE_IC_USB IC_USB mode specified for mode of operation (parameter OTG_ENABLE_IC_USB). To choose IC_USB_MODE, both OTG_FSPHY_INTERFACE and OTG_ENABLE_IC_USB must be 1.
30:26	RO	0x00	TknQDepth Device Mode IN Token Sequence Learning Queue Depth Range: 0-30
25:24	RO	0x0	PTxQDepth Host Mode Periodic Request Queue Depth 2'b00: 2 2'b01: 4 2'b10: 8 Others: Reserved
23:22	RO	0x0	NPTxQDepth Non-periodic Request Queue Depth 2'b00: 2 2'b01: 4 2'b10: 8 Others: Reserved
21	RO	0x0	reserved
20	RO	0x0	MultiProcIntrpt Multi-Processor Interrupt Enabled 1'b0: No 1'b1: Yes
19	RO	0x0	DynFifoSizing Dynamic FIFO Sizing Enabled 1'b0: No 1'b1: Yes
18	RO	0x0	PerioSupport Periodic OUT Channels Supported in Host Mode 1'b0: No 1'b1: Yes
17:14	RO	0x0	NumHstChnl Number of Host Channels Indicates the number of host channels supported by the core in Host mode. The range of this field is 0-15: 0 specifies 1 channel, 15 specifies 16 channels.
13:10	RO	0x0	NumDevEps Number of Device Endpoints Indicates the number of device endpoints supported by the core in Device mode in addition to control endpoint 0. The range of this field is 1-15.

Bit	Attr	Reset Value	Description
9:8	RO	0x0	FSPhyType Full-Speed PHY Interface Type 2'b00: Full-speed interface not supported 2'b01: Dedicated full-speed interface 2'b10: FS pins shared with UTMI+ pins 2'b11: FS pins shared with ULPI pins
7:6	RO	0x0	HSPhyType High-Speed PHY Interface Type 2'b00: High-Speed interface not supported 2'b01: UTMI+ 2'b10: ULPI 2'b11: UTMI+ and ULPI
5	RO	0x0	SingPnt Point-to-Point 1'b0: Multi-point application (hub and split support) 1'b1: Single-point application (no hub and no split support)
4:3	RO	0x0	OtgArch Architecture 2'b00: Slave-Only 2'b01: External DMA 2'b10: Internal DMA Others: Reserved
2:0	RO	0x0	OtgMode Mode of Operation 3'b000: HNP- and SRP-Capable OTG (Host and Device) 3'b001: SRP-Capable OTG (Host and Device) 3'b010: Non-HNP and Non-SRP Capable OTG (Host and Device) 3'b011: SRP-Capable Device 3'b100: Non-OTG Device 3'b101: SRP-Capable Host 3'b110: Non-OTG Host Others: Reserved

USBOTG_GHWCFG3

Address: Operational Base + offset (0x004c)

User HW Config3 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	DfifoDepth DFIFO Depth This value is in terms of 32-bit words. Minimum value is 32 Maximum value is 32,768
15	RO	0x0	OTG_ENABLE_LPM LPM mode specified for Mode of Operation (parameter OTG_ENABLE_LPM).

Bit	Attr	Reset Value	Description
14	RO	0x0	OTG_BC_SUPPORT This bit indicates the HS OTG controller support for Battery Charger. 1'b0: No Battery Charger Support 1'b1: Battery Charger support present.
13	RO	0x0	OTG_ENABLE_HSIC HSIC mode specified for Mode of Operation (parameter OTG_ENABLE_HSIC). Value Range: 0-1 1'b1: HSIC-capable with shared UTMI PHY interface 1'b0: Non-HSIC-capable
12	RO	0x0	OTG_AD_P_SUPPORT This bit indicates whether ADP logic is present within or external to the HS OTG controller 1'b0: No ADP logic present with HSOTG controller 1'b1: ADP logic is present along with HSOTG controller.
11	RO	0x0	RstType Reset Style for Clocked always Blocks in RTL 1'b0: Asynchronous reset is used in the core 1'b1: Synchronous reset is used in the core
10	RO	0x0	OptFeature Optional Features Removed Indicates whether the User ID register, GPIO interface ports, and SOF toggle and counter ports were removed for gate count optimization by enabling Remove Optional Features? 1'b0: No 1'b1: Yes
9	RO	0x0	VndCtlSupt Vendor Control Interface Support 1'b0: Vendor Control Interface is not available on the core. 1'b1: Vendor Control Interface is available.
8	RO	0x0	I2CIntSel I2C Selection 1'b0: I2C Interface is not available on the core. 1'b1: I2C Interface is available on the core.
7	RO	0x0	OtgEn OTG Function Enabled The application uses this bit to indicate the Otg core's OTG capabilities. 1'b0: Not OTG capable 1'b1: OTG Capable

Bit	Attr	Reset Value	Description
6:4	RO	0x0	PktSizeWidth Width of Packet Size Counters 3'b000: 4 bits 3'b001: 5 bits 3'b010: 6 bits 3'b011: 7 bits 3'b100: 8 bits 3'b101: 9 bits 3'b110: 10 bits Others: Reserved
3:0	RO	0x0	XferSizeWidth Width of Transfer Size Counters 4'b0000: 11 bits 4'b0001: 12 bits ... 4'b1000: 19 bits Others: Reserved

USBOTG_GHWCFG4

Address: Operational Base + offset (0x0050)

User HW Config4 Register

Bit	Attr	Reset Value	Description
31	RO	0x0	SGDMA Scatter/Gather DMA 1'b1: Dynamic configuration
30	RO	0x0	SGDMACon Scatter/Gather DMA configuration 1'b0: Non-Scatter/Gather DMA configuration 1'b1: Scatter/Gather DMA configuration
29:26	RO	0x0	INEps Number of Device Mode IN Endpoints Including Control Endpoint Range 0 -15 0:1 IN Endpoint 1:2 IN Endpoints 15:16 IN Endpoints
25	RW	0x0	DedFifoMode Enable Dedicated Transmit FIFO for device IN Endpoints 1'b0: Dedicated Transmit FIFO Operation not enabled. 1'b1: Dedicated Transmit FIFO Operation enabled.
24	RW	0x0	SessEndFltr session_end Filter Enabled 1'b0: No filter 1'b1: Filter

Bit	Attr	Reset Value	Description
23	RW	0x0	BValidFltr "b_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
22	RO	0x0	AValidFltr "a_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
21	RO	0x0	VBusValidFltr "vbus_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
20	RO	0x0	IddgFltr "iddig" Filter Enable 1'b0: No filter 1'b1: Filter
19:16	RO	0x0	NumCtlEps Number of Device Mode Control Endpoints in Addition to Endpoint Range: 0-15
15:14	RO	0x0	PhyDataWidth UTMI+ PHY/ULPI-to-Internal UTMI+ Wrapper Data Width When a ULPI PHY is used, an internal wrapper converts ULPI to UTMI+. 2'b00: 8 bits 2'b01: 16 bits 2'b10: 8/16 bits, software selectable Others: Reserved
13:7	RO	0x0	reserved
6	RO	0x0	EnHiber Enable Hibernation 1'b0: Hibernation feature not enabled 1'b1: Hibernation feature enabled
5	RO	0x0	AhbFreq Minimum AHB Frequency Less Than 60 MHz 1'b0: No 1'b1: Yes
4	RO	0x0	EnParPwrDown Enable Partial Power Down 1'b0: Partial Power Down Not Enabled 1'b1: Partial Power Down Enabled
3:0	RO	0x0	NumDevPerioEps Number of Device Mode Periodic IN Endpoints Range: 0-15

USBOTG_GLPMCFG

Address: Operational Base + offset (0x0054)

Core LPM Configuration Register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31	RW	0x0	<p>InvSelHsic HSIC-Invert Select HSIC</p> <p>The application uses this bit to control the Otg core HSIC enable/disable. This bit overrides and functionally inverts the if_sel_hsic input port signal. If the core operates as non-HSIC-capable, it can only connect to non-HSIC-capable PHYs. If the core operates as HSIC-capable, it can only connect to HSIC capable PHYs. If the if_sel_hsic input signal is1:</p> <ul style="list-style-type: none"> 1'b1: HSIC capability is not enabled. 1'b0: HSIC capability is enabled, <p>If InvSelHsic = 1'b0: HSIC capability is enabled. If the if_sel_hsic input signal is 0:</p> <ul style="list-style-type: none"> 1'b1: HSIC capability is enabled, 1'b0: HSIC capability is not enabled. <p>This bit is writable only if an HSIC mode was specified for Mode of Operation (parameter OTG_ENABLE_HSIC). This bit is valid only if OTG_ENABLE_HSIC is enabled.</p>
30	RW	0x0	<p>HSICCon HSIC-Connect</p> <p>The application must use this bit to initiate the HSIC Attach sequence. Host Mode: Once this bit is set, the host core configures to drive the HSIC Idle state (STROBE = 1 & DATA = 0) on the bus. It then waits for the device to initiate the Connect sequence. Device Mode: Once this bit is set, the device core waits for the HSIC Idle line state on the bus. Upon receiving the Idle line state, it initiates the HSIC Connect sequence. This bit is valid only if OTG_ENABLE_HSIC is 1, if_sel_hsic = 1 and InvSelHSIC is 0. Otherwise, it is read-only.</p>
29:28	RO	0x0	reserved
27:25	RO	0x0	<p>LPM_RetryCnt_Sts LPM Retry Count Status</p> <p>Number of LPM host retries remaining to be transmitted for the current LPM sequence.</p>
24	RW	0x0	<p>SndLPM Send LPM Transaction</p> <p>Host Mode: When the application software sets this bit, an LPM transaction containing two tokens, EXT and LPM, is sent. The hardware clears this bit once a valid response (STALL, NYET, or ACK) is received from the device or the core has finished transmitting the programmed number of LPM retries. Note: This bit must only be set when the host is connected to a local port.</p>

Bit	Attr	Reset Value	Description
23:21	RWSC	0x0	LPM_Retry_Cnt LPM Retry Count When the device gives an ERROR response, this is the number of additional LPM retries that the host performs until a valid device response (STALL, NYET, or ACK) is received.
20:17	RW	0x0	LPM_Chnl_Indx LPM Channel Index The channel number on which the LPM transaction must be applied while sending an LPM transaction to the local device. Based on the LPM channel index, the core automatically inserts the device address and endpoint number programmed in the corresponding channel into the LPM transaction.
16	RO	0x0	L1ResumeOK Sleep State Resume OK Indicates that the application or host can start a resume from the Sleep state. This bit is valid in the LPM Sleep (L1) state. It is set in Sleep mode after a delay of 50 us (TL1Residency). The bit is reset when SlpSts is 0 1'b1: The application/core can start resume from the Sleep state 1'b0: The application/core cannot start resume from the Sleep state

Bit	Attr	Reset Value	Description
15	RO	0x0	<p>SlpSts Port Sleep Status Device Mode: This bit is set as long as a Sleep condition is present on the USB bus. The core enters the Sleep state when an ACK response is sent to an LPM transaction and the timer TL1TokenRetry. has expired. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the PHY Suspend input pin. The application must rely on SlpSts and not ACK in CoreL1Res to confirm transition into sleep.</p> <p>The core comes out of sleep: When there is any activity on the USB line_state When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig) or when the application resets or soft-disconnects the device.</p> <p>Host Mode: The host transitions to the Sleep (L1) state as a sideeffect of a successful LPM transaction by the core to the local port with an ACK response from the device. The read value of this bit reflects the port's current sleep status. The core clears this bit after: The core detects a remote L1 Wakeup signal The application sets the Port Reset bit or the Port L1Resume bit in the HPRT register or The application sets the L1Resume/ Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.L1WkUpInt or GINTSTS.DisconnInt, respectively).</p> <p>Values: 1'b0: Core not in L1 1'b1: Core in L1</p>
14:13	RO	0x0	<p>CoreL1Res LPM Response Device Mode: The core's response to the received LPM transaction is reflected in these two bits. Host Mode: The handshake response received from the local device for LPM transaction.</p> <p>2'b11: ACK 2'b10: NYET 2'b01: STALL 2'b00: ERROR (No handshake response)</p>

Bit	Attr	Reset Value	Description																																																
12:8	RW	0x00	<p>HIRD_Thres HIRD Threshold Device Mode: The core asserts L1SuspendM to put the PHY into Deep Low-Power mode in L1 when the HIRD value is greater than or equal to the value defined in this field (GLPMCFG.HIRD_Thres[3:0]), and HIRD_Thres[4] is set to 1'b1. Host Mode: The core asserts L1SuspendM to put the PHY into Deep Low-Power mode in L1 when HIRD_Thres[4] is set to 1'b1. HIRD_Thres[3:0] specifies the time for which resume signaling is to be reflected by the host TL1HubDrvResume2) on the USB when it detects device-initiated resume. HIRD_Thres must not be programmed with a value greater than 4'b1100 in Host mode, because this exceeds maximum $T_{L1HubDrvResume2}$. No HIRD_Thres[3:0] resume time(us)</p> <table> <tbody> <tr><td>1</td><td>4'b0000</td><td>60</td></tr> <tr><td>2</td><td>4'b0001</td><td>135</td></tr> <tr><td>3</td><td>4'b0010</td><td>210</td></tr> <tr><td>4</td><td>4'b0011</td><td>285</td></tr> <tr><td>5</td><td>4'b0100</td><td>360</td></tr> <tr><td>6</td><td>4'b0101</td><td>435</td></tr> <tr><td>7</td><td>4'b0110</td><td>510</td></tr> <tr><td>8</td><td>4'b0111</td><td>585</td></tr> <tr><td>9</td><td>4'b1000</td><td>660</td></tr> <tr><td>10</td><td>4'b1001</td><td>735</td></tr> <tr><td>11</td><td>4'b1010</td><td>810</td></tr> <tr><td>12</td><td>4'b1011</td><td>885</td></tr> <tr><td>13</td><td>4'b1100</td><td>960</td></tr> <tr><td>14</td><td>4'b1101</td><td>invalid</td></tr> <tr><td>15</td><td>4'b1110</td><td>invalid</td></tr> <tr><td>16</td><td>4'b1111</td><td>invalid</td></tr> </tbody> </table>	1	4'b0000	60	2	4'b0001	135	3	4'b0010	210	4	4'b0011	285	5	4'b0100	360	6	4'b0101	435	7	4'b0110	510	8	4'b0111	585	9	4'b1000	660	10	4'b1001	735	11	4'b1010	810	12	4'b1011	885	13	4'b1100	960	14	4'b1101	invalid	15	4'b1110	invalid	16	4'b1111	invalid
1	4'b0000	60																																																	
2	4'b0001	135																																																	
3	4'b0010	210																																																	
4	4'b0011	285																																																	
5	4'b0100	360																																																	
6	4'b0101	435																																																	
7	4'b0110	510																																																	
8	4'b0111	585																																																	
9	4'b1000	660																																																	
10	4'b1001	735																																																	
11	4'b1010	810																																																	
12	4'b1011	885																																																	
13	4'b1100	960																																																	
14	4'b1101	invalid																																																	
15	4'b1110	invalid																																																	
16	4'b1111	invalid																																																	

Bit	Attr	Reset Value	Description
7	RW	0x0	<p>EnbISlpM Enable utmi_sleep_n For ULPI interface: The application uses this bit to write to the function control [7] in the L1 state, to enable the PHY to go into Low Power mode. For the host, this bit is valid only in Local Device mode.</p> <p>1'b0: Writes to the ULPI Function Control Bit [7] are disabled. 1'b1: The core is enabled to write to the ULPI Function Control Bit [7], which enables the PHY to enter Low-Power mode.</p> <p>Note: When a ULPI interface is configured, enabling this bit results in a write to Bit 7 of the ULPI Function Control register. The ULPI PHY supports writing to this bit, and in the L1 state asserts SleepM when utmi_l1_suspend_n cannot be asserted.</p> <p>When a ULPI interface is configured, this bit must always be set if you are using the ULPI PHY. Note: For ULPI interface, do not clear this bit during the resume. For all other interfaces: The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. For the host, this bit is valid only in Local Device mode.</p> <p>1'b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. 1'b1: utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted.</p>
6	RW	0x0	<p>bRemoteWake RemoteWakeEnable Host Mode: The remote wakeup value to be sent in the LPM transaction's wIndex field. Device Mode: This field is updated with the received bRemoteWake LPM token's bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction.</p>

Bit	Attr	Reset Value	Description																																																			
5:2	RW	0x0	<p>HIRD Host-Initiated Resume Duration Host Mode: The value of HIRD to be sent in an LPM transaction. This value is also used to initiate resume for a duration $T_{L1HubDrvResume1}$ for host initiated resume. Device Mode: This field is updated with the Received LPM Token HIRD bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction</p> <table> <thead> <tr> <th>SI. No</th> <th>HIRD[3:0]</th> <th>THIRD (us)</th> </tr> </thead> <tbody> <tr><td>1</td><td>4'b0000</td><td>50</td></tr> <tr><td>2</td><td>4'b0001</td><td>125</td></tr> <tr><td>3</td><td>4'b0010</td><td>200</td></tr> <tr><td>4</td><td>4'b0011</td><td>275</td></tr> <tr><td>5</td><td>4'b0100</td><td>350</td></tr> <tr><td>6</td><td>4'b0101</td><td>425</td></tr> <tr><td>7</td><td>4'b0110</td><td>500</td></tr> <tr><td>8</td><td>4'b0111</td><td>575</td></tr> <tr><td>9</td><td>4'b1000</td><td>650</td></tr> <tr><td>10</td><td>4'b1001</td><td>725</td></tr> <tr><td>11</td><td>4'b1010</td><td>800</td></tr> <tr><td>12</td><td>4'b1011</td><td>875</td></tr> <tr><td>13</td><td>4'b1100</td><td>950</td></tr> <tr><td>14</td><td>4'b1101</td><td>1025</td></tr> <tr><td>15</td><td>4'b1110</td><td>1100</td></tr> <tr><td>16</td><td>4'b1111</td><td>1175</td></tr> </tbody> </table>	SI. No	HIRD[3:0]	THIRD (us)	1	4'b0000	50	2	4'b0001	125	3	4'b0010	200	4	4'b0011	275	5	4'b0100	350	6	4'b0101	425	7	4'b0110	500	8	4'b0111	575	9	4'b1000	650	10	4'b1001	725	11	4'b1010	800	12	4'b1011	875	13	4'b1100	950	14	4'b1101	1025	15	4'b1110	1100	16	4'b1111	1175
SI. No	HIRD[3:0]	THIRD (us)																																																				
1	4'b0000	50																																																				
2	4'b0001	125																																																				
3	4'b0010	200																																																				
4	4'b0011	275																																																				
5	4'b0100	350																																																				
6	4'b0101	425																																																				
7	4'b0110	500																																																				
8	4'b0111	575																																																				
9	4'b1000	650																																																				
10	4'b1001	725																																																				
11	4'b1010	800																																																				
12	4'b1011	875																																																				
13	4'b1100	950																																																				
14	4'b1101	1025																																																				
15	4'b1110	1100																																																				
16	4'b1111	1175																																																				
1	RW	0x0	<p>AppL1Res LPM response programmed by application Handshake response to LPM token pre-programmed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1'b0, the core always responds with a NYET. If GLPMCFG.LPMCap is 1'b1, the core responds as follows: 1'b1: ACK. Even though an ACK is pre-programmed, the core responds with an ACK only on a successful LPM transaction. The LPM transaction is successful if: There are no PID/CRC5 errors in both the EXT token and the LPM token (else ERROR); A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL); No data is pending in the Transmit queue (else NYET) 1'b0: NYET. The pre-programmed software bit is overridden for response to LPM token when: (1) The received bLinkState is not L1 (STALL response); (2) An error is detected in either of the LPM token packets due to corruption (ERROR response).</p>																																																			

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>LPMCap LPM-Capable</p> <p>The application uses this bit to control the OTG core LPM capabilities. If the core operates as a non-LPM-capable host, it cannot request the connected device/hub to activate LPM mode. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions.</p> <p>1'b0: LPM capability is not enabled. 1'b1: LPM capability is enabled.</p> <p>This bit is writable only if an LPM mode was specified for Mode of Operation (parameter OTG_ENABLE_LPM). Otherwise, reads return 0.</p>

USBOTG_GPWRDN

Address: Operational Base + offset (0x0058)

Global Power Down Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:24	RO	0x00	<p>MultValIdBC Multi Valued ID pin Battery Charger ACA inputs in the following order: Bit 26 - rid_float. Bit 25 - rid_gnd Bit 24 - rid_a Bit 23 - rid_b Bit 22 - rid_c</p> <p>These bits are present only if OTG_BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0.</p>
23	W1C	0x0	<p>ADPInt ADP Interrupt This bit is set whenever there is a ADP event.</p>
22	RO	0x0	<p>BSessVld B Session Valid This field reflects the B session valid status signal from the PHY. 1'b0: B-Valid is 0. 1'b1: B-Valid is 1. This bit is valid only when GPWRDN.PMUActv is 1.</p>

Bit	Attr	Reset Value	Description
21	RO	0x0	<p>IDDIG</p> <p>This bit indicates the status of the signal IDDIG. The application must read this bit after receiving GPWRDN.StsChngInt and decode based on the previous value stored by the application. Indicates the current mode.</p> <p>1'b1: Device mode 1'b0: Host mode</p> <p>This bit is valid only when GPWRDN.PMUActv is 1.</p>
20:19	RO	0x0	<p>LineState</p> <p>This field indicates the current linestate on USB as seen by the PMU module.</p> <p>2'b00: DM = 0, DP = 0. 2'b01: DM = 0, DP = 1. 2'b10: DM = 1, DP = 0. 2'b11: Not-defined.</p> <p>This bit is valid only when GPWRDN.PMUActv is 1.</p>
18	RW	0x0	<p>StsChngIntMsk</p> <p>Mask For StsChng Interrupt</p>
17	W1C	0x0	<p>StsChngInt</p> <p>This field indicates a status change in either the IDDIG or BSessVld signal.</p> <p>1'b0: No Status change 1'b1: status change detected</p> <p>After receiving this interrupt the application should read the GPWRDN register and interpret the change in IDDIG or BSesVld with respect to the previous value stored by the application.</p>
16	RW	0x0	<p>SRPDetectMsk</p> <p>Mask For SRPDetect Interrupt</p>
15	W1C	0x0	<p>SRPDetect</p> <p>This field indicates that SRP has been detected by the PMU. This field generates an interrupt.</p> <p>After detecting SRP during hibernation the application should not restore the core. The application should get into the initialization process.</p> <p>1'b0: SRP not detected 1'b1: SRP detected</p>
14	RW	0x0	<p>ConnDetMsk</p> <p>Mask for ConnectDet interrupt</p> <p>This bit is valid only when OTG_EN_PWRLOPT = 2.</p>
13	W1C	0x0	<p>ConnectDet</p> <p>This field indicates that a new connect has been detected</p> <p>1'b0: Connect not detected 1'b1: Connect detected</p> <p>This bit is valid only when OTG_EN_PWRLOPT = 2.</p>

Bit	Attr	Reset Value	Description
12	RW	0x0	DisconnectDetectMsk Mask For DisconnectDetect Interrupt This bit is valid only when OTG_EN_PWRROPT = 2.
11	W1C	0x0	DisconnectDetect This field indicates that Disconnect has been detected by the PMU. This field generates an interrupt. After detecting disconnect during hibernation the application must not restore the core, but instead start the initialization process. 1'b0: Disconnect not detected 1'b1: Disconnect detected This bit is valid only when OTG_EN_PWRROPT = 2.
10	RW	0x0	ResetDetMsk Mask For ResetDetected interrupt. This bit is valid only when OTG_EN_PWRROPT = 2.
9	W1C	0x0	ResetDetected This field indicates that Reset has been detected by the PMU module. This field generates an interrupt. 1'b0: Reset Not Detected 1'b1: Reset Detected This bit is valid only when OTG_EN_PWRROPT = 2.
8	RW	0x0	LineStageChangeMsk Mask For LineStateChange interrupt This bit is valid only when OTG_EN_PWRROPT = 2.
7	W1C	0x0	LnStsChng Line State Change This interrupt is asserted when there is a Linestate Change detected by the PMU. The application should read GPWRDN.Linestate to determine the current linestate on USB. 1'b0: No LineState change on USB 1'b1: LineState change on USB This bit is valid only when GPWRDN.PMUActv is 1. This bit is valid only when OTG_EN_PWRROPT = 2.
6	RW	0x0	DisableVBUS The application should program this bit if HPRT0.PrtPwr was programmed to 0 before entering Hibernation. This is to indicate PMU whether session was ended before entering Hibernation. 1'b0: HPRT0.PrtPwr was not programmed to 0. 1'b1: HPRT0.PrtPwr was programmed to 0.

Bit	Attr	Reset Value	Description
5	RW	0x0	<p>PwrDnSwtch Power Down Switch This bit indicates to the OTG core VDD switch is in ON/OFF state 1'b0: OTG is in ON state 1'b1: OTG is in OFF state Note: This bit must not be written to during normal mode of operation.</p>
4	RW	0x0	<p>PwrDnRst_n Power Down ResetN The application must program this bit to reset the DWC OTG core during the Hibernation exit process or during ADP when powering up the core (in case the DWC OTG core was powered off during ADP process). 1'b1: OTG is in normal operation 1'b0: reset OTG Note: This bit must not be written to during normal mode of operation.</p>
3	RW	0x0	<p>PwrDnClmp Power Down Clamp The application must program this bit to enable or disable the clamps to all the outputs of the DWC OTG core module to prevent the corruption of other active logic. 1'b0: Disable PMU power clamp 1'b1: Enable PMU power clamp</p>
2	RW	0x0	<p>Restore The application should program this bit to enable or disable restore mode from the PMU module. 1'b0: OTG in normal mode of operation 1'b1: OTG in restore mode Note: This bit must not be written to during normal mode of operation. This bit is valid only when OTG_EN_PWR0PT = 2.</p>
1	RW	0x0	<p>PMUActv PMU Active This is bit is to enable or disable the PMU logic. 1'b0: Disable PMU module 1'b1: Enable PMU module Note: This bit must not be written to during normal mode of operation.</p>

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>PMUIntSel PMU Interrupt Select When the hibernation functionality is selected using the configuration option OTG_EN_PWR_OPT = 2, a write to this bit with 1'b1 enables the PMU to generate interrupts to the application. During this state all interrupts from the core module are blocked to the application. Note: This bit must be set to 1'b1 before the core is put into hibernation 1'b0: Internal OTG_core interrupt is selected 1'b1: the external OTG_pmu interrupt is selected Note: This bit must not be written to during normal mode of operation.</p>

USBOTG_GDFIFO CFG

Address: Operational Base + offset (0x005c)

Global DFIFO Software Config Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>EPIInfoBaseAddr This field provides the start address of the EP info controller.</p>
15:0	RW	0x0000	<p>GDFIFO Cfg This field is for dynamic programming of the DFIFO Size. This value takes effect only when the application programs a non-zero value to this register. The core does not have any corrective logic if the FIFO sizes are programmed incorrectly.</p>

USBOTG_GADPCTL

Address: Operational Base + offset (0x0060)

ADP Timer, Control and Status Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	R/WSC	0x0	<p>AR Access Request 2'b00 Read/Write Valid (updated by the core) 2'b01 Read 2'b10 Write 2'b11 Reserved</p>
26	RW	0x0	<p>AdpTmoutMsk ADP Timeout Interrupt Mask When this bit is set, it unmasks the interrupt because of AdpTmoutInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).</p>

Bit	Attr	Reset Value	Description
25	RW	0x0	AdpSnsIntMsk ADP Sense Interrupt Mask When this bit is set, it unmasks the interrupt due to AdpSnsInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).
24	RW	0x0	AdpPrbIntMsk ADP Probe Interrupt Mask When this bit is set, it unmasks the interrupt due to AdpPrbInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).
23	W1C	0x0	AdpTmoutInt ADP Timeout Interrupt This bit is relevant only for an ADP probe. When this bit is set, it means that the ramp time has completed (GADPCTL.RTIM has reached its terminal value of 0x7FF). This is a debug feature that allows software to read the ramp time after each cycle. This bit is valid only if OTG_Ver = 1'b1.
22	W1C	0x0	AdpSnsInt ADP Sense Interrupt When this bit is set, it means that the VBUS voltage is greater than VadpSns value or VadpSns is reached. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
21	W1C	0x0	AdpPrbInt ADP Probe Interrupt When this bit is set, it means that the VBUS voltage is greater than VadpPrb or VadpPrb is reached. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
20	RW	0x0	ADPEN ADP Enable When set, the core performs either ADP probing or sensing based on EnaPrb or EnaSns. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
19	RWSC	0x0	ADPRes ADP Reset When set, ADP controller is reset. This bit is auto-cleared after the reset procedure is complete in ADP controller. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
18	RW	0x0	EnaSns Enable Sense When programmed to 1'b1, the core performs a sense operation. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).

Bit	Attr	Reset Value	Description
17	RW	0x0	<p>EnaPrb Enable Probe</p> <p>When programmed to 1'b1, the core performs a probe operation. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).</p>
16:6	RO	0x000	<p>RTIM RAMP TIME</p> <p>These bits capture the latest time it took for VBUS to ramp from VADP_SINK to VADP_PRB. The bits are defined in units of 32 kHz clock cycles as follows:</p> <ul style="list-style-type: none"> 3'b000 - 1 cycles 3'b001 - 2 cycles 3'b010 - 3 cycles and so on till 0x7FF - 2048 cycles <p>A time of 1024 cycles at 32 kHz corresponds to a time of 32 msec.</p> <p>(Note for scaledown ramp_timeout = prb_delta == 2'b00 => 200 cycles prb_delta == 2'b01 => 100 cycles prb_delta == 2'b01 => 50 cycles prb_delta == 2'b01 => 25 cycles.)</p>
5:4	RW	0x0	<p>PrbPer Probe Period</p> <p>These bits sets the TadpPrd as follows:</p> <ul style="list-style-type: none"> 2'b00 - 0.625 to 0.925 sec (typical 0.775 sec) 2'b01 - 1.25 to 1.85 sec (typical 1.55 sec) 2'b10 - 1.9 to 2.6 sec (typical 2.275 sec) 2'b11 - Reserved <p>(PRB_PER is also scaledown prb_per== 2'b00 => 400 ADP clocks prb_per== 2'b01 => 600 ADP clocks prb_per== 2'b10 => 800 ADP clocks prb_per==2'b11 => 1000 ADP clocks)</p>
3:2	RW	0x0	<p>PrbDelta Probe Delta</p> <p>These bits set the resolution for RTIM value. The bits are defined in units of 32 kHz clock cycles as follows:</p> <ul style="list-style-type: none"> 2'b00 - 1 cycles 2'b01 - 2 cycles 2'b10 - 3 cycles 2'b11 - 4 cycles <p>For example if this value is chosen to 2'b01, it means that RTIM increments for every three 32Khz clock cycles.</p>

Bit	Attr	Reset Value	Description
1:0	RW	0x0	<p>PrbDschg Probe Discharge These bits set the times for TadpDschg. These bits are defined as follows:</p> <ul style="list-style-type: none"> 2'b00 4 msec (Scaledown 2 32Khz clock cycles) 2'b01 8 msec (Scaledown 4 32Khz clock cycles) 2'b10 16 msec (Scaledown 8 32Khz clock cycles) 2'b11 32 msec (Scaledown 16 32Khz clock cycles)

USBOTG_HPTXFSIZ

Address: Operational Base + offset (0x0100)

Host Periodic Transmit FIFO Size Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>PTxFSize Host Periodic Tx FIFO Depth This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest Host Mode Periodic Tx Data FIFO Depth (parameter OTG_TX_HPERIO_DFIFO_DEPTH). If Enable Dynamic FIFO Sizing? Was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the power-on value set .</p>

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>PTxFStAddr Host Periodic TxFIFO Start Address The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth and Largest Non-periodic Tx Data FIFO Depth specified.</p> <p>These parameters are: In shared FIFO operation: OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH. In dedicated FIFO mode: OTG_RX_DFIFO_DEPTH + OTG_TX_HNPERIO_DFIFO_DEPTH. If Enable Dynamic FIFO Sizing? was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the power-on value.</p>

USBOTG_DIEPTXFn

Address: Operational Base + offset (0x0104+0x4*(n-1)), n = 1 - 15

Device Periodic Transmit FIFO-n Size Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>INEP1TxDep IN Endpoint TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth (parameter OTG_TX_DINEP_DFIFO_DEPTH_n)(0 < n <= 15). If Enable Dynamic FIFO Sizing? was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the Power-on value. If Enable Dynamic FIFO Sizing? was selected (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the Power-on value .</p>

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>INEP1TxFStAddr IN Endpoint FIFO1 Transmit RAM Start Address This field contains the memory start address for IN endpoint Transmit FIFOOn (0 < n <= 15). The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth (parameter OTG_RX_DFIFO_DEPTH). OTG_RX_DFIFO_DEPTH + SUM 0 to n-1 (OTG_DINEP_TXFIFO_DEPTH_n) For example start address of IN endpoint FIFO 1 is OTG_RX_DFIFO_DEPTH + OTG_DINEP_TXFIFO_DEPTH_0. The start address of IN endpoint FIFO 2 is OTG_RX_DFIFO_DEPTH + OTG_DINEP_TXFIFO_DEPTH_0 + OTG_DINEP_TXFIFO_DEPTH_1. If Enable Dynamic FIFO Sizing? was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected (parameter OTG_DFIFO_DYNAMIC = 1), and you have programmed a new value for Rx FIFO depth, you can write that value in this field. Programmed values must not exceed the power-on value set .</p>

USBOTG_HCFG

Address: Operational Base + offset (0x0400)

Host Configuration Register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	<p>PerSchedEna Enable Periodic Scheduling Applicable in Scatter/Gather DMA mode only. Enables periodic scheduling within the core. Initially, the bit is reset. The core will not process any periodic channels. As soon as this bit is set, the core will get ready to start scheduling periodic channels and sets HCFG.PerSchedStat. The setting of HCFG.PerSchedStat indicates the core has enabled periodic scheduling. Once HCFG.PerSchedEna is set, the application is not supposed to again reset the bit unless HCFG.PerSchedStat is set. As soon as this bit is reset, the core will get ready to stop scheduling periodic channels and resets HCFG.PerSchedStat. In non-Scatter/Gather DMA mode, this bit is reserved.</p>

Bit	Attr	Reset Value	Description
25:24	RW	0x0	<p>FrListEn Frame List Entries The value in the register specifies the number of entries in the Frame list. This field is valid only in Scatter/Gather DMA mode.</p>
23	RW	0x0	<p>DescDMA Enable Scatter/gather DMA in Host mode When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: GAHBCFG.DMAEn=0, HCFG.DescDMA=0 => Slave mode GAHBCFG.DMAEn=0, HCFG.DescDMA=1 => Invalid GAHBCFG.DMAEn=1, HCFG.DescDMA=0 => Buffered DMA mode GAHBCFG.DMAEn=1, HCFG.DescDMA=1 => Scatter/Gather DMA mode In non-Scatter/Gather DMA mode, this bit is reserved.</p>
22:16	RO	0x0	reserved
15:8	RW	0x00	<p>ResValid Resume Validation Period This field is effective only when HCFG.Ena32KHzS is set. It controls the resume period when the core resumes from suspend. The core counts the ResValid number of clock cycles to detect a valid resume when this is set.</p>
7	RW	0x0	<p>Ena32KHzS Enable 32-KHz Suspend Mode This bit can only be set if the USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero. When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend.</p>
6:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	<p>FSLSSupp FS- and LS-Only Support The application uses this bit to control the core enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming.</p> <p>1'b0: HS/FS/LS, based on the maximum speed supported by the connected device 1'b1: FS/LS-only, even if the connected device can support HS</p>
1:0	RW	0x0	<p>FSLSPclkSel FS/LS PHY Clock Select 2'b00: PHY clock is running at 30/60 MHz 2'b01: PHY clock is running at 48 MHz Others: Reserved</p>

USBOTG_HFIR

Address: Operational Base + offset (0x0404)

Host Frame Interval Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	<p>FrInt Frame Interval The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro-SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for a FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration.</p> <p>125 us * (PHY clock frequency for HS) 1 ms * (PHY clock frequency for FS/LS)</p>

USBOTG_HFNUM

Address: Operational Base + offset (0x0408)

Host Frame Number/Frame Time Remaining Register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	FrRem Frame Time Remaining Indicates the amount of time remaining in the current micro-frame (HS) or frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB.
15:0	RO	0xffff	FrNum Frame Number This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. This field is writable only if Remove Optional Features? was not selected (OTG_RM_OTG_FEATURES = 0). Otherwise, reads return the frame number value.

USBOTG_HPTXSTS

Address: Operational Base + offset (0x0410)

Host Periodic Transmit FIFO/Queue Status Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	PTxQTop Top of the Periodic Transmit Request Queue This indicates the entry in the Periodic Tx Request Queue that is currently being processed by the MAC. This register is used for debugging. Bit [31]: Odd/Even (micro)frame 1'b0: send in even (micro)frame 1'b1: send in odd (micro)frame Bits [30:27]: Channel/endpoint number Bits [26:25]: Type 2'b00: IN/OUT 2'b01: Zero-length packet 2'b10: CSPLIT 2'b11: Disable channel command Bit [24]: Terminate (last entry for the selected channel/endpoint)

Bit	Attr	Reset Value	Description
23:16	RO	0x00	<p>PTxQSpAvail Periodic Transmit Request Queue Space Available Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests.</p> <ul style="list-style-type: none"> 8'h0: Periodic Transmit Request Queue is full 8'h1: 1 location available 8'h2: 2 locations available n: n locations available (0 <=n <= 16) Others: Reserved
15:0	RW	0x0000	<p>PTxFSpAvail Periodic Transmit Data FIFO Space Available Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words</p> <ul style="list-style-type: none"> . 16'h0: Periodic TxFIFO is full . 16'h1: 1 word available . 16'h2: 2 words available . 16'hn: n words available (where 0 . n . 32,768) . 16'h8000: 32,768 words available . Others: Reserved

USBOTG_HAIT

Address: Operational Base + offset (0x0414)

Host All Channels Interrupt Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	<p>HAI NT Channel Interrupts One bit per channel: Bit 0 for Channel 0 Bit 1 for Channel 1 Bit 15 for Channel 15</p>

USBOTG_HINTMSK

Address: Operational Base + offset (0x0418)

Host All Channels Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	<p>HAI NTMs k Channel Interrupt Mask One bit per channel: Bit 0 for channel 0, bit 15 for channel 15</p>

USBOTG_HPRT

Address: Operational Base + offset (0x0440)

Host Port Control and Status Register

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:17	RO	0x0	PrtSpd Port Speed Indicates the speed of the device attached to this port. 2'b00: High speed 2'b01: Full speed 2'b10: Low speed 2'b11: Reserved
16:13	RW	0x0	PrtTstCtl Port Test Control The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port. 4'b0000: Test mode disabled 4'b0001: Test_J mode 4'b0010: Test_K mode 4'b0011: Test_SE0_NAK mode 4'b0100: Test_Packet mode 4'b0101: Test_Force_Enable Others: Reserved
12	RWSC	0x0	PrtPwr Port Power The application uses this field to control power to this port (write 1'b1 to set to 1'b1 and write 1'b0 to set to 1'b0), and the core can clear this bit on an over current condition. 1'b0: Power off 1'b1: Power on
11:10	RO	0x0	PrtLnSts Port Line Status Indicates the current logic level USB data lines Bit [10]: Logic level of D+ Bit [11]: Logic level of D
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	<p>PrtRst Port Reset When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete.</p> <p>1'b0: Port not in reset 1'b1: Port in reset</p> <p>To start a reset on the port, the application must leave this bit set for at least the minimum duration mentioned below, as specified in the USB 2.0 specification, Section 7.1.7.5. The application can leave it set for another 10 ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard.</p> <p>High speed: 50 ms Full speed/Low speed: 10 ms</p>
7	RWSC	0x0	<p>PrtSusp Port Suspend The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the suspended input pin of the PHY.</p> <p>The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.WkUpInt or GINTSTS.DisconnInt, respectively).</p> <p>1'b0: Port not in Suspend mode 1'b1: Port in Suspend mode</p>

Bit	Attr	Reset Value	Description
6	RWSC	0x0	<p>PrtRes Port Resume The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.WkUpInt), the core starts driving resume signaling without application intervention and clears this bit when it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling.</p> <p>1'b0: No resume driven 1'b1: Resume driven When LPM is enabled and the core is in the L1 (Sleep) state, setting this bit results in the following behavior: The core continues to drive the resume signal until a pre-determined time specified in the GLPMCFG.HIRD_Thres[3:0] field. If the core detects a USB remote wakeup sequence, as indicated by the Port L1 Resume/Remote L1 Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.L1WkUpInt), the core starts driving resume signaling without application intervention and clears this bit at the end of the resume. The read value of this bit indicates whether the core is currently driving resume signaling.</p> <p>1'b0: No resume driven 1'b1: Resume driven</p>
5	W1C	0x0	<p>PrtOvrCurrChng Port Overcurrent Change The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes.</p>
4	RO	0x0	<p>PrtOvrCurrAct Port Overcurrent Active Indicates the overcurrent condition of the port. 1'b0: No overcurrent condition 1'b1: Overcurrent condition</p>
3	W1C	0x0	<p>PrtEnChng Port Enable/Disable Change The core sets this bit when the status of the Port Enable bit [2] of this register changes.</p>

Bit	Attr	Reset Value	Description
2	W1C	0x0	<p>PrtEna Port Enable A port is enabled only by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit. The application cannot set this bit by a register write. It can only clear it to disable the port. This bit does not trigger any interrupt to the application.</p> <p>1'b0: Port disabled 1'b1: Port enabled</p>
1	W1C	0x0	<p>PrtConnDet Port Connect Detected The core sets this bit when a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). The application must write a 1 to this bit to clear the interrupt.</p>
0	RO	0x0	<p>PrtConnSts Port Connect Status 1'b0: No device is attached to the port. 1'b1: A device is attached to the port.</p>

USBOTG_HCCCHARn

Address: Operational Base + offset (0x0500)

Host Channel-n Characteristics Register

Bit	Attr	Reset Value	Description
31	RWSC	0x0	<p>ChEna Channel Enable When Scatter/Gather mode is enabled 1'b0: Indicates that the descriptor structure is not yet ready. 1'b1: Indicates that the descriptor structure and data buffer with data is setup and this channel can access the descriptor. When Scatter/Gather mode is disabled, This field is set by the application and cleared by the OTG host. 1'b0: Channel disabled 1'b1: Channel enabled</p>
30	RWSC	0x0	<p>ChDis Channel Disable The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.</p>

Bit	Attr	Reset Value	Description
29	RW	0x0	<p>OddFrm Odd Frame</p> <p>This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro) frame. This field is applicable for only periodic (isochronous and interrupt) transactions.</p> <p>1'b0: Even (micro)frame 1'b1: Odd (micro)frame</p> <p>This field is not applicable for Scatter/Gather DMA mode and need not be programmed by the application and is ignored by the core.</p>
28:22	RW	0x00	<p>DevAddr Device Address</p> <p>This field selects the specific device serving as the data source or sink.</p>
21:20	RW	0x0	<p>MC_EC Multi Count (MC) / Error Count (EC)</p> <p>When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SplitEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per micro-frame for this periodic endpoint. For non periodic transfers, this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes arbitration.</p> <p>2'b00: Reserved This field yields undefined results. 2'b01: 1 transaction 2'b10: 2 transactions to be issued for this endpoint per micro-frame 2'b11: 3 transactions to be issued for this endpoint per micro-frame</p> <p>When HCSPLTn.SplitEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 2'b01.</p>
19:18	RW	0x0	<p>EPType Endpoint Type</p> <p>Indicates the transfer type selected.</p> <p>2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt</p>
17	RW	0x0	<p>LSpdDev Low-Speed Device</p> <p>This field is set by the application to indicate that this channel is communicating to a low-speed device.</p>

Bit	Attr	Reset Value	Description
16	RO	0x0	reserved
15	RW	0x0	EPDir Endpoint Direction Indicates whether the transaction is IN or OUT. 1'b0: OUT 1'b1: IN
14:11	RW	0x0	EPNum Endpoint Number Indicates the endpoint number on the device serving as the data source or sink.
10:0	RW	0x000	MPS Maximum Packet Size Indicates the maximum packet size of the associated endpoint.

USBOTG_HCSPLTn

Address: Operational Base + offset (0x0504)

Host Channel-n Split Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	SpltEna Split Enable The application sets this field to indicate that this channel is enabled to perform split transactions.
30:17	RO	0x0	reserved
16	RW	0x0	CompSplt Do Complete Split The application sets this field to request the OTG host to perform a complete split transaction.
15:14	RW	0x0	XactPos Transaction Position This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. 2'b11: All. This is the entire data payload is of this transaction (which is less than or equal to 188 bytes). 2'b10: Begin. This is the first data payload of this transaction (which is larger than 188 bytes). 2'b00: Mid. This is the middle payload of this transaction (which is larger than 188bytes). 2'b01: End. This is the last payload of this transaction (which is larger than 188 bytes).
13:7	RW	0x00	HubAddr Hub Address This field holds the device address of the transaction translator's hub.

Bit	Attr	Reset Value	Description
6:1	RO	0x0	reserved
0	RW	0x0	PrtAddr Port Address This field is the port number of the recipient transaction translator.

USBOTG_HCINTn

Address: Operational Base + offset (0x0508)

Host Channel-n Interrupt Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	W1C	0x0	DESC_LST_ROLLIntr Descriptor rollover interrupt This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when the corresponding channel's descriptor list rolls over. For non-Scatter/Gather DMA mode, this bit is reserved.
12	W1C	0x0	XCS_XACT_ERR Excessive Transaction Error This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when 3 consecutive transaction errors occurred on the USB bus. XCS_XACT_ERR will not be generated for Isochronous channels. For non-Scatter/Gather DMA mode, this bit is reserved.
11	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process. BNA will not be generated for Isochronous channels. For non-Scatter/Gather DMA mode, this bit is reserved.
10	W1C	0x0	DataTglErr Data Toggle Error In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
9	W1C	0x0	FrmOvrn Frame Overrun In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core
8	W1C	0x0	BblErr Babble Error In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.

Bit	Attr	Reset Value	Description
7	W1C	0x0	XactErr Transaction Error Indicates one of the following errors occurred on the USB: CRC check failure, Timeout, Bit stuff error, False EOP. In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
6	WO	0x0	NYET NYET Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
5	W1C	0x0	ACK ACK Response Received/Transmitted Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
4	W1C	0x0	NAK NAK Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
3	W1C	0x0	STALL STALL Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
2	W1C	0x0	AHBErr AHB Error This is generated only in DMA mode when there is an AHB error during AHB read/write. The application can read the corresponding channel's DMA address register to get the error address.
1	W1C	0x0	ChHltd Channel Halted In non-Scatter/Gather DMA mode, it indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application or because of a completed transfer. In Scatter/Gather DMA mode, this indicates that transfer completed due to any of the following: EOL being set in descriptor, AHB error, Excessive transaction errors, In response to disable request by the application, Babble, Stall, Buffer Not Available (BNA)
0	W1C	0x0	XferCompl Transfer Completed For Scatter/Gather DMA mode, it indicates that current descriptor processing got completed with IOC bit set in its descriptor. In non-Scatter/Gather DMA mode, it indicates that Transfer completed normally without any errors.

USBOTG_HCINTMSKn

Address: Operational Base + offset (0x050c)

Host Channel-n Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	DESC_LST_ROLLIntrMsk Descriptor rollover interrupt Mask register This bit is valid only when Scatter/Gather DMA mode is enabled. In non-Scatter/Gather DMA mode, this bit is reserved.
12	RO	0x0	reserved
11	RW	0x0	BNAIntrMsk BNA (Buffer Not Available) Interrupt mask register This bit is valid only when Scatter/Gather DMA mode is enabled. In non-Scatter/Gather DMA mode, this bit is reserved.
10	RW	0x0	DataTglErrMsk Data Toggle Error Mask This bit is not applicable in Scatter/Gather DMA mode.
9	RW	0x0	FrmOvrnMsk Frame Overrun Mask This bit is not applicable in Scatter/Gather DMA mode.
8	RW	0x0	BblErrMsk Babble Error Mask This bit is not applicable in Scatter/Gather DMA mode.
7	RW	0x0	XactErrMsk Transaction Error Mask This bit is not applicable in Scatter/Gather DMA mode
6	RW	0x0	NyetMsk NYET Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
5	RW	0x0	AckMsk ACK Response Received/Transmitted Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
4	RW	0x0	NakMsk NAK Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
3	RW	0x0	StallMsk STALL Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.

Bit	Attr	Reset Value	Description
2	RW	0x0	AHBErrMsk AHB Error Mask Note: This bit is only accessible when OTG_ARCHITECTURE = 2
1	RW	0x0	ChHltedMsk Channel Halted Mask
0	RW	0x0	XferComplMsk Transfer Completed Mask This bit is valid only when Scatter/Gather DMA mode is enabled. In non-Scatter/Gather DMA mode, this bit is reserved.

USBOTG_HCTSIZn

Address: Operational Base + offset (0x0510)

Host Channel-n Transfer Size Register

Bit	Attr	Reset Value	Description
31	RW	0x0	DoPng Do Ping This bit is used only for OUT transfers. Setting this field to 1 directs the host to do PING protocol. Note: Do not set this bit for IN transfers. If this bit is set for IN transfers it disables the channel.
30:29	RW	0x0	Pid PID The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA (non-control)/SETUP (control)
28:19	RW	0x000	PktCnt Packet Count This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN). The host decrements this count on every successful transmission or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion. The width of this counter is specified as Width of Packet Counters (parameter OTG_PACKET_COUNT_WIDTH).

Bit	Attr	Reset Value	Description
18:0	RW	0x000000	XferSize Transfer Size For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has Reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic). The width of this counter is specified as Width of Transfer Size Counters (parameter OTG_TRANS_COUNT_WIDTH).

USBOTG_HCDMAN

Address: Operational Base + offset (0x0514)

Host Channel-n DMA Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction.

USBOTG_HCDMABn

Address: Operational Base + offset (0x051c)

Host Channel-n DMA Buffer Address Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HCDMABn Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

USBOTG_DCFG

Address: Operational Base + offset (0x0800)

Device Configuration Register

Bit	Attr	Reset Value	Description
31:26	RW	0x02	ResValid Resume Validation Period This field controls the period when the core resumes from a suspend. When this bit is set, the core counts for the ResValid number of clock cycles to detect a valid resume. This field is effective only when DCFG.Ena32KHzSusp is set.

Bit	Attr	Reset Value	Description
25:24	RW	0x0	<p>PerSchIntvl Periodic Scheduling Interval PerSchIntvl must be programmed only for Scatter/Gather DMA mode. Description: This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame. When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. When no periodic endpoints are active, then the internal DMA engine services nonperiodic endpoints, ignoring this field. After the specified time within a (micro) frame, the DMA switches to fetching for nonperiodic endpoints.</p> <p>2'b00: 25% of (micro) frame. 2'b01: 50% of (micro) frame. 2'b10: 75% of (micro) frame. 2'b11: Reserved.</p>
23	RW	0x0	<p>DescDMA Enable Scatter/Gather DMA in Device mode When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Slave mode GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Buffered DMA mode GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode</p>
22:18	RW	0x08	<p>EPMisCnt IN Endpoint Mismatch Count This field is valid only in shared FIFO operation. The application programs this field with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.</p>
17:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:11	RW	0x0	<p>PerFrInt Periodic Frame Interval Indicates the time within a (micro) frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro) frame is complete.</p> <p>2'b00: 80% of the (micro)frame interval 2'b01: 85% 2'b10: 90% 2'b11: 95%</p>
10:4	RW	0x00	<p>DevAddr Device Address The application must program this field after every SetAddress control command.</p>
3	RW	0x0	<p>Ena32KHzS Enable 32-KHz Suspend Mode When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend. This bit can only be set if USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero.</p>
2	RW	0x0	<p>NZStsOUTHShk Non-Zero-Length Status OUT Handshake The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage.</p> <p>1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. 1'b0: Send the received OUT packet to the application (zero-length or nonzerolength) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.</p>

Bit	Attr	Reset Value	Description
1:0	RW	0x0	<p>DevSpd Device Speed Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected.</p> <p>2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b10: Reserved 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz)</p>

USBOTG_DCTL

Address: Operational Base + offset (0x0804)

Device Control Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	<p>NakOnBble Set NAK automatically on babble The core sets NAK automatically for the endpoint on which babble is received.</p>

Bit	Attr	Reset Value	Description
15	RW	0x0	<p>IgnrFrmNum Ignore frame number for isochronous endpoints in case of Scatter/Gather DMA mode. Do NOT program IgnrFrmNum bit to 1'b1 when the core is operating in Threshold mode. Note: When Scatter/Gather DMA mode is enabled this feature is not applicable to highspeed, high-bandwidth transfers. When this bit is enabled, there must be only one packet per descriptor.</p> <p>1'b0: The core transmits the packets only in the frame number in which they are intended to be transmitted.</p> <p>1'b1: The core ignores the frame number, sending packets immediately as the packets are ready.</p> <p>Scatter/Gather:</p> <p>In Scatter/Gather DMA mode, when this bit is enabled, the packets are not flushed when an ISOC IN token is received for an elapsed frame.</p> <p>When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple (micro) frames.</p> <p>1'b0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame</p> <p>1'b1: Periodic transfer interrupt feature is enabled; the application can program transfers for multiple (micro)frames for periodic endpoints.</p> <p>In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro) frames are completed.</p>
14:13	RW	0x1	<p>GMC Global Multi Count GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that endpoint before moving to the next endpoint. It is only for nonperiodic endpoints.</p> <p>2'b00: Invalid. 2'b01: 1 packet. 2'b10: 2 packets. 2'b11: 3 packets.</p> <p>When Scatter/Gather DMA mode is disabled, this field is reserved. and reads 2'b00.</p>
12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	PWROnPrgDone Power-On Programming Done The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode.
10	WO	0x0	CGOUTNak Clear Global OUT NAK A write to this field clears the Global OUT NAK.
9	WO	0x0	SGOUTNak Set Global OUT NAK A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.
8	WO	0x0	CGNPInNak Clear Global Non-periodic IN NAK A write to this field clears the Global Non-periodic IN NAK.
7	WO	0x0	SGNPInNak Set Global Non-periodic IN NAK A write to this field sets the Global Non-periodic IN NAK. The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared.
6:4	RW	0x0	TstCtl Test Control 3'b000: Test mode disabled 3'b001: Test_J mode 3'b010: Test_K mode 3'b011: Test_SE0_NAK mode 3'b100: Test_Packet mode 3'b101: Test_Force_Enable Others: Reserved
3	RO	0x0	GOUTNakSts Global OUT NAK Status 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped

Bit	Attr	Reset Value	Description
2	RO	0x0	<p>GNPINNaksts Global Non-periodic IN NAK Status 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.</p>
1	RW	0x0	<p>SftDiscon Soft Disconnect The application uses this bit to signal the Otg core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit.</p> <p>1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration.</p> <p>1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host.</p>
0	RW	0x0	<p>RmtWkUpSig Remote Wakeup Signaling When the application sets this bit, the core initiates remote signaling to wake the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1-15 ms after setting it. If LPM is enabled and the core is in the L1 (Sleep) state, when the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware automatically clears this bit 50 us (TL1DevDrvResume) after being set by the application. The application must not set this bit when GLPMCFG.bRemoteWake from the previous LPM transaction is zero.</p>

USBOTG_DSTS

Address: Operational Base + offset (0x0808)

Device Status Register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:8	RW	0x0000	<p>SOFFN Frame or Micro-frame Number of the Received SOF</p> <p>When the core is operating at high speed, this field contains a micro-frame number. When the core is operating at full or low speed, this field contains a frame number.</p>
7:4	RO	0x0	reserved
3	RW	0x0	<p>ErrticErr Erratic Error</p> <p>The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the Otg core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover.</p>
2:1	RW	0x0	<p>EnumSpd Enumerated Speed</p> <p>Indicates the speed at which the OTG core has come up after speed detection through a chirp sequence.</p> <p>2'b00: High speed (PHY clock is running at 30 or 60 MHz)</p> <p>2'b01: Full speed (PHY clock is running at 30 or 60 MHz)</p> <p>2'b10: Low speed (PHY clock is running at 48 MHz, internal phy_clk at 6 MHz)</p> <p>2'b11: Full speed (PHY clock is running at 48 MHz)</p> <p>Low speed is not supported for devices using a UTMI+ PHY.</p>
0	RW	0x0	<p>SuspSts Suspend Status</p> <p>In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state when there is no activity on the utmi_lonestate signal for an extended period of time. The core comes out of the suspend: When there is any activity on the utmi_lonestate signal, When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig).</p>

USBOTG_DIEPMSK

Address: Operational Base + offset (0x0810)

Device IN Endpoint common interrupt mask register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	NAKMsk NAK interrupt Mask
12:10	RO	0x0	reserved
9	RW	0x0	BNAInIntrMsk BNA Interrupt Mask
8	RW	0x0	TxfifoUndrnMsk Fifo Underrun Mask
7	RO	0x0	reserved
6	RW	0x0	INEPNakEffMsk IN Endpoint NAK Effective Mask
5	RW	0x0	INTknEPMisMsk IN Token received with EP Mismatch Mask
4	RW	0x0	INTknTxFTEmpMsk IN Token Received When TxFIFO Empty Mask
3	RW	0x0	TimeOUTMsk Timeout Condition Mask
2	RW	0x0	AHBErrMsk AHB Error Mask
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DOEPMSK

Address: Operational Base + offset (0x0814)

Device OUT Endpoint common interrupt mask register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	NYETMsk NYET Interrupt Mask
13	RW	0x0	NAKMsk NAK Interrupt Mask
12	RW	0x0	BbleErrMsk Babble Interrupt Mask
11:10	RO	0x0	reserved
9	RW	0x0	BnaOutIntrMsk BNA interrupt Mask
8	RW	0x0	OutPktErrMsk OUT Packet Error Mask
7	RO	0x0	reserved
6	RW	0x0	Back2BackSETup Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.
5	RO	0x0	reserved
4	RW	0x0	OUTTknEPdisMsk OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only.

Bit	Attr	Reset Value	Description
3	RW	0x0	SetUPMsk SETUP Phase Done Mask Applies to control endpoints only.
2	RW	0x0	AHBErrMsk AHB Error
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DAINT

Address: Operational Base + offset (0x0818)

Device All Endpoints interrupt register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	OutEPInt OUT Endpoint Interrupt Bits One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15
15:0	RO	0x0000	InEpInt IN Endpoint Interrupt Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0, bit 15 for endpoint 15

USBOTG_DAINTMSK

Address: Operational Base + offset (0x081c)

Device All Endpoint interrupt mask register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	OutEpMsk OUT EP Interrupt Mask Bits One per OUT Endpoint: Bit 16 for OUT EP 0, bit 31 for OUT EP 15
15:0	RW	0x0000	InEpMsk IN EP Interrupt Mask Bits One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15

USBOTG_DTKNQR1

Address: Operational Base + offset (0x0820)

Device IN token sequence learning queue read register1

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	EPTkn Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 5 Bits [27:24]: Endpoint number of Token 4 Bits [15:12]: Endpoint number of Token 1 Bits [11:8]: Endpoint number of Token 0

Bit	Attr	Reset Value	Description
7	RO	0x0	WrapBit Wrap Bit This bit is set when the write pointer wraps. It is cleared when the learning queue is cleared.
6:5	RO	0x0	reserved
4:0	RO	0x00	INTknWPtr IN Token Queue Write Pointer

USBOTG_DTKNQR2

Address: Operational Base + offset (0x0824)

Device IN token sequence learning queue read register2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPTkn Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 13 Bits [27:24]: Endpoint number of Token 12 Bits [7:4]: Endpoint number of Token 7 Bits [3:0]: Endpoint number of Token 6

USBOTG_DVBUSDIS

Address: Operational Base + offset (0x0828)

Device VBUS discharge time register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0b8f	DVBUSDis Device VBUS Discharge Time Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals: VBUS discharge time in PHY clocks / 1,024. The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width). Depending on your VBUS load, this value can need adjustment.

USBOTG_DVBUSPULSE

Address: Operational Base + offset (0x082c)

Device VBUS Pulsing Timer Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	DVBUSPulse Device VBUS Pulsing Time Specifies the VBUS pulsing time during SRP. This value equals: VBUS pulsing time in PHY clocks / 1,024. The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width).

USBOTG_DTHRCTL

Address: Operational Base + offset (0x0830)

Device Threshold Control Register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x1	ArbPrkEn Arbiter Parking Enable This bit controls internal DMA arbiter parking for IN endpoints. When threshold is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into under-run conditions. By default the parking is enabled.
26	RO	0x0	reserved
25:17	RW	0x008	RxThrLen Receive Threshold Length This field specifies Receive threshold size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).
16	RW	0x0	RxThrEn Receive Threshold Enable When this bit is set, the core enables thresholding in the receive direction.
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:11	RW	0x0	<p>AHBThrRatio AHB Threshold Ratio These bits define the ratio between the AHB threshold and the MAC threshold for the transmit path only. The AHB threshold always remains less than or equal to the USB threshold, because this does not increase overhead. Both the AHB and the MAC threshold must be DWORD-aligned. The application needs to program TxThrLen and the AHBThrRatio to make the AHB Threshold value DWORD aligned. If the AHB threshold value is not DWORD aligned, the core might not behave correctly. When programming the TxThrLen and AHBThrRatio, the application must ensure that the minimum AHB threshold value does not go below 8 DWORDS to meet the USB turnaround time requirements.</p> <p>2'b00: AHB threshold = MAC threshold 2'b01: AHB threshold = MAC threshold / 2 2'b10: AHB threshold = MAC threshold / 4 2'b11: AHB threshold = MAC threshold / 8</p>
10:2	RW	0x008	<p>TxThrLen Transmit Threshold Length This field specifies Transmit threshold size in DWORDS. This field also forms the MAC threshold and specifies the amount of data, in bytes, to be in the corresponding endpoint transmit FIFO before the core can start a transaction on the USB. When the value of AHBThrRatio is 2'h00, the threshold length must be at least 8 DWORDS. If the AHBThrRatio is nonzero, the application must ensure that the AHB threshold value does not go below the recommended 8 DWORDs. This field controls both isochronous and non-isochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).</p>
1	RW	0x0	<p>ISOThrEn ISO IN Endpoints Threshold Enable When this bit is set, the core enables threshold for isochronous IN endpoints.</p>
0	RW	0x0	<p>NonISOThrEn Non-ISO IN Endpoints Threshold Enable When this bit is set, the core enables threshold for Non Isochronous IN endpoints.</p>

USBOTG_DIEPEMPMSK

Address: Operational Base + offset (0x0834)

Device IN endpoint FIFO empty interrupt mask register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	InEpTxfEmpMsk IN EP Tx FIFO Empty Interrupt Mask Bits These bits act as mask bits for DIEPINTn. TxFEmp interrupt One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DEACHINT

Address: Operational Base + offset (0x0838)

Device each endpoint interrupt register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	EchOutEPInt OUT Endpoint Interrupt Bits One bit per OUT endpoint: Bit 16 for OUT endpoint 0 ... Bit 31 for OUT endpoint 15
15:0	RO	0x0000	EchInEPInt IN Endpoint Interrupt Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DEACHINTMSK

Address: Operational Base + offset (0x083c)

Device each endpoint interrupt register mask

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	EchOutEpMsk OUT EP Interrupt Mask Bits One per OUT Endpoint: Bit 16 for IN endpoint 0 ... Bit 31 for endpoint 15
15:0	RW	0x0000	EchInEpMsk IN EP Interrupt Mask Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DIEPEACHMSKn

Address: Operational Base + offset (0x0840)

Device each IN endpoint -n interrupt Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	NAKMsk NAK interrupt Mask
12:10	RO	0x0	reserved
9	RW	0x0	BNAInIntrMsk BNA interrupt Mask
8	RW	0x0	TxfifoUndrnMsk Fifo Under run Mask
7	RO	0x0	reserved
6	RW	0x0	INEPNakEffMsk IN Endpoint NAK Effective Mask
5	RW	0x0	INTknEPMisMsk IN Token received with EP Mismatch Mask
4	RW	0x0	INTknTXFEmpMsk IN Token Received When TxFIFO Empty Mask
3	RW	0x0	TimeOUTMsk Timeout Condition Mask(Non-isochronous endpoints)
2	RW	0x0	AHBErrMsk AHB Error Mask
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DOEPEACHMSKn

Address: Operational Base + offset (0x0880)

Device each out endpoint-n interrupt register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	NYETMsk NYET interrupt Mask
13	RW	0x0	NAKMsk NAK interrupt Mask
12	RW	0x0	BbleErrMsk Babble interrupt Mask
11:10	RO	0x0	reserved
9	RW	0x0	BnaOutIntrMsk BNA interrupt Mask
8	RW	0x0	OutPktErrMsk OUT Packet Error Mask
7	RO	0x0	reserved
6	RW	0x0	Back2BackSETup Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.
5	RO	0x0	reserved
4	RW	0x0	OUTTknEPdisMsk OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only.

Bit	Attr	Reset Value	Description
3	RW	0x0	SetUPMsk SETUP Phase Done Mask Applies to control endpoints only.
2	RW	0x0	AHBErrMsk AHB Error
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DIEPCTL0

Address: Operational Base + offset (0x0900)

Device control IN endpoint 0 control register

Bit	Attr	Reset Value	Description
31	RWSC	0x0	EPEna Endpoint Enable When Scatter/Gather DMA mode is enabled, for IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. When Scatter/Gather DMA mode is disabled-such as in buffer-pointer based DMA mode-this bit indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting the following interrupts on this endpoint: Endpoint Disabled; Transfer Completed.
30	RWSC	0x0	EPDis Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.
29:28	RO	0x0	reserved
27	WO	0x0	SNAK Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.

Bit	Attr	Reset Value	Description
26	WO	0x0	CNAK Clear NAK A write to this bit clears the NAK bit for the endpoint.
25:23	RO	0x0	reserved
22	RW	0x0	TxFNum TxFIFO Number For Shared FIFO operation, this value is always set to 0, indicating that control IN endpoint 0 data is always written in the Non-Periodic Transmit FIFO. For Dedicated FIFO operation, this value is set to the FIFO number that is assigned to IN Endpoint 0.
21	RWSC	0x0	Stall STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.
20	RO	0x0	reserved
19:18	RO	0x0	EPType Endpoint Type Hardcoded to 00 for control
17	RO	0x0	NAKsts NAK Status Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status 1'b1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
16	RO	0x0	reserved
15	RO	0x1	USBActEP USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.

Bit	Attr	Reset Value	Description
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is not set. This field is not valid in Slave mode. Note: This field is valid only for Shared FIFO operations.
10:2	RO	0x0	reserved
1:0	RW	0x0	MPS Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes

USBOTG_DIEPINTn

Address: Operational Base + offset (0x0908)

Device Endpoint-n Interrupt Register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	W1C	0x0	NYETIntrpt NYET interrupt The core generates this interrupt when a NYET response is transmitted for a non-isochronous OUT endpoint.
13	W1C	0x0	NAKIntrpt NAK interrupt The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo.
12	W1C	0x0	BbleErrIntrpt BbleErr (Babble Error) interrupt The core generates this interrupt when babble is received for the endpoint.

Bit	Attr	Reset Value	Description
11	W1C	0x0	PktDrpSts Packet Dropped Status This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non-Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.
10	RO	0x0	reserved
9	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done Dependency: This bit is valid only when Scatter/Gather DMA mode is enabled.
8	W1C	0x0	TxfifoUndrn FIFO Under-run Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO under-run condition for this endpoint. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_ENDED_TX_FIFO==1; Threshold is enabled; OUT Packet Error(OutPktErr). Applies to OUT endpoints only. This interrupt is asserted when the core detects an overflow or a CRC error for an OUT packet. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_ENDED_TX_FIFO==1; Threshold is enabled.
7	W1C	0x0	TxFEmp Transmit FIFO Empty This bit is valid only for IN Endpoints. This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)).

Bit	Attr	Reset Value	Description
6	W1C	0x0	<p>INEPNakEff IN Endpoint NAK Effective Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. This bit is applicable only when the endpoint is enabled.</p> <p>Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint.</p>
5	W1C	0x0	<p>INTknEPMis IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received.</p> <p>Status Phase Received For Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints and only in Scatter Gather DMA mode. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in case of Scatter Gather DMA mode.</p>

Bit	Attr	Reset Value	Description
4	W1C	0x0	INTknTxFEmp IN Token Received When TxFIFO is Empty Indicates that an IN token was received when the associated TxFIFO periodic/nonperiodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. OUT Token Received When Endpoint Disabled (OUTTknEPdis) Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.
3	W1C	0x0	TimeOUT Timeout Condition In shared TX FIFO mode, applies to non-isochronous IN endpoints only. In dedicated FIFO mode, applies only to Control IN endpoints. In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint. SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.
2	W1C	0x0	AHBErr AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.
1	W1C	0x0	EPDisbld Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

Bit	Attr	Reset Value	Description
0	W1C	0x0	XferCompl Transfer Completed Interrupt Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled: For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.

USBOTG_DIEPTSIZn

Address: Operational Base + offset (0x0910)

Device endpoint n transfer size register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:29	RW	0x0	<p>MC Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints.</p> <p>2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets</p> <p>For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register (DIEPCTLn.NextEp). Received Data PID (RxDPID)</p> <p>Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint.</p> <p>2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA</p> <p>SETUP Packet Count (SUPCnt). Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive.</p> <p>2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets</p>
28:19	RW	0x000	<p>PktCnt Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. The power-on value is specified for Width of Packet Counters during coreConsultant configuration (parameter OTG_PACKET_COUNT_WIDTH). IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.</p>

Bit	Attr	Reset Value	Description
18:0	RW	0x00000	XferSize Transfer Size This field contains the transfer size in bytes for the current endpoint. The power-on value is specified for Width of Transfer Size Counters during configuration (parameter OTG_TRANS_COUNT_WIDTH). The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

USBOTG_DIEPDMA_n

Address: Operational Base + offset (0x0914)

Device endpoint-n DMA address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

USBOTG_DTXFSTS_n

Address: Operational Base + offset (0x0918)

Device IN endpoint transmit FIFO status register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>INEPTxFSpcAvail IN Endpoint TxFIFO Space Avail Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words.</p> <ul style="list-style-type: none"> 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 . n . 32,768) 16'h8000: 32,768 words available Others: Reserved

USBOTG_DIEPDMABn

Address: Operational Base + offset (0x091c)

Device endpoint-n DMA buffer address register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>DMABufferAddr DMA Buffer Address Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.</p>

USBOTG_DIEPCTLn

Address: Operational Base + offset (0x0920)

Device endpoint-n control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31	RWSC	0x0	<p>EPEna Endpoint Enable Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is enabled-such as for buffer-pointer based DMA mode: For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint ; For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.</p>
30	RWSC	0x0	<p>EPDis Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.</p>
29	WO	0x0	<p>SetD1PID Set DATA1 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Set Odd (micro) frame (SetOddFr). Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to odd (micro) frame. This field is not applicable for Scatter/Gather DMA mode.</p>

Bit	Attr	Reset Value	Description
28	WO	0x0	SetDOPID Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. In non-Scatter/Gather DMA mode: Set Even (micro) frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to even (micro) frame. When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in receipt descriptor structure.
27	WO	0x0	SNAK Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	WO	0x0	CNAK Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.

Bit	Attr	Reset Value	Description
25:22	RW	0x0	<p>TxFNum Tx FIFO Number Shared FIFO Operation: non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic Tx FIFO number. 4'h0: Non-Periodic Tx FIFO; Others: Specified Periodic Tx FIFO number. Note: An interrupt IN endpoint can be configured as a non-periodic endpoint for applications such as mass storage. The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic endpoint saves the extra periodic FIFO area. Dedicated FIFO Operation: these bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.</p>
21	RW	0x0	<p>Stall STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>
20	RW	0x0	<p>Snp Snoop Mode Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.</p>

Bit	Attr	Reset Value	Description
19:18	RW	0x0	<p>EPType Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint.</p> <p>2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt</p>
17	RO	0x0	<p>NAKsts NAK Status Applies to IN and OUT endpoints. Indicates the following:</p> <p>1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.</p> <p>When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>

Bit	Attr	Reset Value	Description
16	RO	0x0	<p>DPID Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID.</p> <p>1'b0: DATA0 1'b1: DATA1</p> <p>This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Even/Odd (Micro)Frame (EO_FrNum) In non-Scatter/Gather DMA mode:</p> <p>Applies to isochronous IN and OUT endpoints only. Indicates the (micro) frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register.</p> <p>1'b0: Even (micro)frame 1'b1: Odd (micro)frame</p> <p>When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure.</p>
15	RWSC	0x0	<p>USBActEP USB Active Endpoint Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.</p>

Bit	Attr	Reset Value	Description
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation. Note: This field is valid only for Shared FIFO operations.
10:0	RW	0x000	MPS Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

USBOTG_DOEPCTL0

Address: Operational Base + offset (0x0b00)

Device control OUT endpoint 0 control register

Bit	Attr	Reset Value	Description
31	RWSC	0x0	EPEna Endpoint Enable When Scatter/Gather DMA mode is enabled, for OUT endpoints this bit indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is disabled? such as for buffer-pointer based DMA mode)-this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.
30	WO	0x0	EPDis Endpoint Disable The application cannot disable control OUT endpoint 0.
29:28	RO	0x0	reserved
27	WO	0x0	SNAK Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.

Bit	Attr	Reset Value	Description
26	WO	0x0	CNAK Clear NAK A write to this bit clears the NAK bit for the endpoint.
25:22	RO	0x0	reserved
21	RWSC	0x0	Stall STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
20	RW	0x0	Snp Snoop Mode This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19:18	RO	0x0	EPType Endpoint Type Hardcoded to 2'b00 for control.
17	RO	0x0	NAKsts NAK Status Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit setting, the core always responds to SETUP data packets with an ACK handshake.
16	RO	0x0	reserved
15	RO	0x0	USBActEP USB Active Endpoint This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.
14:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RO	0x0	MPS Maximum Packet Size The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes

USBOTG_DOEPINTn

Address: Operational Base + offset (0x0b08)

Device endpoint-n control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	W1C	0x0	NYETIntrpt NYET interrupt The core generates this interrupt when a NYET response is transmitted for a non-isochronous OUT endpoint.
13	W1C	0x0	NAKIntrpt NAK interrupt The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo.
12	W1C	0x0	BbleErrIntrpt BbleErr (Babble Error) interrupt The core generates this interrupt when babble is received for the endpoint.
11	W1C	0x0	PktDrpSts Packet Dropped Status This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non-Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.
10	RO	0x0	reserved
9	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. Dependency: This bit is valid only when Scatter/Gather DMA mode is enabled.

Bit	Attr	Reset Value	Description
8	W1C	0x0	<p>TxfifoUndrn FIFO Underrun Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO under-run condition for this endpoint. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_ENDED_TX_FIFO==1, Threshold is enabled, OUT Packet Error (OutPktErr).</p> <p>Applies to OUT endpoints only. This interrupt is asserted when the core detects an overflow or a CRC error for an OUT packet.</p> <p>Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_ENDED_TX_FIFO==1, Thresholding is enabled.</p>
7	W1C	0x0	<p>TxFEmp Transmit FIFO Empty This bit is valid only for IN Endpoints. This interrupt is asserted when the Tx FIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the Tx FIFO Empty Level bit in the Core AHB Configuration register(GAHBCFG.NPTxFEmpLvl)).</p>
6	W1C	0x0	<p>INEPNakEff IN Endpoint NAK Effective Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. This bit is applicable only when the endpoint is enabled. Back-to-Back SETUP Packets Received (Back2BackSETUp) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint.</p>

Bit	Attr	Reset Value	Description
5	W1C	0x0	INTknEPMIs IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Status Phase Received For Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints and only in Scatter Gather DMA mode.
4	W1C	0x0	INTknTxFEmp IN Token Received When TxFIFO is Empty Indicates that an IN token was received when the associated TxFIFO periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. OUT Token Received When Endpoint Disabled (OUTTknEPdis) Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.
3	W1C	0x0	TimeOUT Timeout Condition In shared TX FIFO mode, applies to non-isochronous IN endpoints only. In dedicated FIFO mode, applies only to Control IN endpoints. In Scatter/Gather DMA mode, the Timeout interrupt is not asserted. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint. SETUP Phase Done (SetUp). Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.
2	W1C	0x0	AHBErr AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.

Bit	Attr	Reset Value	Description
1	W1C	0x0	EPDisbld Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.
0	W1C	0x0	XferCompl Transfer Completed Interrupt Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.

USBOTG_DOEPTSIZn

Address: Operational Base + offset (0x0b10)

Device endpoint n transfer size register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:29	RW	0x0	<p>MC Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per micro-frame on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints.</p> <p>2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets</p> <p>For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register (DIEPCTLn.NextEp). Received Data PID (RxDPID)</p> <p>Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint.</p> <p>2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA</p> <p>SETUP Packet Count (SUPCnt). Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive.</p> <p>2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets</p>
28:19	RW	0x000	<p>PktCnt Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. The power-on value is specified for Width of Packet Counters (parameter OTG_PACKET_COUNT_WIDTH).</p> <p>IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the Tx FIFO. OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the Rx FIFO.</p>

Bit	Attr	Reset Value	Description
18:0	RW	0x00000	XferSize Transfer Size This field contains the transfer size in bytes for the current endpoint. The power-on value is specified for Width of Transfer Size Counters (parameter OTG_TRANS_COUNT_WIDTH). The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

USBOTG_DOEPDMA_n

Address: Operational Base + offset (0x0b14)

Device Endpoint-n DMA Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

USBOTG_DOEPDMAB_n

Address: Operational Base + offset (0x0b1c)

Device endpoint-n DMA buffer address register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DMABufferAddr DMA Buffer Address Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

USBOTG_DOEPCCTLn

Address: Operational Base + offset (0x0b20)

Device endpoint-n control register

Bit	Attr	Reset Value	Description
31	RWSC	0x0	EPEna Endpoint Enable Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is enabled-such as for buffer-pointer based DMA mode: For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint; For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.
30	RWSC	0x0	EPDis Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.

Bit	Attr	Reset Value	Description
29	RO	0x0	SetD1PID Field0001 Abstract Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Set Odd (micro) frame (SetOddFr). Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to odd (micro) frame. This field is not applicable for Scatter/Gather DMA mode.
28	WO	0x0	SetD0PID Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. In non-Scatter/Gather DMA mode: Set Even (micro) frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to even (micro) frame. When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in receipt descriptor structure.
27	WO	0x0	SNAK Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	WO	0x0	CNAK Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.

Bit	Attr	Reset Value	Description
25:22	RW	0x0	<p>TxFNum Tx FIFO Number</p> <p>Shared FIFO Operation: non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic Tx FIFO number. 4'h0: Non-Periodic Tx FIFO; Others: Specified Periodic Tx FIFO number. Note: An interrupt IN endpoint can be configured as a non-periodic endpoint for applications such as mass storage. The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint using coreConsultant, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic endpoint saves the extra periodic FIFO area.</p> <p>Dedicated FIFO Operation: these bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.</p>
21	RW	0x0	<p>Stall STALL Handshake</p> <p>Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core.</p> <p>Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>
20	RW	0x0	<p>Snp Snoop Mode</p> <p>Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.</p>

Bit	Attr	Reset Value	Description
19:18	RW	0x0	<p>EPType Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint.</p> <p>2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt</p>
17	RO	0x0	<p>NAKSts NAK Status Applies to IN and OUT endpoints. Indicates the following:</p> <p>1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.</p> <p>When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet.</p>

Bit	Attr	Reset Value	Description
16	RO	0x0	<p>DPID Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID.</p> <p>1'b0: DATA0 1'b1: DATA1</p> <p>This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Even/Odd (Micro) Frame (EO_FrNum). In non-Scatter/Gather DMA mode: Applies to isochronous IN and OUT endpoints only. Indicates the (micro) frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register.</p> <p>1'b0: Even (micro)frame 1'b1: Odd (micro)frame</p> <p>When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure.</p>
15	RWSC	0x0	<p>USBActEP USB Active Endpoint Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.</p>

Bit	Attr	Reset Value	Description
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation. Note: This field is valid only for Shared FIFO operations.
10:0	RW	0x000	MPS Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

USBOTG_PCGCR

Address: Operational Base + offset (0x0b24)

Power and clock gating control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:14	RW	0x0802e	<p>RestoreValue Restore Value (Applicable only when Hibernation is enabled (OTG_EN_PWROPT=2). Defines port clock select for different speeds.</p> <ul style="list-style-type: none"> [31] if_dev_mode <ul style="list-style-type: none"> - 1: Device mode, core restored as device - 0: Host mode, core restored as host [30:29] p2hd_prt_spd (PRT speed) <ul style="list-style-type: none"> - 00: HS - 01: FS - 10: LS - 11: Reserved [28:27] p2hd_dev_enum_spd (Device enumerated speed) <ul style="list-style-type: none"> - 00: HS - 01: FS (30/60 MHz clock) - 10: LS - 11: FS (48 MHz clock) [26:20] mac_dev_addr (MAC device address) Device address [19] mac_termselect (Termination selection) <ul style="list-style-type: none"> - 0: HS_TERM (Program for High Speed) - 1: FS_TERM (Program for Full Speed) [18:17] mac_xcvrselect (Transceiver select) <ul style="list-style-type: none"> - 00: HS_XCVR (High Speed) - 01: FS_XCVR (Full Speed) - 10: LS_XCVR (Low Speed) - 11: LFS_XCVR (Reserved) [16] sh2pl_prt_ctl[0] <ul style="list-style-type: none"> - 1: port_power enabled - 0: port_power disabled [15:14] prt_clk_sel (Refer prt_clk_sel table)
13	RW	0x0	<p>EssRegRestored Essential Register Values Restored (Applicable only when Hibernation is enabled (OTG_EN_PWROPT=2). When a value of 1 is written to this field, it indicates that register values of essential registers have been restored.</p>
12:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RO	0x0	<p>RestoreMode Restore Mode (Applicable only when Hibernation is enabled (OTG_EN_PWR_OPT=2). The application should program this bit to specify the restore mode during RESTORE POINT before programming PCGCCTL.EssRegRest bit is set.</p> <p>Host Mode: 1'b0: Host Initiated Resume, Host Initiated Reset 1'b1: Device Initiated Remote Wake up</p> <p>Device Mode: 1'b0: Device Initiated Remote Wake up 1'b1: Host Initiated Resume, Host Initiated Reset</p>
8	RW	0x0	<p>ResetAfterSusp Reset After Suspend Applicable in Partial power-down mode. In partial power-down mode of operation, this bit needs to be set in host mode before clamp is removed if the host needs to issue reset after suspend. If this bit is not set, then the host issues resume after suspend. This bit is not applicable in device mode and non-partial power-down mode. In Hibernation mode, this bit needs to be set at RESTORE_POINT before PCGCCTL.EssRegRestored is set. In this case, PCGCCTL.restore_mode needs to be set to wait_restore.</p>
7	RO	0x0	<p>L1Suspended Deep Sleep This bit indicates that the PHY is in deep sleep when in L1 state.</p>
6	RO	0x0	<p>PhySleep PHY in Sleep This bit indicates that the PHY is in the Sleep state.</p>
5	RW	0x0	<p>Enbl_L1Gating Enable Sleep Clock Gating When this bit is set, core internal clock gating is enabled in Sleep state if the core cannot assert utmi_l1_suspend_n. When this bit is not set, the PHY clock is not gated in Sleep state.</p>
4	RO	0x0	reserved
3	RW	0x0	<p>RstPdwnModule Reset Power-Down Modules This bit is valid only in Partial Power-Down mode. The application sets this bit when the power is turned off. The application clears this bit after the power is turned on and the PHY clock is up.</p>

Bit	Attr	Reset Value	Description
2	RW	0x0	PwrClmp Power Clamp This bit is valid only in Partial Power-Down mode (OTG_EN_PWRROPT = 1). The application sets this bit before the power is turned off to clamp the signals between the power-on modules and the power-off modules. The application clears the bit to disable the clamping before the power is turned on.
1	RW	0x0	GateHclk Gate Hclk The application sets this bit to gate hclk to modules other than the AHB Slave and Master and wakeup logic when the USB is suspended or the session is not valid. The application clears this bit when the USB is resumed or a new session starts.
0	RW	0x0	StopPclk Stop Pclk The application sets this bit to stop the PHY clock (phy_clk) when the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit when the USB is resumed or a new session starts.

26.7 Interface description

Table 26-1 USB OTG 2.0 Interface Description

Module Pin	Direction	Pad Name	pinmux
VSSA	AG	VSSA	-
VCCA3P3	AP	VCCA3P3	-
VCCCORE1P2	AP	VCCCORE1P2	-
USB0PN	A	USB0PN	-
USBRBIAS	A	USBRBIAS	-
USB0PP	A	USB0PP	-
VBUS_0	A	VBUS_0	-
USB0ID	A	USB0ID	-

Note: **A**—Analog pad ; **AP**—Analog power; **AG**—Analog ground ;**DP**—Digital power ;**DG**—Digital ground;

26.8 Application Note

26.8.1 Resume from Suspend Mode

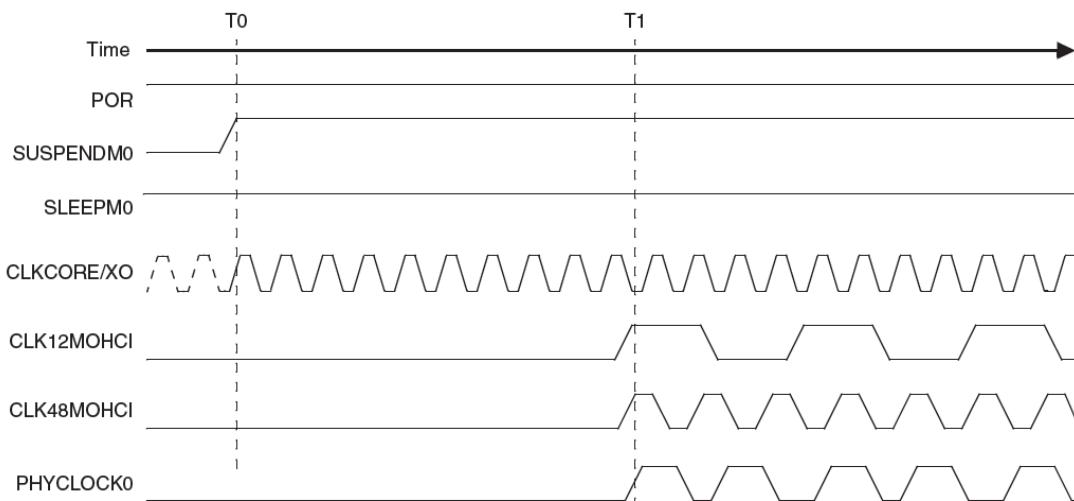


Fig. 26-11 Resume Timing Sequence

When COMMONONNN = 1'b1, $T1 < T0 + 805 \mu s$

When COMMONONNN = 1'b0, $T1 < T0 + 16 \mu s$

26.8.2 Reset a port

Because the assertion of PORTRESET can occur during data reception or transmission, PORTRESET must be de-asserted as follows:

- ❖ Reception:
 - ◆ FS device: After a minimum of 3 μs of stable SE0 on LINESTATE [1:0]
 - ◆ FS/LS host: After a minimum of 8 bit times of J state on LINESTATE [1:0]
 - ◆ HS host/device: A minimum of 150 μs after asserting PORTRESET
- ❖ Transmission:
 - ◆ FS device: After the controller sets both TXVALID and TXVALIDH to 1'b0, followed by a minimum of 3 μs of stable SE0 on LINESTATE [1:0]
 - ◆ FS/LS host: After the controller sets both TXVALID0 and TXVALIDH0 to 1'b0, followed by a minimum of 8 bit times of J state on LINESTATE [1:0]
 - ◆ HS host/device: A minimum of 150 μs after the controller sets both TXVALID0 and TXVALIDH0 to 1'b0. The preceding requirements ensure that there is no activity on the USB when PORTRESET0 is de-asserted.

To avoid any data glitches during port reset, the controller must place the USB 2.0 PHY into a safe state. A safe state for host and device ports is defined as follows:

- ❖ Host: The USB 2.0 PHY is set to Non-Driving (OPMODE [1:0] = 2'b01), and the 15-k Ω pull-down resistors are enabled (DPPULLDOWN and

DMPULLDOWN = 1'b1).

- ❖ Device: The USB 2.0 PHY is set to Non-Driving (OPMODE [1:0] = 2'b01), which disconnects the 1.5-kΩ resistor from the D+ line.

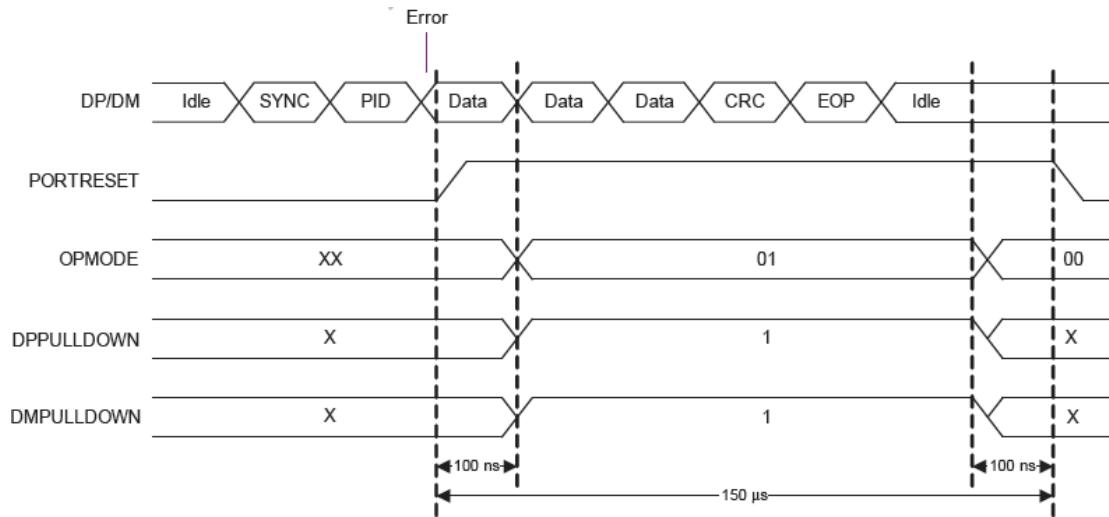


Fig. 26-12 Reset a port when receiving

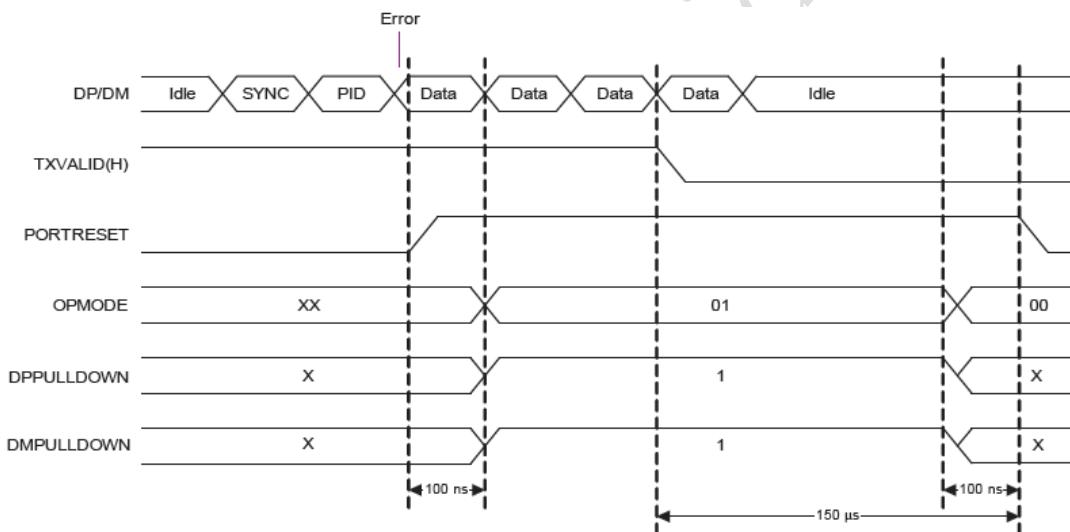


Fig. 26-13 Reset a port when transmitting

CRU_SOFRST4_CON contains the OTG reset signal. Please refer to "Chapter CRU" for more details.

26.8.3 Relative GRF Registers

GRF_UOC0_CON0 ~ GRF_UOC0_CON2 is OTG PHY register.

GRF_UOC0_CON3 is OTG Controller register.

Please refer to "Chapter GRF" for more details.

Rockchip Confidential

Chapter 27 SPDIF Transmitter

27.1 Overview

The SPDIF transmitter is a self-clocking, serial, unidirectional interface for the interconnection of digital audio equipment for consumer and professional applications, using linear PCM coded audio samples.

It provides the basic structure of the interface. Separate documents define items specific to particular applications.

When used in a professional application, the interface is primarily intended to carry monophonic or stereophonic programmes, at a 48 kHz sampling frequency and with a resolution of up to 24bits per sample; it may alternatively be used to carry signals sampled at 32 kHz or 44.1 kHz.

When used in a consumer application, the interface is primarily intended to carry stereophonic programmes, with a resolution of up to 20 bits per sample, an extension to 24 bits per sample being possible.

When used for other purposes, the interface is primarily intended to carry audio data coded other than as linear PCM coded audio samples. Provision is also made to allow the interface to carry data related to computer software or signals coded using non-linear PCM. The format specification for these applications is not part of this standard.

In all cases, the clock references and auxiliary information are transmitted along with the programme.

It supports following features:

- Support one internal 32-bit wide and 32-location deep sample data buffer
- Support two 16-bit audio data store together in one 32-bit wide location
- Support AHB bus interface
- Support biphase format stereo audio data output
- Support DMA handshake interface and configurable DMA water level
- Support sample data buffer empty and block terminate interrupt
- Support combine interrupt output
- Support 16 to 31 bit audio data left or right justified in 32-bit wide sample data buffer
- Support 48, 44.1, 32kHz sample rate
- Support 16, 20, 24 bits audio data transfer

27.2 Block Diagram

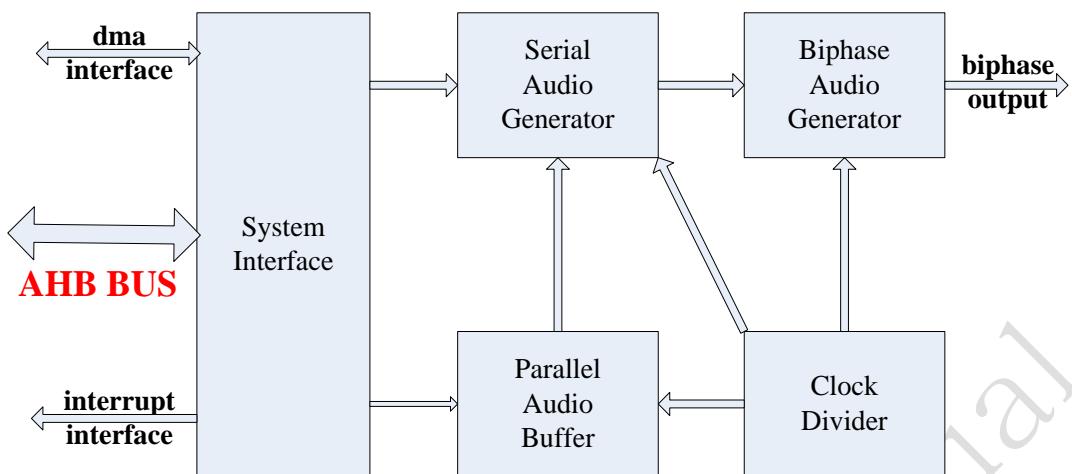


Fig.27-1SPDIF transmitter Block Diagram

The SPDIF is composed by:

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshaking interface.

Clock Divider

The clock divider implements clock generation function. It divides the source clock MCLK to generate the working clock used for the digital audio data transformation and transmission.

Parallel Audio Buffer

The parallel audio buffer stores the audio data to be transmitted. The size of the FIFO is 32bits x 32.

Serial Audio Converter

The serial audio converter converts the parallel audio data from the parallel audio buffer to the serial audio data.

Biphase Audio Generator

The biphase audio generator reads serial audio data from the serial audio converter and generates biphase audio data based on IEC-60958 standard.

27.3 Function Description

27.3.1 Frame Format

A frame is uniquely composed of two sub-frames. For linear coded audio applications, the rate of transmission of frames corresponds exactly to the source sampling frequency.

In the 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive sub-frames. The first

sub-frame(left channel in stereophonic operation and primary channel in monophonic operation) normally use preamble M. However, the preamble is changed to preamble B once every 192 frame to identify the start of the block structure used to organize the channel status information. The second sub-frame (right in stereophonic operation and secondary channel in monophonic operation) always use preamble W.

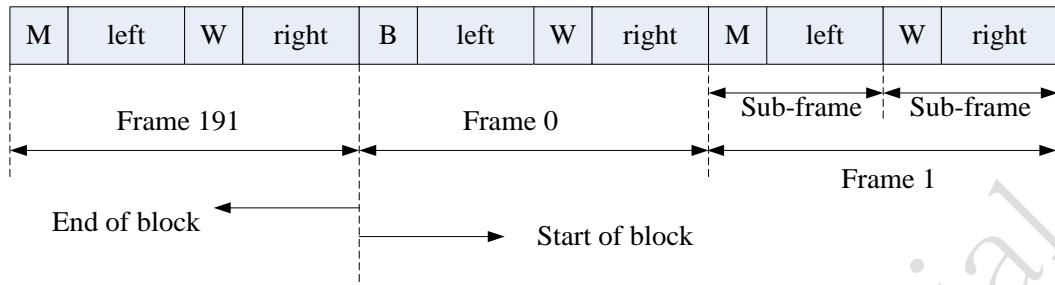


Fig.27-2SPDIF Frame Format

In the single channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried only in the first sub-frame and may be duplicated in the second sub-frame. If the second sub-frame is not carrying duplicate data, then time slot 28 (validity flag) shall be set to logical '1' (not valid).

27.3.2 Sub-frame Format

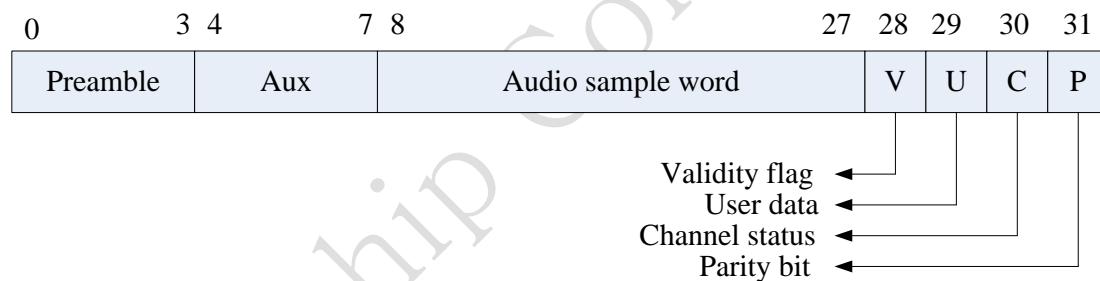


Fig.27-3SPDIF Sub-frame Format

Each sub-frame is divided into 32 time slots, numbered from 0 to 31. Time slot 0 to 3 carries one of the three permitted preambles. Time slot 4 to 27 carry the audio sample word in linear 2's complement representation. The MSB is carried by time slot 27. When a 24-bit coding range is used, the LSB is in time slot 4. When a 20-bit coding range is used, time slot 8 to 27 carry the audio sample word with the LSB in time slot 8. Time slot 4 to 7 may be used for other application. Under these circumstances, the bits in the time slot 4 to 7 are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (either 24 or 20), the unused LSBs are set to a logical '0'. For a non-linear PCM audio application or a data application the main data field may carry any other information. Time slot 28 carries the validity flag associated with the main data field. Time slot 29 carries 1 bit of the user data associated with the audio channel transmitted in the same sub-frame. Time slot 30 carries one bit of the channel status words associated with the main data field channel transmitted in the same sub-frame. Time slot 31 carries a parity bit such that time slots 4 to 31 inclusive carries an even number of ones and an even number of zeros.

27.3.3 Channel Coding

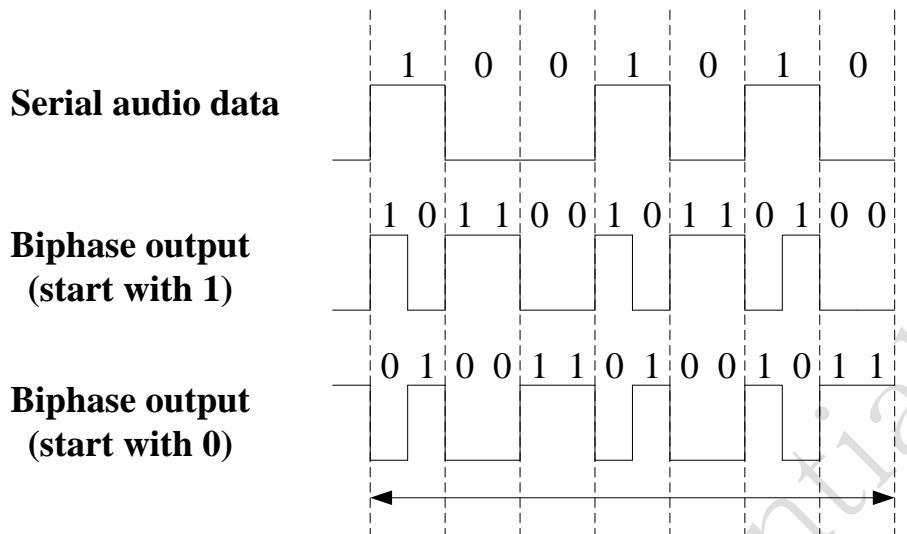


Fig.27-4SPDIF Channel Coding

To minimize the direct current component on the transmission line, to facilitate clock recovery from the data stream and to make the interface insensitive to the polarity of connections, time slots 4 to 31 are encoded in biphasic-mark.

Each bit to be transmitted is represented by a symbol comprising two consecutive binary states. The first state of a symbol is always different from the second state of the previous symbol. The second state of the symbol is identical to the first if the bit to be transmitted is logical '0'. However, it is different from the first if the bit is logical '1'.

27.3.4 Preamble

Preambles are specific patterns providing synchronization and identification of the sub-frames and blocks.

To achieve synchronization within one sampling period and to make this process completely reliable, these patterns violate the biphasic-mark code rules, thereby avoiding the possibility of data imitating the preambles.

A set of three preambles is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states. The first state of the preamble is always different from the second state of the previous symbol.

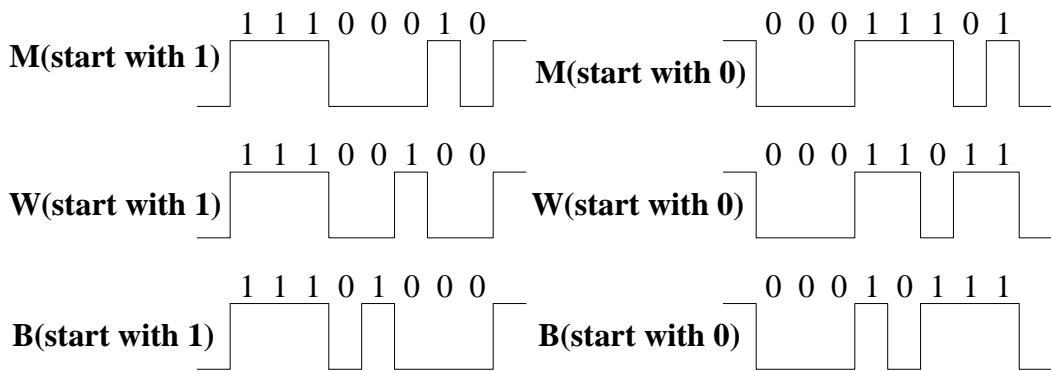


Fig.27-5SPDIF Preamble

Like biphasic code, these preambles are dc free and provide clock recovery. They differ in at least two states from any valid biphasic sequence.

27.4 Register description

27.4.1 Register Summary

Name	Offset	Size	Reset value	Description
SPDIF_CFGR	0x00	W	0x0	Transfer Configuration Register
SPDIF_SDBLR	0x04	W	0x0	Sample Date Buffer Level Register
SPDIF_DMACR	0x08	W	0x0	DMA Control Register
SPDIF_INTCR	0x0C	W	0x0	Interrupt Control Register
SPDIF_INTSR	0x10	W	0x0	Interrupt Status Register
SPDIF_XFER	0x18	W	0x0	Transfer Start Register
SPDIF_SMPDR	0x20 or 0x200 ~0x3FC	W	0x0	Sample Data Register
SPDIF_VLDFR	0x60~ 0x8C	W	0x0	Validity Flag Register
SPDIF_USRDR	0x90~ 0xBC	W	0x0	User Data Register
SPDIF_CHNSR	0xC0~	W	0x0	Channel Status Register

	0xEC			
--	------	--	--	--

Notes:

Size: B – Byte (8 bits) access, HW – Half WORD (16 bits) access, W –WORD (32 bits) access

27.4.2 Detail Register Description

SPDIF_CFGR

Address: operationalbase+offset(0x00)

Transfer Configuration Register

Bit	Attr	Reset Value	Description
31:24	-	-	reserved
23:16	RW	0x0	MCD mclk divider Fmclk/Fsdo This parameter can be caculated by Fmclk/(Fs*128). Fs=the sample frequency be wanted
15:8	-	-	reserved
7	W	0x0	CLR mclk domain logic clear Write 1 to clear mclk domain logic. Read return zero.
6	RW	0x0	CSE Channel status enable 0: disable 1: enable
5	RW	0x0	UDE User data enable 0: disable 1: enable
4	RW	0x0	VFE Validity flag enable 0: disable 1: enable
3	RW	0x0	ADJ audio data justified The justified format of the audio data stored in FIFO. If right justified is chosen, the lower bits of the 32-bit data stored in the FIFO are valid for data sample, otherwise the higher ones are valid. 1'b0:Right justified

			1'b1:Left justified
2	RW	0x0	<p>HWT</p> <p>Halfword word transform enable</p> <p>This bit is only valid when data width is 16-bit. When enabled, the received data from the AHB will be treated as two sets of 16bit valid data sample and stored in the same unit of the FIFO. When disabled, only the 16bit of the data is valid and stored into the FIFO. Whether higher bit or lower 16 bit is valid depends on the value of ADJ</p> <p>1'b0:disable</p> <p>1'b1:enable</p>
1:0	RW	0x0	<p>VDW</p> <p>Valid data width</p> <p>2'b00: 16bit</p> <p>2'b01: 20bit</p> <p>2'b10: 24bit</p> <p>2'b11: reserved</p>

SPDIF_SDBLR

Address: operationalbase+offset(0x04)

Sample Date Buffer Level Register

Bit	Attr	Reset Value	Description
31:6	-	-	Reserved.
5:0	R	0x0	<p>SDBLR</p> <p>Sample Date Buffer Level Register. Contains the number of valid data entries in the sample data buffer.</p>

SPDIF_DMCR

Address: operationalbase+offset(0x08)

DMA Control Register

Bit	Attr	Reset Value	Description

31:6	-	-	Reserved.
5	RW	0x0	TDE Transmit DMA Enable 0: Transmit DMA disabled 1: Transmit DMA enabled
4:0	RW	0x0	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the Sample Date Buffer is equal to or below this field value

SPDIF_INTCR

Address: operationalbase+offset(0x0C)

Interrupt Control Register

Bit	Attr	Reset Value	Description
31: 17	-	-	reserved
16	W	0x0	BTTIC Block transfer terminate interrupt clear Write 1 to clear block transfer terminate interrupt.
15: 10	-	-	reserved
9:5	RW	0x0	SDBT Sample Date Buffer Threshold Sample Date Buffer Threshold for empty interrupt
4	RW	0x0	SDBEIE Sample Date Buffer empty interrupt enable 0: disable 1: enable
3	RW	0x0	BTTIE Block transfer terminate interrupt enable 0: disable 1: enable
2:0	-	-	reserved

SPDIF_INTSR

Address: operationalbase+offset(0x10)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31: 5	-	-	reserved
4	R	0x0	SDBEIS Sample Date Buffer empty interrupt status 0: inactive 1: active
3	R	0x0	BTTIS Block transfer terminate interrupt status 0: inactive 1: active
2:0	-	-	Reserved.

SPDIF_XFER

Address: operationalbase+offset(0x18)

Transfer Start Register

Bit	Attr	Reset Value	Description
31: 1	-	-	Reserved.
0	RW	0x0	XFER Transfer Start Register

SPDIF_SMPDR

Address: operationalbase+offset(0x20 or 0x200 ~0x3FC)

Sample Data Register

Bit	Attr	Reset Value	Description
31:0	W	0x0	SMPDR Sample Data Register

SPDIF_VLDFR

Address: operationalbase+offset(0x60~0x8C)

Validity Flag Register

Bit	Attr	Reset Value	Description

31:0	RW	0x0	VLDFR Validity Flag Register
------	----	-----	---------------------------------

SPDIF_USRDR

Address: operationalbase+offset(0x90~0xBC)

User Data Register

Bit	Attr	Reset Value	Description
31:0	RW	0x0	USRDR User Data Register

SPDIF_CHNSR

Address: operationalbase+offset(0xC0~0xEC)

Channel Status Register.

Bit	Attr	Reset Value	Description
31:0	RW	0x0	CHNSR Channel Status Register

27.5 Interface description

Table 27-1 Input clock description in clock architecture diagram

Module Pin	Dir.	Pad Name	IOMUX Setting
spdif_tx	O	SPDIFTx_GPIO3d3	GRF_GPIO3D_IOMUX[6]=1'b11

Notes: I=input, O=output, I/O=input/output, bidirectional

27.6 Application Notes

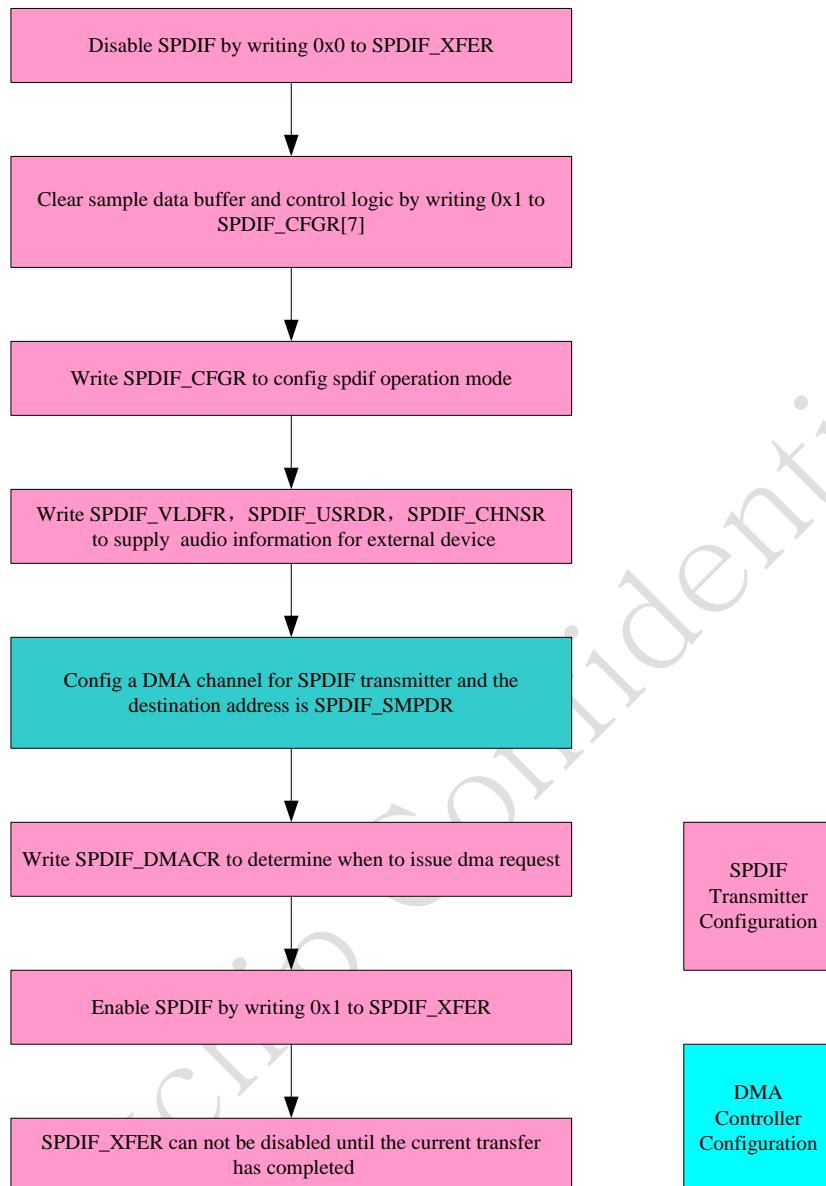


Fig.27-6 SPDIF transmitter operation flow chart

The Fig shows the operation flow of SPDIF operation. Note that the configuration register can be written only when the transfer is stopped.

Chapter 28 I2S 2-channel

28.1 Overview

The I2S/PCM controller is designed for interfacing between the AHB bus and the I2S bus.

The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and is invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.

I2S bus is widely used in the devices such as ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

28.1.1 Features

Not only I2S but also PCM mode stereo audio output and input are supported in I2S/PCM1/2 controller.

- Support two internal 32-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving audio data
- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshaking interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combined interrupt output
- Support 2-channel audio transmitting in I2S mode and PCM mode
- Support 2-channel audio receiving in I2S and PCM mode
- Support up to 192kHz sample rate
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 2 independent LRCK signals, one for receiving and the other for transmitting audio data
- Support configurable SCLK and LRCK polarity
- Support SCLK is equivalent to MCLK divided by an even number range from 2 to 64 in master mode

28.2 Block Diagram

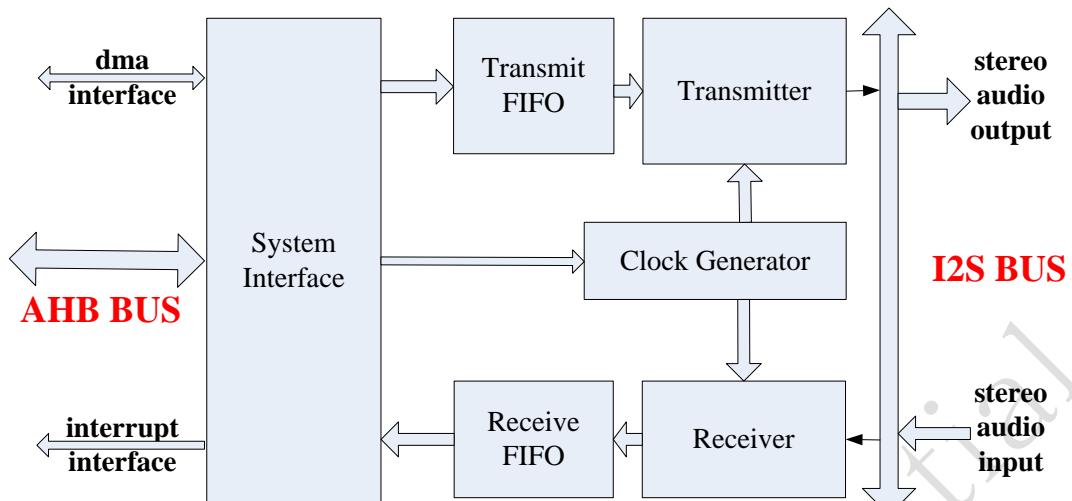


Fig.28-1I2S/PCM1/2 controller (2 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshaking interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either a master or a slave, with I2S or PCM mode surround (up to 7.1 channel) serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either a master or a slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

28.3 Function description

In the I2S/PCM1/2 controller, there are four types: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

In broadcasting application, the I2S/PCM1/2 controller is used as a transmitter

and external or internal audio CODEC is used as a receiver. In recording application, the I2S/PCM1/2 controller is used as a receiver and external or internal audio CODEC is used as a transmitter. Either the I2S/PCM1/2 controller or the audio CODEC can act as a master or a slave, but if one is master, the other must be slave.

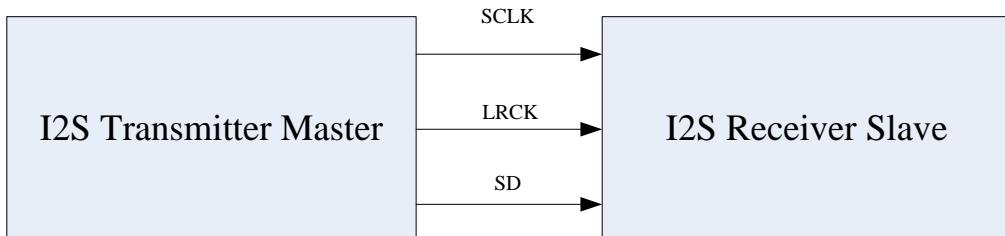


Fig.28-2I2S transmitter-master & receiver-slave condition

When the transmitter acts as a master, it sends all signals to the receiver (the slave), and CPU controls when to send clock and data to the receiver. When acts as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from the receiver (the master) to the transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when the transmitter to send data.

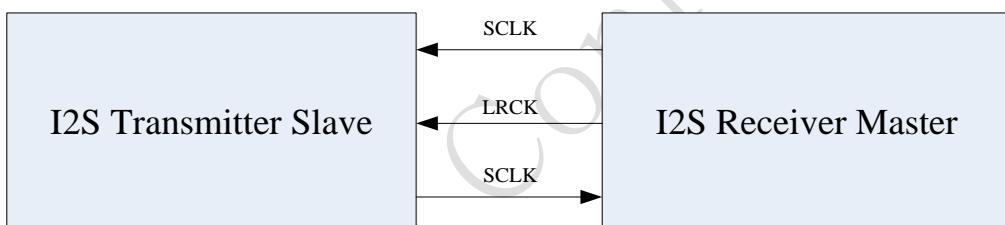


Fig.28-3I2S transmitter-slave & receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (the slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

28.3.1 I2S normal mode

This is the waveform of I2S normal mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

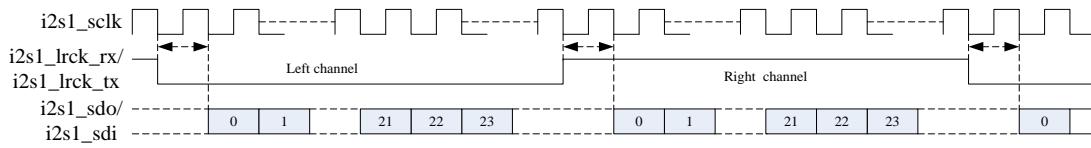


Fig.28-4I2S normal mode timing format

28.3.2 I2S left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s1_lrck_rx / i2s1_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

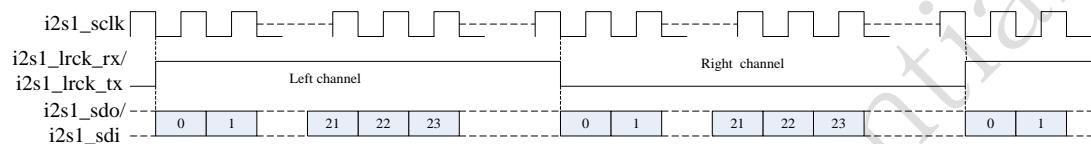


Fig.28-5I2S left justified mode timing format

28.3.3 I2S right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s1_lrck_rx/ i2s1_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first; but what is different from I2S normal or left justified mode, the last bit of the transferred data is aligned to the transition edge of the LRCK signal while one bit is transferred at one SCLK cycle. The range of SD signal width is from 16 to 32bits.

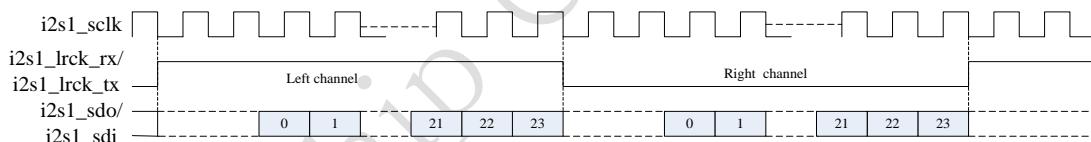


Fig.28-6I2S right justified mode timing format

28.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSBor LSB) at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

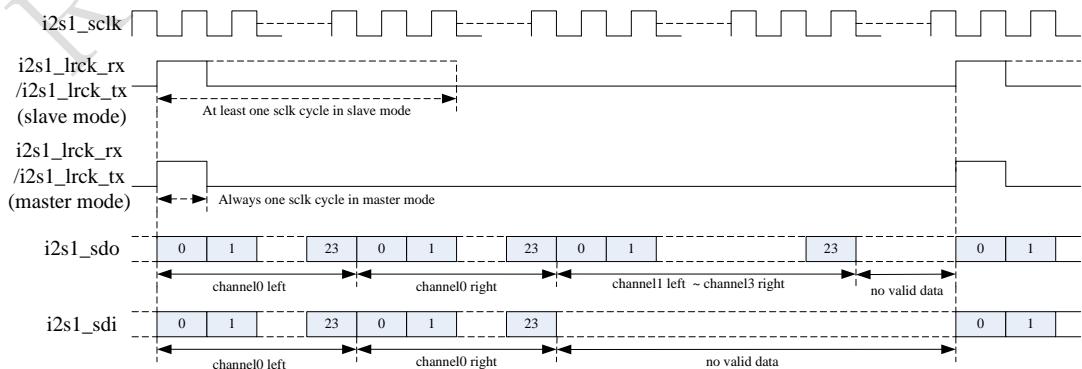


Fig.28-7PCM early mode timing format

28.3.5 PCM late1 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

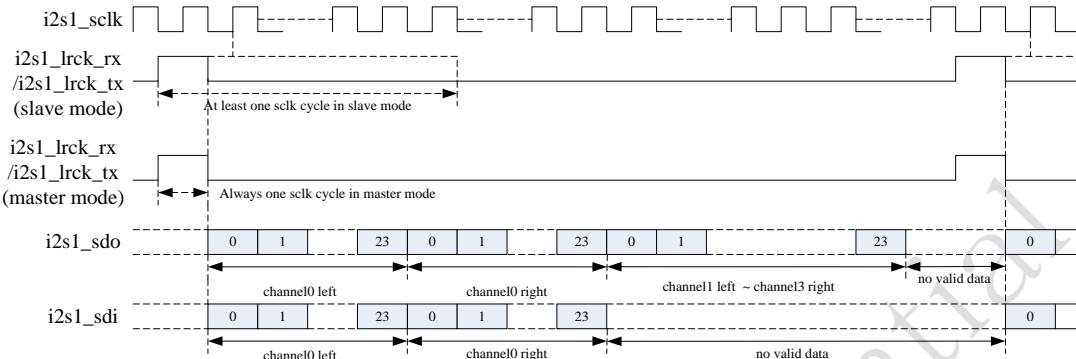


Fig.28-8PCM late1 mode timing format

28.3.6 PCM late2 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

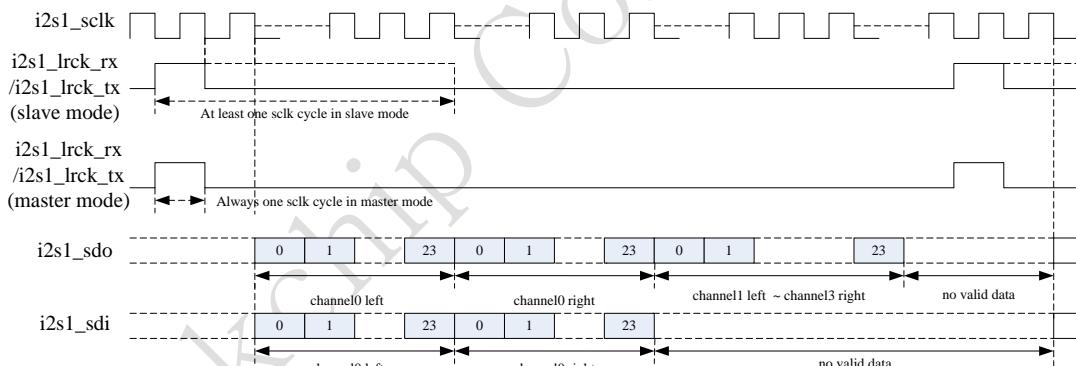


Fig.28-9PCM late2 mode timing format

28.3.7 PCM late3 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

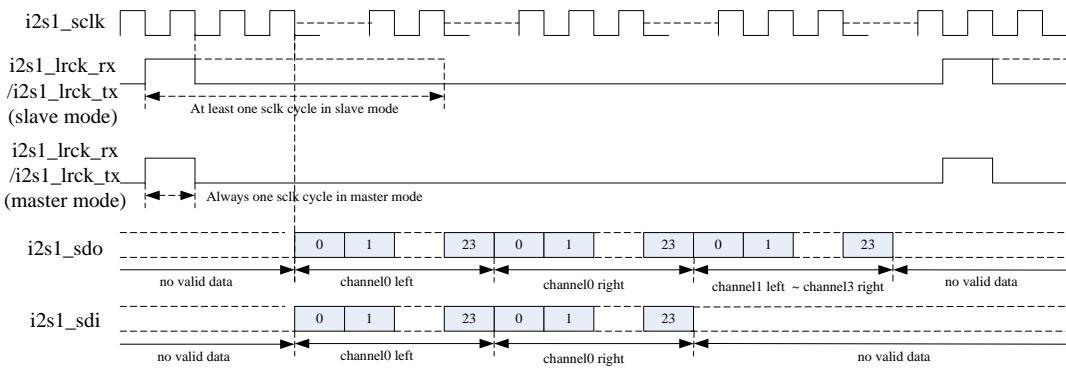


Fig.28-10PCM late3 mode timing format

28.4 Register description

28.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
I2Sx_TXCR	0x0000	W	0x0000000f	transmit operation control register
I2Sx_RXCR	0x0004	W	0x0000000f	receive operation control register
I2Sx_CKR	0x0008	W	0x00071f1f	clock generation register
I2Sx_FIFOLR	0x000c	W	0x00000000	FIFO level register
I2Sx_DMACR	0x0010	W	0x001f0000	DMA control register
I2Sx_INTCR	0x0014	W	0x00000000	interrupt control register
I2Sx_INTSR	0x0018	W	0x00000000	interrupt status register
I2Sx_XFER	0x001c	W	0x00000000	Transfer Start Register
I2Sx_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2Sx_TXDR	0x0400 ~0x7FC	W	0x00000000	Transmit FIFO Data Register
I2Sx_RXDR	0x0800 ~0xBFC	W	0x00000000	Receive FIFO Data Register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, x=1,2

28.4.2 Detail Register Description

I2Sx_TXCR

Address: Operational Base + offset (0x0000)
transmit operation control register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT right justified counter (Can be written only when XFER[0] bit is 0.) Only valid in I2S Right justified format and slave tx mode is selected. Start to transmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	CSR Channel select register Must be 2'b00.

Bit	Attr	Reset Value	Description
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[0] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[0] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[0] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[0] bit is 0.) 0: I2S format 1: PCM format

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	<p>VDW Valid Data width (Can be written only when XFER[0] bit is 0.)</p> <p>0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit n:(n+1)bit 28:29bit 29:30bit 30:31bit 31:32bit</p>

I2Sx_RXCR

Address: Operational Base + offset (0x0004)
receive operation control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	<p>HWT Halfword word transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid to AHB/APB bus, high 16 bit data invalid.</p>
13	RO	0x0	reserved
12	RW	0x0	<p>SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified</p>
11	RW	0x0	<p>FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 0:MSB 1:LSB</p>

Bit	Attr	Reset Value	Description
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[1] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[1] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[1] bit is 0.) 0:i2s 1:pcm
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit n:(n+1)bit 28:29bit 29:30bit 30:31bit 31:32bit

I2Sx_CKR

Address: Operational Base + offset (0x0008)
clock generation register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	MSS Master/slave mode select (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:master mode(sclk output) 1:slave mode(sclk input)

Bit	Attr	Reset Value	Description
26	RW	0x0	<p>CKP Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedgesclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedgesclk</p>
25	RW	0x0	<p>RLP Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: normal polarity (I2S normal: low for left channel, high for right channel) I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1: oppsite polarity (I2S normal: high for left channel, low for right channel) I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)</p>
24	RW	0x0	<p>TLP Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: normal polarity (I2S normal: low for left channel, high for right channel) I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1: oppsite polarity (I2S normal: high for left channel, low for right channel) I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)</p>

Bit	Attr	Reset Value	Description
23:16	RW	0x07	<p>MDIV mclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Serial Clock Divider = Fmclk / Ftxsclk-1.(mclkfrequency / txsclk frequency-1)</p> <p>0 :Fmclk=Ftxsclk; 1 :Fmclk=2*Ftxsclk; 2,3 :Fmclk=4*Ftxsclk; 4,5 :Fmclk=6*Ftxsclk; 2n,2n+1:Fmclk=(2n+2)*Ftxsclk; 60,61:Fmclk=62*Ftxsclk; 62,63:Fmclk=64*Ftxsclk; 252,253:Fmclk=254*Ftxsclk; 254,255:Fmclk=256*Ftxsclk;</p>
15:8	RW	0x1f	<p>RSD Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Receive sclk divider= Fsclk/Frxlrck</p> <p>0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs n: (n+1)fs 253: 254fs 254: 255fs 255: 256fs</p>
7:0	RW	0x1f	<p>TSD Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Transmit sclk divider=Ftxsclk/Ftxlrck</p> <p>0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs n: (n+1)fs 253: 254fs 254: 255fs 255: 256fs</p>

I2Sx_FIFOLR

Address: Operational Base + offset (0x000c)

FIFO level register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	RO	0x00	RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO.
23:6	RO	0x0	reserved
5:0	RO	0x00	TFL Transmit FIFO Level Contains the number of valid data entries in the transmit FIFO0.

I2Sx_DMCR

Address: Operational Base + offset (0x0010)

DMA control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 0 : Receive DMA disabled 1 : Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if CSR=00;TXFIFO1 if CSR=01,TXFIFO2 if CSR=10,TXFIFO3 if CSR=11)is equal to or below this field value.

I2Sx_INTCR

Address: Operational Base + offset (0x0014)

interrupt control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x00	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX overrun interrupt enable 0:disable 1:enable
16	RW	0x0	RXFIE RX full interrupt enable 0:disable 1:enable
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO (TXFIFO0 if CSR=00; TXFIFO1 if CSR=01, TXFIFO2 if CSR=10, TXFIFO3 if CSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC TX underrun interrupt clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX underrun interrupt enable 0:disable 1:enable
0	RW	0x0	TXEIE TX empty interrupt enable 0:disable 1:enable

I2Sx_INTSR

Address: Operational Base + offset (0x0018)

interrupt status register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI RX overrun interrupt 0:inactive 1:active

Bit	Attr	Reset Value	Description
16	RO	0x0	RXFI RX full interrupt 0:inactive 1:active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX underrun interrupt 0:inactive 1:active
0	RO	0x0	TXEI TX empty interrupt 0:inactive 1:active

I2Sx_XFER

Address: Operational Base + offset (0x001c)

Transfer Start Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS RX Transfer start bit 0:stop RX transfer. 1:start RX transfer
0	RW	0x0	TXS TX Transfer start bit 0:stop TX transfer. 1:start TX transfer

I2Sx_CLR

Address: Operational Base + offset (0x0020)

SCLK domain logic clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC RX logic clear This is a self cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX logic clear This is a self cleared bit. Write 1 to clear all transmit logic.

I2Sx_TXDR

Address: Operational Base + offset (0x0024)

Transmit FIFO Data Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transmit FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2Sx_RXDR

Address: Operational Base + offset (0x0028)

Receive FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR Receive FIFO Data Register When the register is read, data in the receive FIFO is accessed.

28.5 Interface description

I2S has 2 IOMUX, which controlled by GRF_SOC_CON1[5].

When GRF_SOC_CON1[5] is 1'b0, the IOMUX is as follow.

Module Pin	Direction	Pad Name	IOMUX Setting
I2S			
i2s_clk	O	IO_I2Smclk_MMC1clkout_XIN32k_GPIO1a0	GRF_GPIO1A_IOMUX[1:0]=2'b1
i2s_sclk	I/O	IO_I2Ssclk_MMC1d0_PMICsleep1_GPIO1a1	GRF_GPIO1A_IOMUX[3:2]=2'b1
i2s_lrck_rx	I/O	IO_I2Slrckrx_MMC1d1_GPIO1a2	GRF_GPIO1A_IOMUX[5:4]=2'b1
i2s_lrck_tx	I/O	IO_I2Slrcktx_GPIO1a3	GRF_GPIO1A_IOMUX[6]=1'b1
i2s_sdo	O	IO_I2Ssdo_MMC1d2_GPIO1a4	GRF_GPIO1A_IOMUX[9:8]=2'b1
i2s_sdi	I	IO_I2Ssdi_MMC1d3_GPIO1a5	GRF_GPIO1A_IOMUX[11:10]=2'b1

When GRF_SOC_CON1[5] is 1'b1, the IOMUX is as follow.

Module Pin	Direction	Pad Name	IOMUX Setting
I2S			
i2s_clk	O	IO_I2S1mclk_GPIO0b0	GRF_GPIO0B_IOMUX[0]=1'b1
i2s_sclk	I/O	IO_I2S1sclk_SPIclk_GPIO0b1	GRF_GPIO0B_IOMUX[3:2]=2'b01

i2s_lrck_rx	I/O	IO_I2S1lrck_rx_SPIxdm_GPIO0b3	GRF_GPIO0B_IOMUX[7:6]=2'b01
i2s_lrck_tx	I/O	IO_I2S1lrck_tx_GPIO0b4	GRF_GPIO0B_IOMUX[8]=1'b1
i2s_sdo	O	IO_I2S1sdo_SPIxdm_GPIO0b5	GRF_GPIO0B_IOMUX[11:10]=2'b1
i2s_sdi	I	IO_I2S1sdi_SPIcsn0m_GPIO0b6	GRF_GPIO0B_IOMUX[13:12]=1'b1

28.6 Application Notes

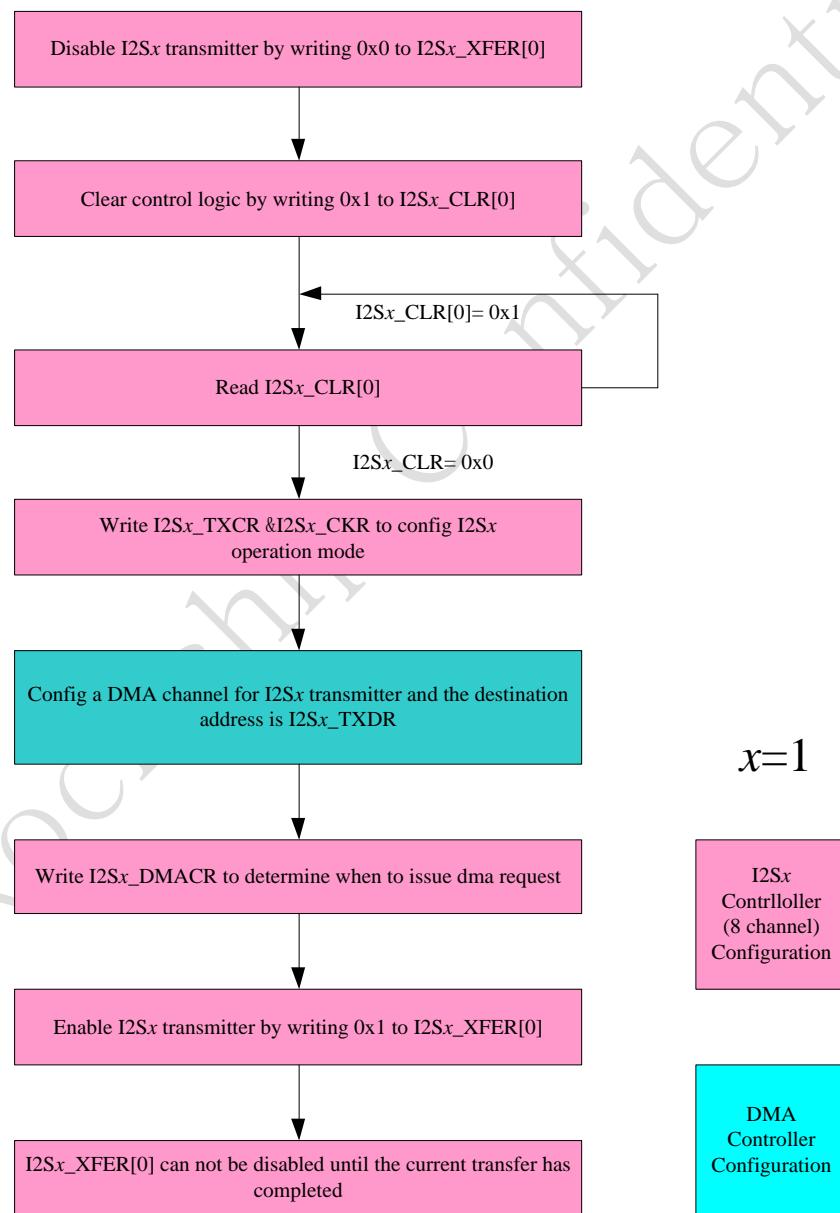


Fig.28-11I2S/PCM1/2 controller transmit operation flow chart

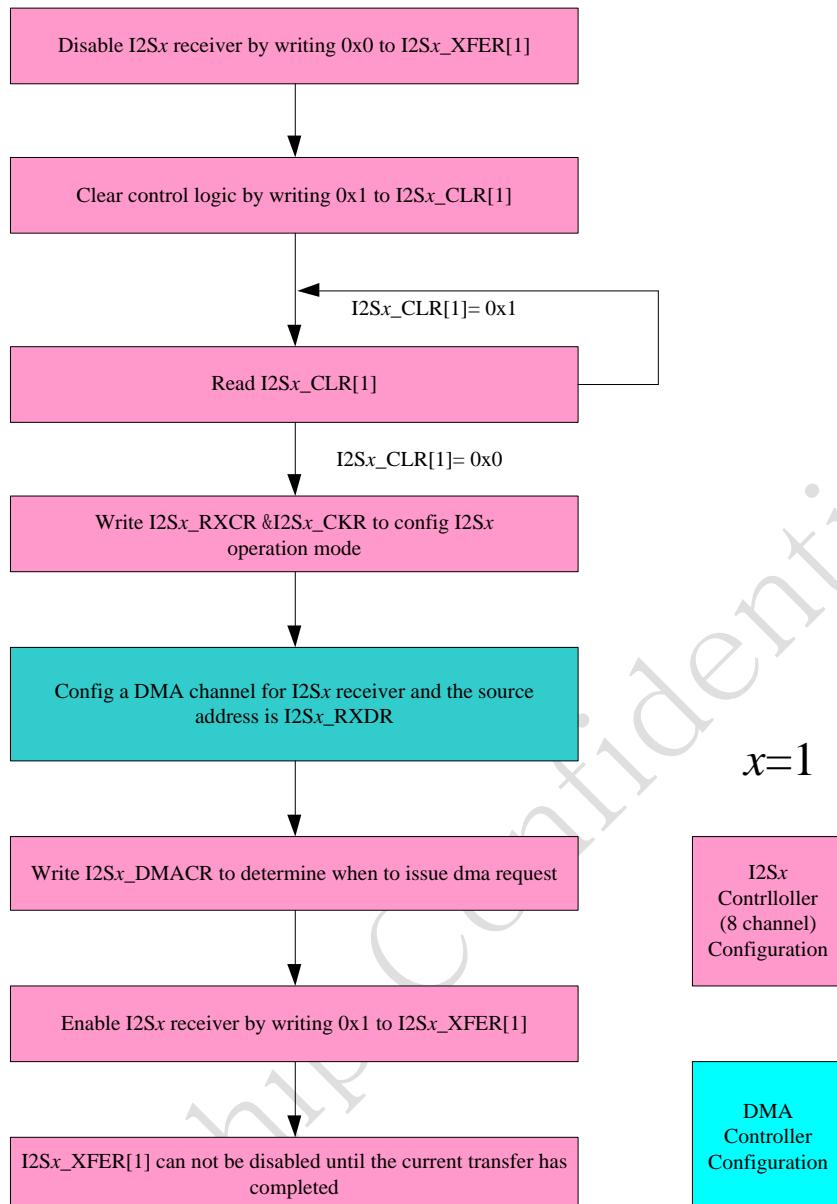


Fig.28-12I2S/PCM1/2 controller receive operation flow chart

Chapter 29 Audio Codec

29.1 Overview

The Audio Codec is a capless, low power, high resolution, stereo CODEC solution that employs Sigma-Delta noise-shaping technique. The ADC with 24bit resolution, DAC with 24bit resolution and power amplifier are integrated.

Key Features

- 24 bit DAC with 95dB SNR
- Support DC-coupled capless headphone output
- Support 16Ω to 32Ω headphone output and speaker output
- 24 bit ADC with 92dB SNR
- Support single-ended and differential microphone input and line input
- Automatic Level Control (ALC) for smooth audio recording
- Support Mono, Stereo, 5.1 and 7.1 HiFi channel performance
- Programmable input and output analog gains
- Digital interpolation and decimation filter integrated
- Sampling rate of 8/12/16/24/32/44.1/48/96kHz

29.2 Block Diagram

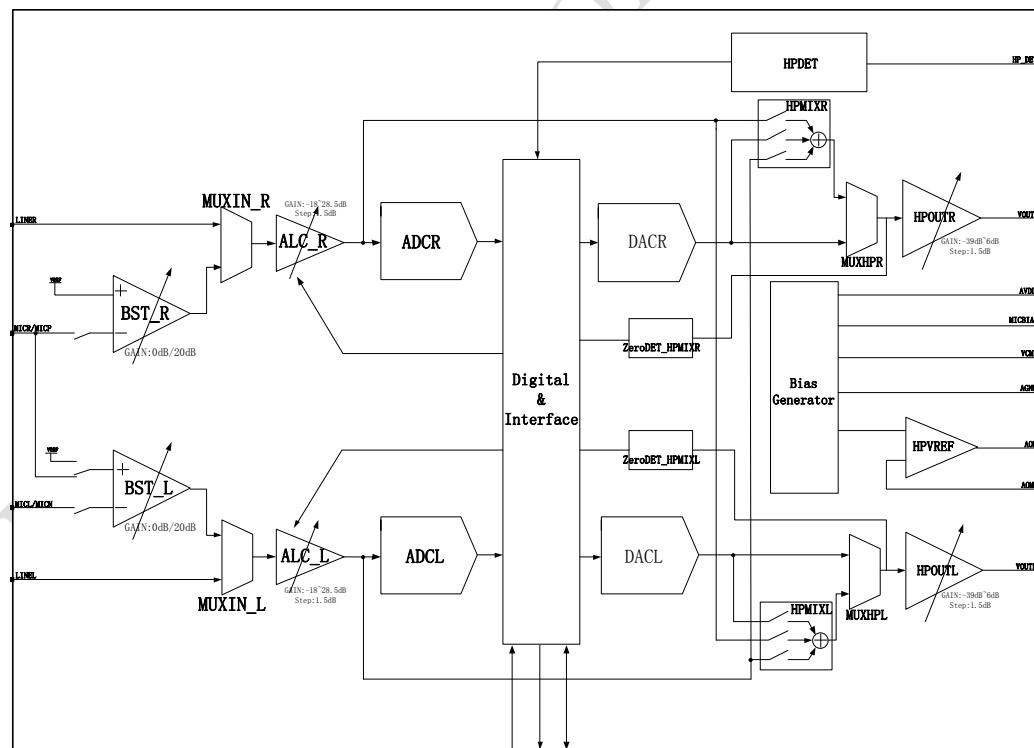


Fig.29-1 Audio Codec Block Diagram

29.3 Electrical Specification

Test conditions: AVDD = 3.3V, DVDD = 1.1V, TA = 25°C, 1KHz Sine Input, Fs = 48KHz.

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Condition						
Analog Supply	AVDD		2.97	3.3	3.63	V
Digital Supply	DVDD		0.99	1.1	1.21	V
Junction Temperature	T _J		-40		125	°C
Microphone Bias						
Bias Voltage	V _{MICB}		0.5* AVDD		0.85* AVDD	V
Bias Current	I _{MICB}				3	mA
Microphone Gain Boost PGA						
Programmable Gain	G _{BST}		0		20	dB
Gain Step Size				20		dB
Input Resistance	R _{IN}	G _{BST} =0dB		83		KΩ
		G _{BST} =20dB		15		KΩ
Input Capacitance	C _{IN}			10		pF
ALC PGA						
Programmable Gain	G _{ALC}		-18		28.5	dB
Gain Step Size				1.5		dB
ADC Input Path (Microphone or Line input to ADC)						
Signal to Noise Ratio	SNR	A-weighted		95		dB
Total Harmonic Distortion	THD	-3dBFS input		-83		dB
Power Supply Rejection	PSRR	1KHz		80		dB
ADC						
Signal to Noise Ratio	SNR	A-weighted		92		dB
Total Harmonic Distortion	THD	-3dBFS input		-81		dB
Channel Separation				80		dB
DAC						
Signal to Noise Ratio	SNR	A-weighted		95		dB
Total Harmonic Distortion	THD	-3dBFS output 10KΩ load		-84		dB
Channel Separation				85		dB
Output Driver						
Programmable Gain	G _{DRV}		-39		6	dB
Gain Step Size				1.5		dB
Output Resistance	R _{OUT}			1		KΩ
Output Capacitance	C _{OUT}			20		pF
Power Supply Rejection	PSRR	1KHz		70		dB
Line Output						
Signal to Noise Ratio	SNR	A-weighted		93		dB
Total Harmonic Distortion	THD	-3dBFS output 10KΩ load		-84		dB
Headphone Output						
Signal to Noise Ratio	SNR	A-weighted		92		dB
Total Harmonic Distortion	THD	16Ω load P _O =20mW		-70		dB
		32Ω load P _O =20mW		-75		dB

Power Consumption						
Standby				0.05		mA
Stereo Recording				6.5		mA
Stereo Playback		Quiescent output		11		mA

29.4 Function description

29.4.1 Digital Interface

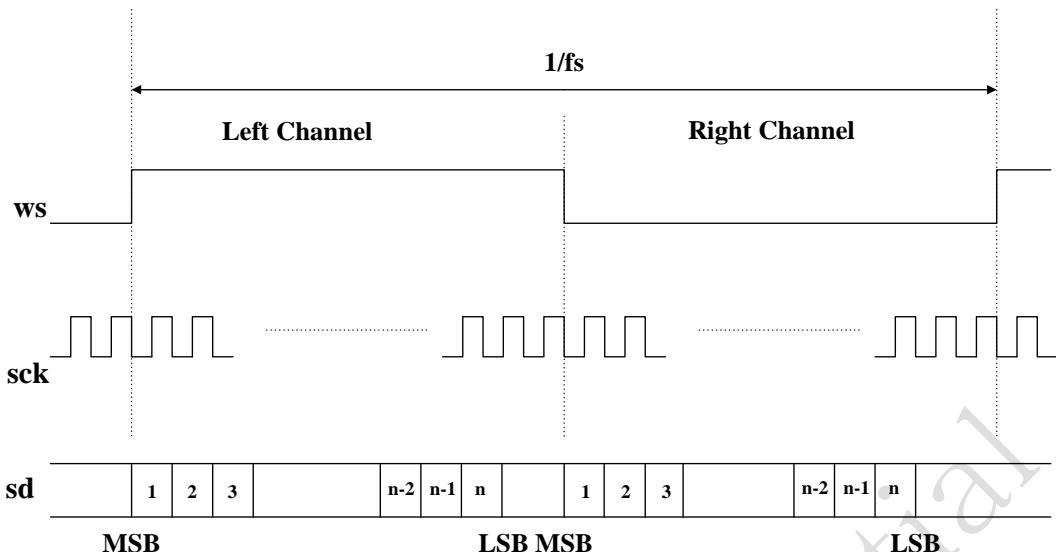
The Codec has the I2S PCM interface of audio data stream in for DAC and out for ADC, both of which can be configured in master or slave mode. Different audio data formats are available for different operating modes, which are demonstrated in below table.

Table 29-1 Supported Data Formats in Different Modes

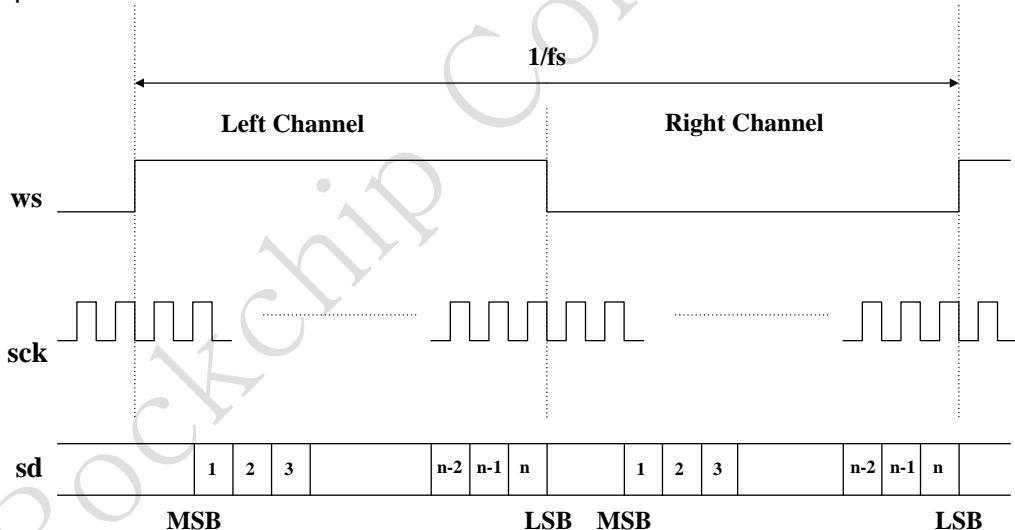
Data Formats	ADC		DAC	
	Master	Slave	Master	Slave
Left Justified	✓	✗	✓	✓
Right Justified	✓	✓	✓	✓
I ² S	✓	✓	✓	✓
DSP/PCM mode A	✓	✓	✓	✓
DSP/PCM mode B	✓	✗	✓	✓

I2S_PCM interface supports five audio data formats: Left Justified mode, Right Justified mode, I²S mode, DSP/PCM mode A and mode B. They are valid when the device operates as a master or slave.

For Left Justified mode, the data format is illustrated in Fig. 29-2. The MSB is valid at the first rising edge of sck after ws transition is done. The other valid bits up to the LSB are transmitted sequentially. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may appear before every ws transition, which means the data in this period is invalid.

Fig.29-2 Left Justified Mode (assuming n -bit word length)

For Right Justified mode, the data format is shown in Fig. 29-3. The LSB becomes valid at the last rising edge of sck before ws transition is done. As the MSB is transmitted first, the other valid bits up to the MSB are followed in order. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may exist after every ws transition, which means the data in this period is invalid.

Fig.29-3 Right Justified Mode (assuming n -bit word length)

For I²S mode, the data format is depicted in Fig. 29-4. The MSB becomes available at the second rising edge of sck when ws transition is done. The other valid bits up to the LSB are transmitted in order. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may appear between the LSB of the current sample and the MSB of the next one, which means the data in this period can be ignored.

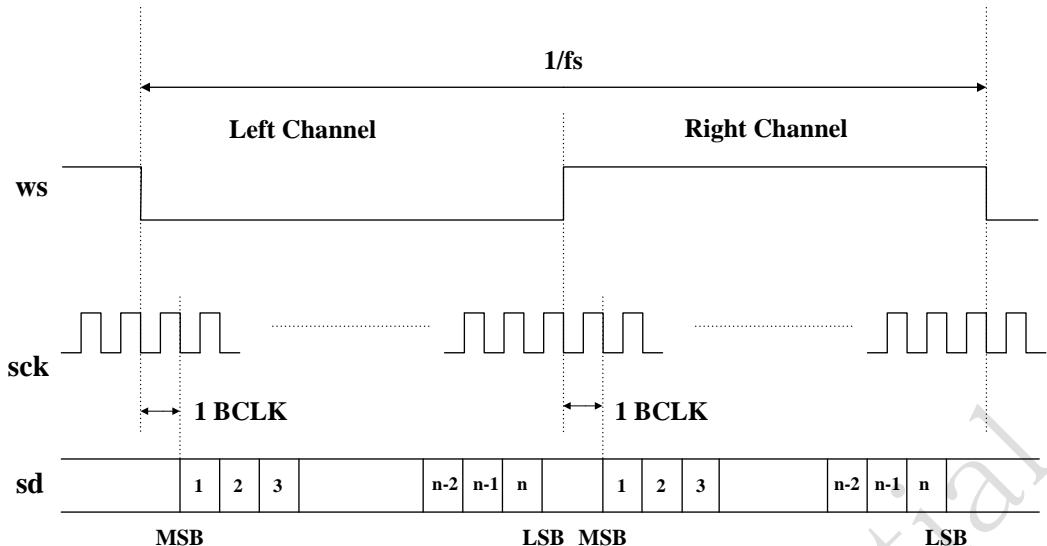


Fig.29-4 I2S Mode (assuming n-bit word length)

For DSP/PCM mode, the left channel data is transmitted first, followed by right channel data. For DSP/PCM mode A/B, the MSB is available at the second and first rising edge of sck after the rising edge of ws respectively, as shown in Fig. 29-5 and Fig. 29-6. Based on word length, sck frequency and sample rate, there may be some invalid data between the LSB of the right channel data and the next sample.

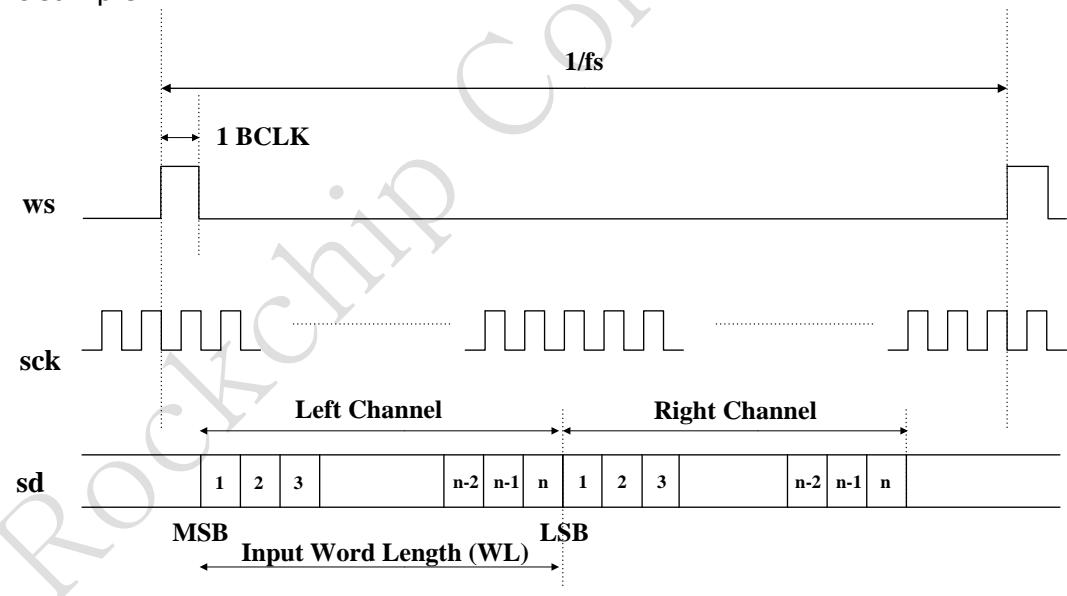


Fig.29-5 DSP/PCM Mode A (assuming n-bit word length)

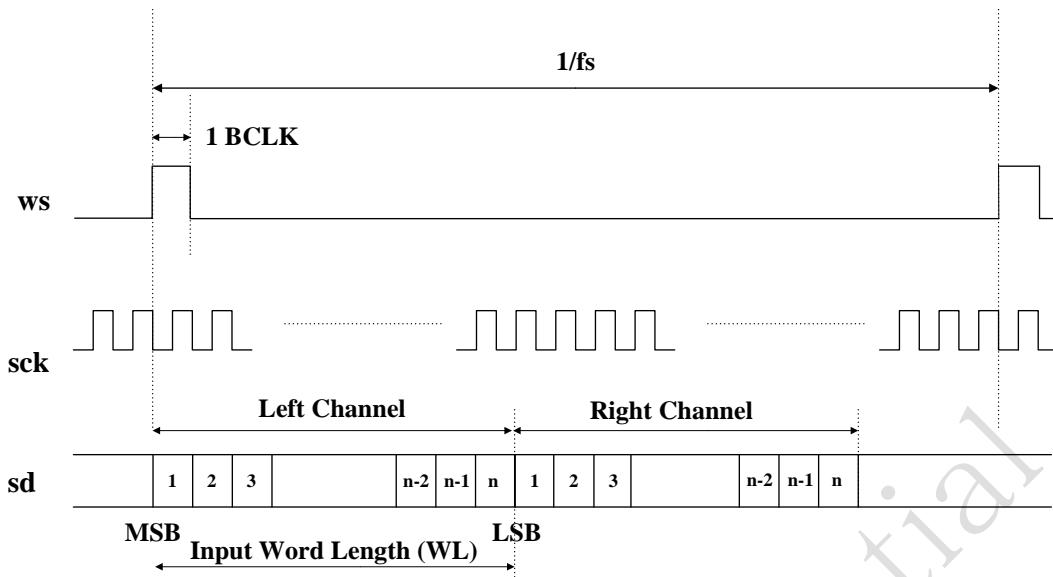


Fig.29-6 DSP/PCM Mode B (assuming n-bit word length)

29.4.2 Analog Interface

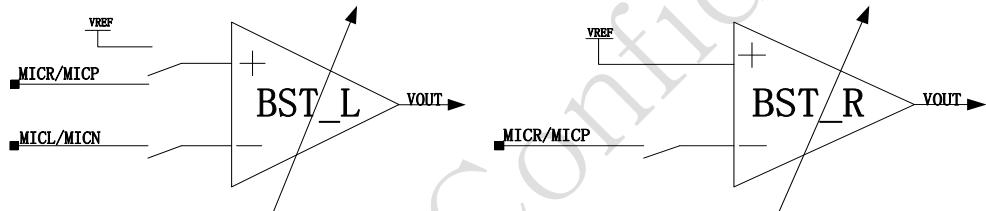


Fig.29-7MicroPhone Input

There are two microphone input channels, left and right channel. In left channel, there are two differential inputs, and they can be configured as either single-ended input or differential inputs by the microphone PGA (BST_L). In right channel, there is only one input, and it is configured as single-ended input by the microphone PGA (BST_R).

In left channel, microphone inputs are MICL and MICR. When working in single-ended configuration, the input signal should be input through MICL. In right channel, microphone input is MICR.

Microphone PGA has two gains to amplify the input signal, that is, 0dB and +20dB.

There are two line input channels, INL and INR. They are input to left and right channel MUX (MUXIN_L and MUXIN_R), respectively. In each channel, the input MUX can choose line input or microphone PGA output as the input of ALC PGA. Automatic Level Control (ALC) function is included to adjust the signal level, which is input into ADC. ALC will measure the signal magnitude and compare it to defined threshold. Then it will adjust the ALC controlled PAG (ALC_L and ALC_R) gain according to the comparison result.

The programmable gain range of ALC controlled PAG is from -18dB to +28.5dB. The tuning step is 1.5dB.

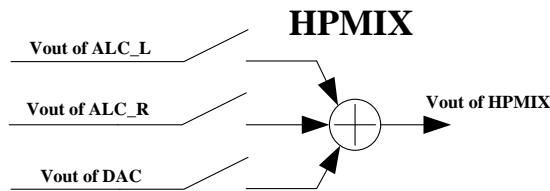


Fig.29-8output mixer

DAC output and ADC input can be mixed by output mixer. There are two output channel mixers, HPMIXL and HPMIXR.

In HPMIXL mixer, output of left channel DAC, output of left channel ALC PGA and output of right channel ALC PGA can be mixed. In HPMIXR mixer, output of right channel DAC, output of left channel ALC PGA and output of right channel ALC PGA can be mixed.

This Codec supports two headphone output configurations. The headphone output can drive 16Ω or 32Ω headphone load either through DC-blocking capacitor or DC-coupled capless configuration.

In the configuration using DC-blocking capacitor, shown in following figure, the headphone ground is connected to the real ground. The capacitance and the load resistance determine the lower cut-off frequency. For instance, if 16Ω headphone and 100uF DC-blocking capacitor are used, the lower cut-off frequency is:

$$f = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 16 \times 100 \times 10^{-6}} = 99.5Hz$$

The DC-blocking capacitor can be increased to lower the cut-off frequency for better bass response.

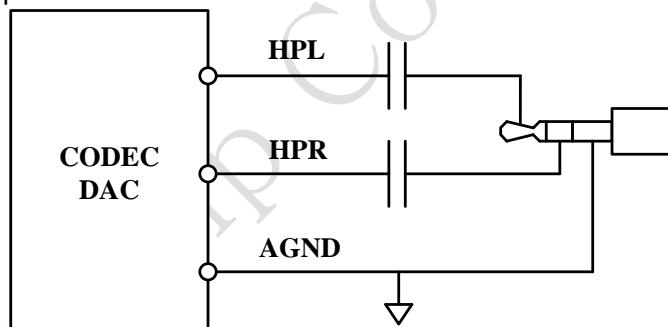


Fig.29-9DC-blocking capacitor

In the DC-coupled capless configuration, shown in following figure, the headphone ground is connected to a virtual ground, AOM. AOM is a DC output driver with a DC voltage of AVDD/2, that is, half of the analog supply. The requirement for DC-blocking capacitor is removed, which can save the cost.

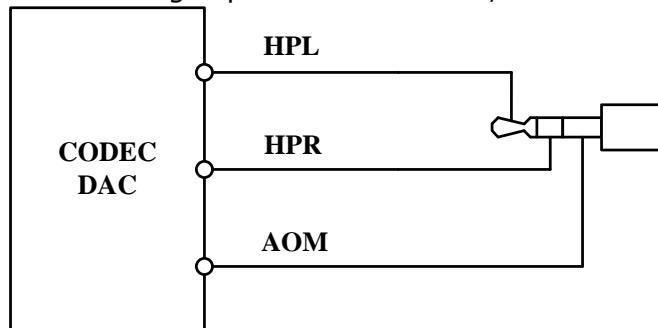


Fig.29-10DC-coupled capless

The headphone driver can choose mixer output or DAC output as input. It has a gain range from -39dB to +6dB with a tuning step of 1.5dB.

Microphone bias output is used to bias external microphones. The bias voltage can varies from $0.5 \times \text{AVDD}$ to $0.85 \times \text{AVDD}$ with a step of $0.05 \times \text{AVDD}$.

29.4.3 Interface Relationship

In broadcasting application, the I2S/PCM1/2 controller is used as a transmitter and audio CODEC is used as a receiver. In recording application, the I2S/PCM1/2 controller is used as a receiver and audio CODEC is used as a transmitter. Either the I2S/PCM1/2 controller or the audio CODEC can act as a master or a slave, but if one is master, the other must be slave.

Fig.29-11 and Fig.29-12 illustrate the relationship between I2S interface and the parallel audio data in ADC and DAC channels.

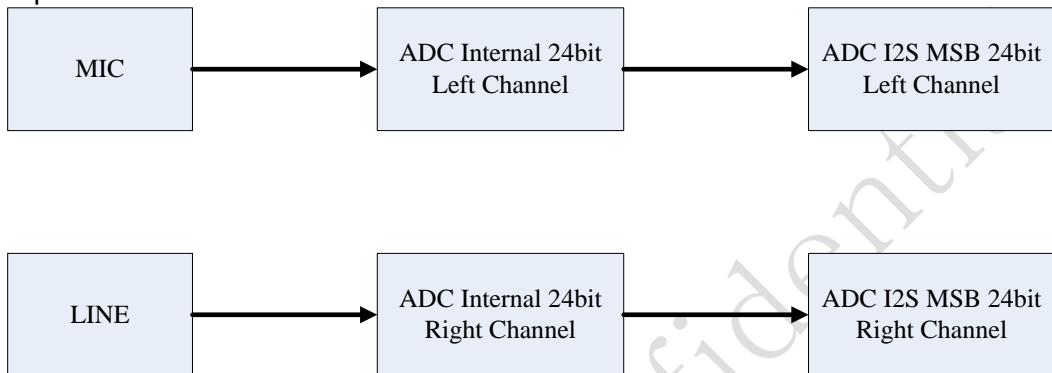


Fig.29-8 ADC Channels Relationship

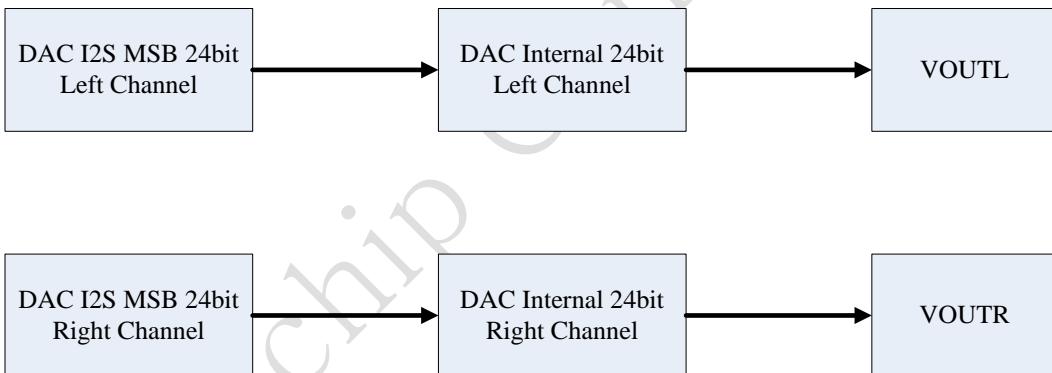


Fig.29-9 DAC Channels Relationship

29.5 Register description

29.5.1 Register Summary

Name	Offset	Size	Reset Value	Description
Codec_REG0	0x0000	W	0x00000003	Codec register 0
Codec_REG2	0x0008	W	0x000000050	Codec register 2
Codec_REG3	0x000c	W	0x0000000e	Codec register 3
Codec_REG4	0x0010	W	0x000000050	Codec register 4
Codec_REG5	0x0014	W	0x0000000e	Codec register 5
Codec_REG34	0x0088	W	0x000000000	Codec register 34
Codec_REG35	0x008c	W	0x000000000	Codec register 35
Codec_REG36	0x0090	W	0x000000044	Codec register 36

Name	Offset	Size	Reset Value	Description
Codec_REG37	0x0094	W	0x0000000c	Codec register 37
Codec_REG38	0x0098	W	0x0000000c	Codec register 38
Codec_REG39	0x009c	W	0x00000000	Codec register 39
Codec_REG40	0x00a0	W	0x00000000	Codec register 40
Codec_REG41	0x00a4	W	0x00000000	Codec register 41
Codec_REG42	0x00a8	W	0x00000000	Codec register 42
Codec_REG43	0x00ac	W	0x00000000	Codec register 43
Codec_REG44	0x00b0	W	0x00000000	Codec register 44
Codec_REG45	0x00b4	W	0x00000000	Codec register 45
Codec_REG46	0x00b8	W	0x00000000	Codec register 46
Codec_REG47	0x00bc	W	0x0000001e	Codec register 47
Codec_REG64	0x00c0	W	0x00000000	Codec register 64
Codec_REG65	0x00c4	W	0x00000046	Codec register 65
Codec_REG66	0x00c8	W	0x00000041	Codec register 66
Codec_REG67	0x00cc	W	0x00000002c	Codec register 67
Codec_REG68	0x00d0	W	0x00000000	Codec register 68
Codec_REG69	0x00d4	W	0x000000026	Codec register 69
Codec_REG70	0x00d8	W	0x000000040	Codec register 70
Codec_REG71	0x00dc	W	0x000000036	Codec register 71
Codec_REG72	0x00e0	W	0x000000020	Codec register 72
Codec_REG73	0x00e4	W	0x000000038	Codec register 73

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

29.5.2 Detail Register Description

Codec_REG0

Address: Operational Base + offset (0x0000)

Codec register 0

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	Power reset bypass 0: not 1: bypass
5:2	RO	0x0	reserved
1	RW	0x1	Codec digital core reset This reset only reset the codec data path. 0: reset 1: work
0	RW	0x1	Codec system reset This signal will reset the registers which control all the digital and analog part. 0: reset 1: work

Codec_REG2

Address: Operational Base + offset (0x0008)

Codec register 2

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	ADC LRC Polarity 0: normal 0: reversal
6:5	RW	0x2	ADC Valid Word Length in one 1/2Frame 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits
4:3	RW	0x2	ADC mode 11: PCM Mode 10: I2S Mode 01: Left Justified Mode 00: Right Justified Mode Note. Same word length in 1/2frame and valid data is not supported in Right Justified Mode. For example, 32/24 or 24/20 is supported, but 32/32 or 24/24 is not supported. (1/2frame length/valid data length)
2	RO	0x0	reserved
1	RW	0x0	ADC Left-Right SWAP 0: normal 1: swap
0	RW	0x0	ADC type 1: Mono 0: Stereo

Codec_REG3

Address: Operational Base + offset (0x000c)

Codec register 3

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	ADC and DAC I2S Mode Select 0: slave mode 1: master mode
3:2	RW	0x3	ADC 1/2Frame Word Length 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits
1	RW	0x1	ADC Reset 0: reset 1: work
0	RW	0x0	ADC Bit Clock Polarity 0: normal 1: reversal

Codec_REG4

Address: Operational Base + offset (0x0010)

Codec register 4

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	DAC LRC Polarity 0: normal 1: reversal
6:5	RW	0x2	DAC Valid Word Length in one 1/2Frame 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits
4:3	RW	0x2	DAC mode 11: PCM Mode 10: I2S Mode 01: Left Justified Mode 00: Right Justified Mode Note. Same word length in 1/2frame and valid data is not supported in Right Justified Mode. For example, 32/24 or 24/20 is supported, but 32/32 or 24/24 is not supported. (1/2frame length/valid data length)
2	RW	0x0	DAC Left-Right SWAP 0: normal 1: swap
1:0	RO	0x0	reserved

Codec_REG5

Address: Operational Base + offset (0x0014)

Codec register 5

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:2	RW	0x3	DAC 1/2Frame Word Length 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits
1	RW	0x1	DAC reset 0: reset 1: work
0	RW	0x0	DAC Bit Clock polarity 0: normal 1: reversal

Codec_REG34

Address: Operational Base + offset (0x0088)

Codec register 34

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	The enable signal of current source for ADC 0: Stop Working 1: Work

Bit	Attr	Reset Value	Description
6	RW	0x0	The enable signal of MIC bias voltage (MICBIAS) buffer 0: Stop Working 1: Work
5	RW	0x0	The enable signal of the ADCL input zero-crossing detection module: 0: Stop Working ,output 0 electrical level 1: Work
4	RW	0x0	The enable signal of the ADCR input zero-crossing detection module: 0: Stop Working ,output 0 electrical level 1: Work
3	RO	0x0	reserved
2:0	RW	0x0	The level range control signal of MIC bias voltage (MICBIAS) 000 .0*VREF 111 .7*VREF Step .1*VREF VREF = AVDD/2

Codec_REG35

Address: Operational Base + offset (0x008c)

Codec register 35

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	The enable signal of BST_L module 0: Stop Working 1: Work
5	RW	0x0	The gain control of BST_L module 1: 20dB 0: 0dB
4	RW	0x0	The mute signal of BST_L module 0: Mute 1: Work
3	RO	0x0	reserved
2	RW	0x0	The enable signal of BST_R module 0: Stop Working 1: Work
1	RW	0x0	The gain control of BST_R module 1: 20dB 0: 0dB
0	RW	0x0	The mute signal of BST_R module 0: Mute 1: Work

Codec_REG36

Address: Operational Base + offset (0x0090)

Codec register 36

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x1	The signal to select input of MUXINL module 00/11: don't select 10: select INL 01: select output of BST_L module
5	RW	0x0	The enable signal of ALCL module 0: Stop Working 1: Work
4	RW	0x0	The mute signal of ALCL module 0: Mute 1: Work
3:2	RW	0x1	The signal to select input of MUXINR module 00/11:don't select 10:Select INR 01:Select output of BST_R module
1	RW	0x0	The enable signal of ALCR module 0: Stop Working 1: Work
0	RW	0x0	The mute signal of ALCR module 0: Mute 1: Work

Codec_REG37

Address: Operational Base + offset (0x0094)

Codec register 37

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	The signal to select the input mode of BST_L module 1: Single-ended input 0: Full differential input
4:0	RW	0x0c	The gain control of ALC_L module 00000: -18dB 01100: 0dB 11111 : 28.5dB Step: 1.5dB

Codec_REG38

Address: Operational Base + offset (0x0098)

Codec register 38

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x0c	The gain control of ALC_R module 00000: -18dB 01100: 0dB 11111: 28.5dB Step: 1.5dB

Codec_REG39

Address: Operational Base + offset (0x009c)

Codec register 39

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	The enable signal of reference Voltage buffer for left ADC PATH 0: Stop working 1: Work
6	RW	0x0	The enable signal of CLOCK for ADCL module 0: Set to logic "1" 1: Work
5	RW	0x0	The enable signal of Amplifier in ADCL module 0: Stop working all amplifier 1: Work
4	RW	0x0	The reset signal of different levels integrator in ADCL module 0: Work 1: Clear
3	RW	0x0	The enable signal of reference Voltage buffer for right ADC PATH 0: Stop working 1: Work
2	RW	0x0	The enable signal of CLOCK for ADCR module 0: Set to logic "1" 1: Work
1	RW	0x0	The enable signal of Amplifier in ADCR module 0: Stop working all amplifier 1: Work
0	RW	0x0	The reset signal of different levels integrator in ADCR module 0: Work 1: Clear

Codec_REG40

Address: Operational Base + offset (0x00a0)

Codec register 40

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	The enable signal of current source for CODEC DAC 0: Stop Working 1: Work
5	RW	0x0	The enable signal of reference Voltage buffer for left DAC PATH 0: Stop Working 1: Work
4	RW	0x0	The enable signal of zero-crossing detection module for VOUTL: 0: Stop Working ,output 0 electrical level 1: Work

Bit	Attr	Reset Value	Description
3	RW	0x0	The enable signal of module to detect earphone 0: Don't detect, output logic "0" 1: Detect
2	RO	0x0	reserved
1	RW	0x0	The enable signal of reference Voltage buffer for right DAC PATH 0: Stop working 1: Work
0	RW	0x0	The enable signal of zero-crossing detection module for VOUTR: 0: Stop Working ,output 0 electrical level 1: Work

Codec_REG41

Address: Operational Base + offset (0x00a4)

Codec register 41

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	The enable signal of high and low reference Voltage buffer for DACL module 0: stop working 1: Work
6	RW	0x0	The enable signal of CLOCK module for DACL 0: Set CLOCK to logic "1" 1: Work
5	RW	0x0	The enable signal of DACL module 0: Stop work 1: Work
4	RW	0x0	The Initial signal of DACL module 0: Initialization 1: Work
3	RW	0x0	The enable signal of high and low reference Voltage buffer for DACL module 0: Stop working 1: Work
2	RW	0x0	The power down signal of the ADCR input zero-crossing detection: 1: Power down ,output 0 electrical level 0: Work
1	RW	0x0	The enable signal of DACR module 0: Stop work 1: Work
0	RW	0x0	The Initial signal of DACR module 0: Initialization 1: Work

Codec_REG42

Address: Operational Base + offset (0x00a8)

Codec register 42

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	The enable signal of HPMIXL module 0: Stop work 1: Work
5	RW	0x0	The Initial1 signal of HPMIXL module 0: Initialization 1: Work
4	RW	0x0	The Initial2 signal of HPMIXL module 0: Initialization 1: Work
3	RO	0x0	reserved
2	RW	0x0	The enable signal of HPMIXR module 0: Stop work 1: Work
1	RW	0x0	The Initial1 signal of HPMIXR module 0: Initialization 1: Work
0	RW	0x0	The Initial2 signal of HPMIXR module 0: Initialization 1: Work

Codec_REG43

Address: Operational Base + offset (0x00ac)

Codec register 43

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	The bypass signal of HPMIXL module 1: Bypass HPMIXL 0: Don't bypass
6:4	RW	0x0	The signal to select input of HPMIXL module: [6]: select output of ALCL module 0: Don't select 1: Select [5]: select output of ALCR module 0: Don't select 1: Select [4]: select output of DACL module 0: Don't select 1: Select
3	RW	0x0	The bypass signal of HPMIXR module 1: Bypass HPMIXR 0: Don't bypass

Bit	Attr	Reset Value	Description
2:0	RW	0x0	The signal to select input of HPMIXR module: [2]: select output of ALCL module 0: Don't select 1: Select [1]: select output of ALCR module 0: Don't select 1: Select [0]: select output of DACR module 0: Don't select 1: Select

Codec_REG44

Address: Operational Base + offset (0x00b0)

Codec register 44

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	The enable signal of HPOUTL module 0: Stop working 1: Work
6	RW	0x0	The Initial signal of HPOUTL module 0: Initialization 1: Work
5	RW	0x0	The mute signal of HPOUTL module 0: Mute 1: Work
4	RW	0x0	The enable signal of HPOUTR module 0: Stop working 1: Work
3	RW	0x0	The Initial signal of HPOUTR module 0: Initialization 1: Work
2	RW	0x0	The mute signal of HPOUTR module 0: MUTE 1: Work
1	RW	0x0	The enable signal of HPVREF module 0: Stop working 1: Work
0	RW	0x0	The Initial signal of HPVREF module 0: Initialization 1: Work

Codec_REG45

Address: Operational Base + offset (0x00b4)

Codec register 45

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	The signal to select gain of HPOUTL module: 00000: -39dB 11010: 6dB 11111: 0dB Step: 1.5dB

Codec_REG46

Address: Operational Base + offset (0x00b8)

Codec register 46

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	The signal to select gain of HPOUTR module: 00000: -39dB 11010: 6dB 11111: 0dB Step: 1.5dB

Codec_REG47

Address: Operational Base + offset (0x00bc)

Codec register 47

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	The signal to select current to Precharge/Discharge [4]: Select 10uA 0: Select 1: Don't select [3]: Select current I0 0: Select 1: Don't select [2]: Select current 2*I0 0: Select 1: Don't select [1]: Select current 4*I0 0: Select 1: Don't select [0]: Select current 4*I0 0: Select 1: Don't select I0 is a reference current, the current can stack
4:0	RW	0x1e	Field0000 Abstract Field0000 Description

Codec_REG64

Address: Operational Base + offset (0x00c0)

Codec register 64

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	Choose the method to control the gain attack 0 : Normal way 1 : Jack way
5:4	RW	0x0	There are four methods to generate the control signals 00: Normal way 01: Jack way 1 10: Jack way 2 11: Jack way 3 This register is used to according to the physical truth to choose the method to generate the control signals.
3:0	RW	0x0	AGC hold time before gain is increased in normal mode. 0000: 0ms 0001: 2ms 0010: 4ms 0011: 8ms 0100: 16ms 0101: 32ms 0110: 64ms 0111: 128ms 1000: 256ms 1001: 512ms 1010: 1 s 1011~1111: 0ms

Codec_REG65

Address: Operational Base + offset (0x00c4)

Codec register 65

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x4	Decay (gain ramp-up) time Normal MODE(reg_agc_mde = 0) 0000: 500us 0001: 1ms 0010: 2ms 0011: 4ms 0100: 8ms 0101: 16ms 0110: 32ms 0111: 64ms 1000: 128ms 1001: 256ms 1010: 512ms 1001~1111: 512ms

Bit	Attr	Reset Value	Description
3:0	RW	0x6	<p>Attack (gain ramp-down) Time Noraml MODE(reg_agc_mde =0)</p> <p>0000:125us 0001:250us 0010:500us 0011:1ms 0100:2ms 0101:4ms 0110:8ms 0111:16ms 1000:32ms 1001:64ms 1010:128ms 1011~1111:125us Noraml MODE(reg_agc_mde =1)</p> <p>0000:32us 0001:64us 0010:125us 0011:250us 0100:500us 0101:1ms 0110:2ms 0111:4ms 1000:8ms 1001:16ms 1010:32ms 1011~1111:32us</p>

Codec_REG66

Address: Operational Base + offset (0x00c8)

Codec register 66

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	Determines the AGC mode of operation 0:AGC mode(normal mode) 1:Limiter mode
6	RW	0x1	AGC users zero cross enable 0:Disabled 1:Enabled, the AGC gain will update at zero cross enable
5	RW	0x0	When in the limiter mode, the low amplitude signal will recovery in two modes: 0:The gain will recovery to the value of the reg_pga_lvol 1:The gain will recovery to the gain at the moment that the mode changes from AGC to Limiter.
4	RW	0x0	When the amplitude of the signal is more than 87.5% of the Full scale, use this signal to control the fast decrement: 0:Disabled 1:Enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	AGC noise gate function enable 0:Disabled 1:Enabled
2:0	RW	0x1	AGC noise gate threshold 000:-39dB 001:-45dB 010:-51dB 011:-57dB 100:-63dB 101:-69dB 110:-75dB 111:-81dB

Codec_REG67

Address: Operational Base + offset (0x00cc)

Codec register 67

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x1	Left channel input PGA zero cross enable 0:Update gain when gain register changes. 1:Update gain on 1st zero cross after gain register write.
4:0	RW	0x0c	Left channel input PGA gain 00000:-18dB 00001:-16.5dB 00010:-15dB n: (1.5xn-18)dB 01100:0dB 01101:+1.5dB 01110:+3dB 11111:+28.5dB

Codec_REG68

Address: Operational Base + offset (0x00d0)

Codec register 68

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	Slow clock enabled used for the zero cross timeout.

Bit	Attr	Reset Value	Description
2:0	RW	0x0	Approximate sample rate 000:96kHz 001:48kHz 010:44.1kHz 011:32kHz 100:24kHz 101:16kHz 110:12kHz 111: 8kHz 110~111:reserved

Codec_REG69

Address: Operational Base + offset (0x00d4)

Codec register 69

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x26	agc_max_l The low 8 bits of the AGC maximum level

Codec_REG70

Address: Operational Base + offset (0x00d8)

Codec register 70

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x40	agc_max_h The high 8 bits of the AGC maximum level

Codec_REG71

Address: Operational Base + offset (0x00dc)

Codec register 71

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x36	agc_min_l The low 8 bits of the AGC minimum level

Codec_REG72

Address: Operational Base + offset (0x00e0)

Codec register 72

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x20	agc_min_h The high 8 bits of the AGC minimum level

Codec_REG73

Address: Operational Base + offset (0x00e4)

Codec register 73

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	agcf AGC function select 0: AGC function off 1: AGC function enable
5:3	RW	0x7	pga_maxg Set maximum gain of PGA 000:-13.5dB 001:- 7.5dB 010:- 1.5dB 011:+ 4.5dB 100:+10.5dB 101:+16.5dB 110:+22.5dB 111:+28.5dB
2:0	RW	0x0	pga_ming Set minimum gain of PGA 000:-18dB 001:-12dB 010:- 6dB 011: 0dB 100:+ 6dB 101:+12dB 110:+18dB 111:+24dB

29.6 Application Note

Chapter 30 Global Positioning System (GPS)

30.1 Overview

The GPS is a high-performance baseband device which has an AHB master interface and an AHB slave interface.

CPU can access GPS registers through the AHB slave interface.

The GPS has a 32-channel DMA inside, which can read/write data to system memory through the AHB master interface.

The GPS supports following features:

- Single chip, integrate GPS baseband with CPU
- 32 DMA channels for AHB master access
- Complete L1-band, C/A, and NMEA-0183 compatibility
- Support reference frequency 16.368MHz
- High sensitivity for indoor fixes
- Low power consumption
- Low cost with smaller size
- Multi modes support both standalone GPS and A_GPS

30.2 Block Diagram

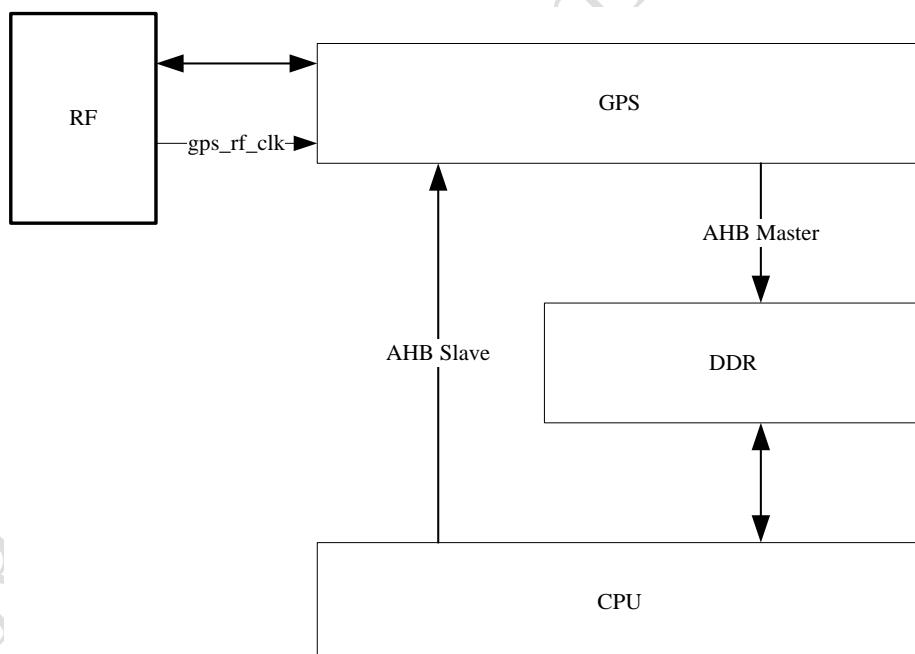


Fig. 30-1 RKaudi GPS block diagram

As shown in Fig. 35-1, the GPS controller should be connected to RF chip through GPIO. The RF chip has 3 1-bit signals output to GPS controller inside chip. They are GPS_RF_CLK, GPS_SIG, GPS_MAG.

The CPU can start RF chip through GPIO, after GPS configuring completion.

30.3 Register Summary

The base address of GPS is 0x10400000.

30.3.1 Base band register summary

Table 39-1 RKaudi GPS base band register summary

Name	Offset	Size	Reset Value	Description
BB_CTRL	0x400	W	0x0	base band control register
BB_START_ADR	0x404	W	0x0	correlator start address register
BB_DS_PARAMETER	0x408	W	0x0	down-sample control register
GPS_INT_ENA	0x40c	W	0x0	interrupt enable register
GPS_INT_STATUS	0x410	W	0x0	interrupt status register
BB_CHN_COR_STATUS	0x414	W	0x0	channel correlator status register
BB_CHN_COR_VALID	0x418	W	0x0	channel correlator valid register
BB_RF_TIMER_VAL	0x41c	W	0x0	RF clock timer value register
BB_RF_WT_ADDR	0x420	W	0x1fffffff	RF FIFO write address register
CH_MISCx	x<<5+0	W	N/A	control information
CH_CAR_NCOx	x<<5+4	W	N/A	carrier NCO
CH_CODE_NCOx	x<<5+8	W	N/A	code NCO
CH_DMA_ADDRx	x<<5+12	W	N/A	dma address
CH_CAR_FREQx	x<<5+16	W	N/A	carrier frequency
CH_CODE_FREQx	x<<5+20	W	N/A	carrier frequency

Note: x is for the channel number from 0-31

30.3.2 ACC operation register summary

Table 39-2 RKaudi GPS acc register summary

Name	Offset	Size	Reset Value	Description
ACC_CTRL	0x8000	W	0x0	acc control register
DMA_CH0_START_ADDR	0x8004	W	0xffffffff	channel 0 dma start address
DMA_CH1_START_ADDR	0x800c	W	0xaaaaaaaa	channel 1 dma start address
DMA_CH2_START_ADDR	0x8014	W	0x55555555	channel 2 dma start address
FFT_VLD_NUM_START_POINT	0x801c	W	0x0	FFT_START_POINT, FFT_VALID_NUM
FFT_COR_NUM_LOOP_NUM	0x8020	W	0x0	FFT_LOOP_NUM, TOT_EPD_COR_NUM, EPD_START_COR_NUM
EPD_MAXCORID	0x8024	W	0x0	EPD_MAXCORID, MAX_FREQINDEX
EPD_MAX_DATA_MANTISSA	0x8028	W	0x0	the maximum energy mantissa
EPD_MAX_DATA_EXPONENT	0x802c	W	0x0	the exponent of the maximum energy
EPD_MAX_RIGHT_MANTISSA	0x8030	W	0x0	the mantissa of the right data to the max energy
EPD_MAX_RIGHT_EXPONENT	0x8034	W	0x0	the exponent of the right data to the max energy

Name	Offset	Size	Reset Value	Description
EPD_MAX_LEFT_M AN	0x8038	W	0x0	the mantissa of the left data to the max energy
EPD_MAX_LEFT_E XP	0x803c	W	0x0	the exponent of the left data to the max energy
EPD_L0_DATA_MA N	0x8040	W	0x0	the mantissa of the max energy of the cor0 data
EPD_L0_DATA_EXP	0x8044	W	0x0	the exponent of the max energy of the cor0 data
EPD_L1_DATA_MA N	0x8048	W	0x0	the mantissa of the max energy of the cor1 data
EPD_L1_DATA_EXP	0x804c	W	0x0	the exponent of the max energy of the cor1 data
EPD_L2_DATA_MA N	0x8050	W	0x0	the mantissa of the max energy of the cor2 data
EPD_L2_DATA_EXP	0x8054	W	0x0	the exponent of the max energy of the cor2 data
EPD_L3_DATA_MA N	0x8058	W	0x0	the mantissa of the max energy of the cor3 data
EPD_L3_DATA_EXP	0x805c	W	0x0	the exponent of the max energy of the cor3 data
EPD_L4_DATA_MA N	0x8060	W	0x0	the mantissa of the max energy of the cor4 data
EPD_L4_DATA_EXP	0x8064	W	0x0	the exponent of the max energy of the cor4 data
EPD_L5_DATA_MA N	0x8068	W	0x0	the mantissa of the max energy of the cor5 data
EPD_L5_DATA_EXP	0x806c	W	0x0	the exponent of the max energy of the cor5 data
EPD_L6_DATA_MA N	0x8070	W	0x0	the mantissa of the max energy of the cor6 data
EPD_L6_DATA_EXP	0x8074	W	0x0	the exponent of the max energy of the cor6 data
EPD_L7_DATA_MA N	0x8078	W	0x0	the mantissa of the max energy of the cor7 data
EPD_L7_DATA_EXP	0x807c	W	0x0	the exponent of the max energy of the cor7 data
EPD_L8_DATA_MA N	0x8080	W	0x0	the mantissa of the max energy of the cor8 data
EPD_L8_DATA_EXP	0x8084	W	0x0	the exponent of the max energy of the cor8 data
EPD_L9_DATA_MA N	0x8088	W	0x0	the mantissa of the max energy of the cor9 data
EPD_L9_DATA_EXP	0x808c	W	0x0	the exponent of the max energy of the cor9 data

30.4 Interface Description

The IOMUX configuration of GPS is shown below:

Table 39-3 RKaudi GPS IOMUX Setting

Module Pin	IO	Pad Name	IOMUX Setting
gps_rfclk	I	IO_LCD0d22_EBCgdwr1_GPSclk_GMACcol_GPI	GPIO2D_IOMUX[14:12]=3'b011

		O2d0	
gps_mag	I	IO_LCD0d21_EBCborder1_GPSmag_GMACtxd3_G PIO2c7	GPIO2C_IOMUX2[14:12]=3'b011
gps_sig	I	IO_LCD0d20_EBCborder0_GPSsign_GMACtxd2_G PIO2c6	GPIO2C_IOMUX2[10:8]=3'b011

Notes: 1. I=input, O=output, I/O=input/output, bidirectional

30.5 Application note

GPS has 2 interrupt output signals: gps_irq and gps_timer_irq. The gps_irq interrupt has a system interrupt ID 44, and gps_timer_irq has a system interrupt ID 45.

Rockchip Confidential

Chapter 31 SPI

31.1 Overview

The serial peripheral interface is an APB slave device. A fourwire fullduplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of slave select signals or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled. This SPI controller can work as either master or slave mode.

31.1.1 Features

SPI Controller supports the following features:

- Support Motorola SPI, TI Synchronous Serial Protocol and National Semiconductor Microwire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bit serial data transfer
- Support configurable interrupt polarity
- Support asynchronous APB bus and SPI clock
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow, interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support up to half of SPI clock frequency transfer in master mode and one sixth of SPI clock frequency transfer in slave mode
- Support full and half duplex mode transfer
- Stop transmitting SCLK if transmit FIFO is empty or receive FIFO is full in master mode
- Support configurable delay from chip select active to SCLK active in master mode
- Support configurable period of chip select inactive between two parallel data in master mode
- Support big and little endian, MSB and LSB first transfer
- Support two 8-bit audio data store together in one 16-bit wide location
- Support sample RXD 0~3 SPI clock cycles later
- Support configurable SCLK polarity and phase
- Support fix and incremental address access to transmit and receive FIFO

31.2 Block Diagram

The SPI Controller comprises with:

- AMBA APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block

- Shift control and interrupt

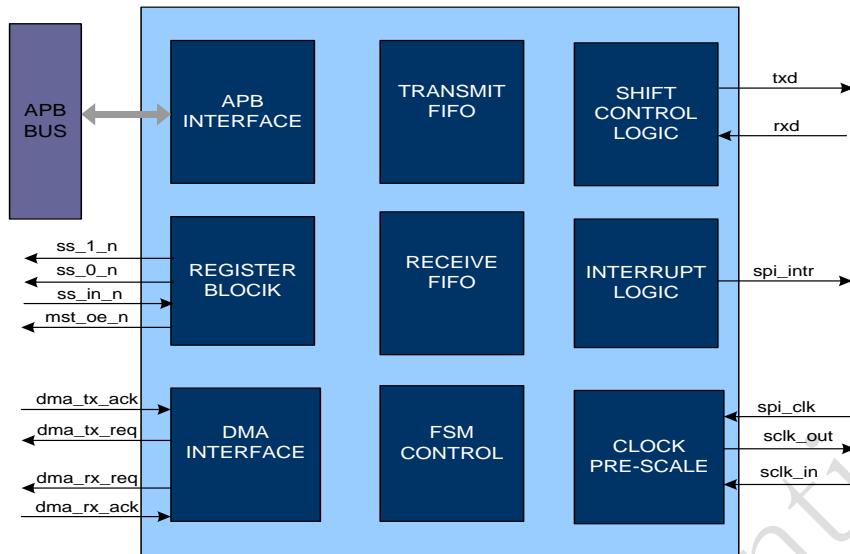


Fig.31-1SPI Controller Block diagram

APB INTERFACE

The host processor accesses data, control, and status information on the SPI through the APB interface. The SPI supports APB data bus widths of 8, 16, and 32 bits.

DMA INTERFACE

This block has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA Controller.

FIFO LOGIC

For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the transmit FIFO. Data received from the external serial device into the SPI is pushed into the receive FIFO. Both fifos are 32x16bits.

FSM CONTROL

Control the state's transformation of the design.

REGISTER BLOCK

All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the AHB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL

Shift control logic shift the data from the transmit fifo or to the receive fifo. This logic automatically right-justifies receive data in the receive FIFO buffer.

INTERRUPT CONTROL

The SPI supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other SPI interrupts after masking.

31.3 Function description

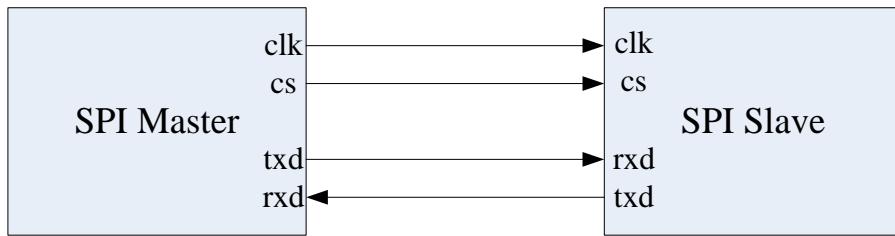


Fig.31-2SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram show how the SPI controller connects with other SPI devices.

Operation Modes

The SPI can be configured in the following two fundamental modes of operation: Master Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

Transfer Modes

The SPI operates in the following three modes when transferring data on the serial bus.

1. Transmit and Receive

When SPI_CTRLR0 [19:18]== 2'b00, both transmit and receive logic are valid.

2. Transmit Only

When SPI_CTRLR0 [19:18] == 2'b01, the receive data are invalid and should not be stored in the receive FIFO.

3. Receive Only

When SPI_CTRLR0 [19:18]== 2'b10, the transmit data are invalid.

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the $F_{spi_clk} \geq 2 \times (\text{maximum } F_{sclk_out})$

When SPI Controller works as slave, the $F_{spi_clk} \geq 6 \times (\text{maximum } F_{sclk_in})$

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4/8/16 bits in length.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. The following two figures show a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

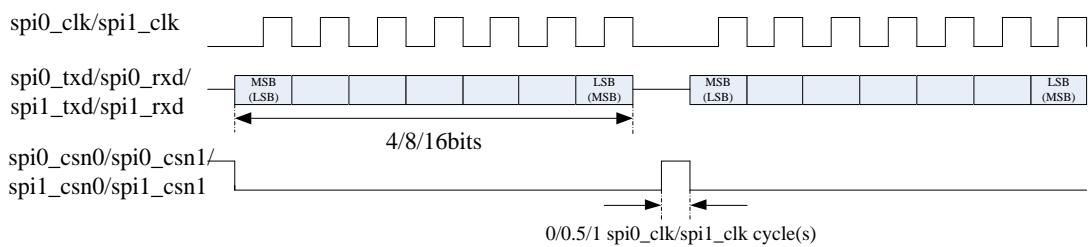


Fig.31-3SPI Format (SCPH=0 SCPOL=0)

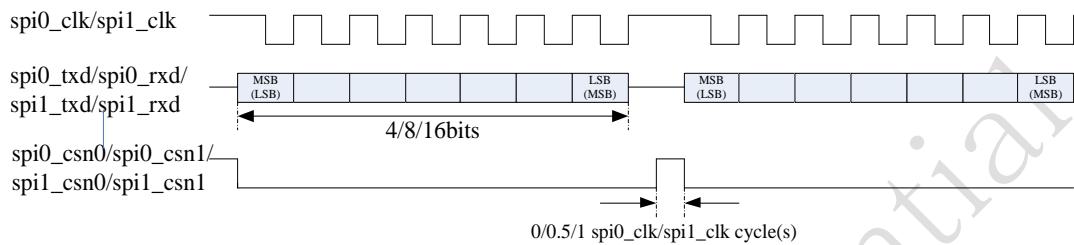


Fig.31-4SPI Format (SCPH=0 SCPOL=1)

When the configuration parameter SCPH = 1, both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data are propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured. The following two figures show the timing diagram for the SPI format when the configuration parameter SCPH = 1.

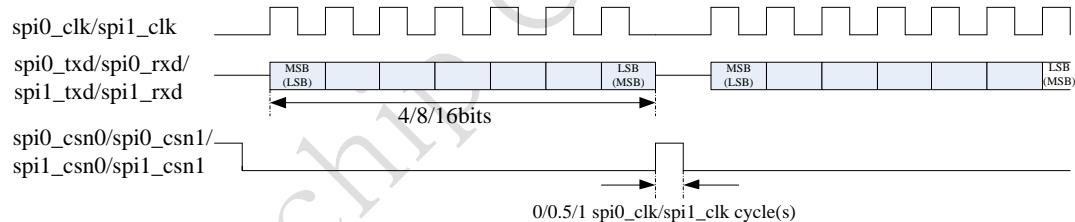


Fig.31-5SPI Format (SCPH=1 SCPOL=0)

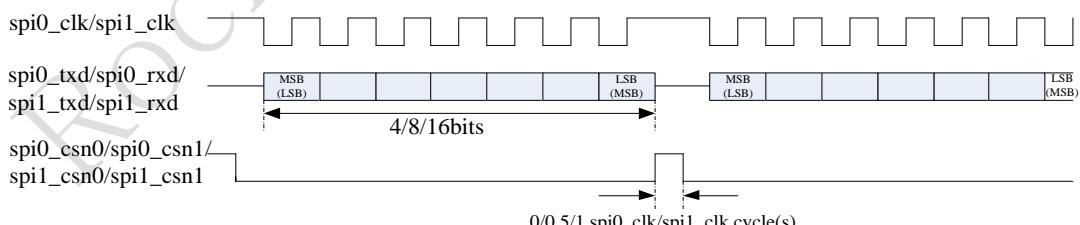


Fig.31-6SPI Format (SCPH=1 SCPOL=1)

31.4 Register Description

This section describes the control/status registers of the design. Pay attention that there are two SPI controllers in the chip: spi0 & spi1, so the base address in the following register descriptions can be either spi0 or spi1 base address.

31.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI_CTRLR0	0x0000	W	0x00000002	Control Register 0
SPI_CTRLR1	0x0004	W	0x00000000	Control Register 1
SPI_ENR	0x0008	W	0x00000000	SPI Enable
SPI_SER	0x000c	W	0x00000000	Slave Enable Register
SPI_BAUDR	0x0010	W	0x00000000	Baud Rate Select
SPI_TXFTLR	0x0014	W	0x00000000	Transmit FIFO Threshold Level
SPI_RXFTLR	0x0018	W	0x00000000	Receive FIFO Threshold Level
SPI_TXFLR	0x001c	W	0x00000000	Transmit FIFO Level
SPI_RXFLR	0x0020	W	0x00000000	Receive FIFO Level
SPI_SR	0x0024	W	0x0000000c	SPI Status
SPI_IPR	0x0028	W	0x00000000	Interrupt Polarity
SPI_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI_ISR	0x0030	W	0x00000000	Interrupt Status
SPI_RISR	0x0034	W	0x00000001	Raw Interrupt Status
SPI_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI_DMACR	0x003c	W	0x00000000	DMA Control
SPI_DMATDLR	0x0040	W	0x00000000	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x00000000	DMA Receive Data Level
SPI_TXDR	0x0400~0x07fc	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x0800~0x0bfc	W	0x00000000	Receive FIFO Data

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

31.4.2 Detail Register Description

SPI_CTRLR0

Address: Operational Base + offset (0x0000)
Control Register 0

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	MTM Microwire Transfer Mode Valid when frame format is set to National Semiconductors Microwire. 1'b0: non-sequential transfer 1'b1: sequential transfer
20	RW	0x0	OPM Operation Mode 1'b0: Master Mode 1'b1: Slave Mode

Bit	Attr	Reset Value	Description
19:18	RW	0x0	XFM Transfer Mode 2'b00 :Transmit & Receive 2'b01 : Transmit Only 2'b10 : Receive Only 2'b11 :reserved
17:16	RW	0x0	FRF Frame Format 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11 : Reserved
15:14	RW	0x0	RSD Rxd Sample Delay When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 2'b00:do not delay 2'b01:1 cycle delay 2'b10:2 cycles delay 2'b11:3 cycles delay
13	RW	0x0	BHT Byte and Halfword Transform Valid when data frame size is 8bit. 1'b0:apb 16bit write/read, spi 8bit write/read 1'b1: apb 8bit write/read, spi 8bit write/read
12	RW	0x0	FBM First Bit Mode 1'b0:first bit is MSB 1'b1:first bit is LSB
11	RW	0x0	EM Endian Mode Serial endian mode can be configured by this bit. Apb endian mode is always little endian. 1'b0:little endian 1'b1:big endian
10	RW	0x0	SSD ss_n to sclk_out delay Valid when the frame format is set to Motorola SPI and SPI used as a master. 1'b0: the period between ss_n active and sclk_out active is half sclk_out cycles. 1'b1: the period between ss_n active and sclk_out active is one sclk_out cycle.

Bit	Attr	Reset Value	Description
9:8	RW	0x0	CSM Chip Select Mode Valid when the frame format is set to Motorola SPI and SPI used as a master. 2'b00: ss_n keep low after every frame data is transferred. 2'b01: ss_n be high for half sclk_out cycles after every frame data is transferred. 2'b10: ss_n be high for one sclk_out cycle after every frame data is transferred. 2'b11: reserved
7	RW	0x0	SCPOL Serial Clock Polarity Valid when the frame format is set to Motorola SPI. 1'b0: Inactive state of serial clock is low 1'b1: Inactive state of serial clock is high
6	RW	0x0	SCPH Serial Clock Phase Valid when the frame format is set to Motorola SPI. 1'b0: Serial clock toggles in middle of first data bit 1'b1: Serial clock toggles at start of first data bit
5:2	RW	0x0	CFS Control Frame Size Selects the length of the control word for the Microwire frame format. 4'b0000~4'b0010: reserved 4'b0011: 4-bit serial data transfer 4'b0100: 5-bit serial data transfer 4'b0101: 6-bit serial data transfer 4'b0110: 7-bit serial data transfer 4'b0111: 8-bit serial data transfer 4'b1000: 9-bit serial data transfer 4'b1001: 10-bit serial data transfer 4'b1010: 11-bit serial data transfer 4'b1011: 12-bit serial data transfer 4'b1100: 13-bit serial data transfer 4'b1101: 14-bit serial data transfer 4'b1110: 15-bit serial data transfer 4'b1111: 16-bit serial data transfer
1:0	RW	0x2	DFS Data Frame Size Selects the data frame length. 2'b00: 4bit data 2'b01: 8bit data 2'b10: 16bit data 2'b11: reserved

SPI_CTRLR1

Address: Operational Base + offset (0x0004)

Control Register 1

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	NDM Number of Data Frames When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.

SPI_ENR

Address: Operational Base + offset (0x0008)

SPI Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	ENR SPI Enable Enables and disables all SPI operations. Transmit and receive FIFO buffers are cleared when the device is disabled.

SPI_SER

Address: Operational Base + offset (0x000c)

Slave Enable Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	SER Slave Select Enable This register is valid only when SPI is configured as a master device.

SPI_BAUDR

Address: Operational Base + offset (0x0010)

Baud Rate Select

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>BAUDR Baud Rate Select SPI Clock Divider.</p> <p>This register is valid only when the SPI is configured as a master device.</p> <p>The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register.</p> <p>If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation:</p> $Fsclk_{out} = F_{spi_clk} / SCKDV$ <p>Where SCKDV is any even value between 2 and 65534.</p> <p>For example: for $F_{spi_clk} = 3.6864\text{MHz}$ and $SCKDV = 2$ $Fsclk_{out} = 3.6864/2 = 1.8432\text{MHz}$</p>

SPI_TXFTLR

Address: Operational Base + offset (0x0014)

Transmit FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>TXFTLR Transmit FIFO Threshold Level</p> <p>When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.</p>

SPI_RXFTLR

Address: Operational Base + offset (0x0018)

Receive FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>RXFTLR Receive FIFO Threshold Level</p> <p>When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.</p>

SPI_TXFLR

Address: Operational Base + offset (0x001c)

Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RO	0x00	TXFLR Transmit FIFO Level Contains the number of valid data entries in the transmit FIFO.

SPI_RXFLR

Address: Operational Base + offset (0x0020)

Receive FIFO Level

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	RXFLR Receive FIFO Level Contains the number of valid data entries in the receive FIFO.

SPI_SR

Address: Operational Base + offset (0x0024)

SPI Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFF Receive FIFO Full 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full
3	RO	0x1	RFE Receive FIFO Empty 1'b0: Receive FIFO is not empty 1'b1: Receive FIFO is empty
2	RO	0x1	TFE Transmit FIFO Empty 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty
1	RO	0x0	TFF Transmit FIFO Full 1'b0: Transmit FIFO is not full 1'b1: Transmit FIFO is full
0	RO	0x0	BSF SPI Busy Flag When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 1'b0: SPI is idle or disabled 1'b1: SPI is actively transferring data

SPI_IPR

Address: Operational Base + offset (0x0028)

Interrupt Polarity

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	IPR Interrupt Polarity Interrupt Polarity Register 1'b0: Active Interrupt Polarity Level is HIGH 1'b1: Active Interrupt Polarity Level is LOW

SPI_IMR

Address: Operational Base + offset (0x002c)

Interrupt Mask

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	RFFIM Receive FIFO Full Interrupt Mask 1'b0: spi_rxf_intr interrupt is masked 1'b1: spi_rxf_intr interrupt is not masked
3	RW	0x0	RFOIM Receive FIFO Overflow Interrupt Mask 1'b0: spi_rxo_intr interrupt is masked 1'b1: spi_rxo_intr interrupt is not masked
2	RW	0x0	RFUIM Receive FIFO Underflow Interrupt Mask 1'b0: spi_rxu_intr interrupt is masked 1'b1: spi_rxu_intr interrupt is not masked
1	RW	0x0	TFOIM Transmit FIFO Overflow Interrupt Mask 1'b0: spi_txo_intr interrupt is masked 1'b1: spi_txo_intr interrupt is not masked
0	RW	0x0	TFEIM Transmit FIFO Empty Interrupt Mask 1'b0: spi_txe_intr interrupt is masked 1'b1: spi_txe_intr interrupt is not masked

SPI_ISR

Address: Operational Base + offset (0x0030)

Interrupt Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFIS Receive FIFO Full Interrupt Status 1'b0: spi_rxf_intr interrupt is not active after masking 1'b1: spi_rxf_intr interrupt is full after masking
3	RO	0x0	RFOIS Receive FIFO Overflow Interrupt Status 1'b0: spi_rxo_intr interrupt is not active after masking 1'b1: spi_rxo_intr interrupt is active after masking

Bit	Attr	Reset Value	Description
2	RO	0x0	RFUIS Receive FIFO Underflow Interrupt Status 1'b0: spi_rxu_intr interrupt is not active after masking 1'b1: spi_rxu_intr interrupt is active after masking
1	RO	0x0	TFOIS Transmit FIFO Overflow Interrupt Status 1'b0: spi_txo_intr interrupt is not active after masking 1'b1: spi_txo_intr interrupt is active after masking
0	RO	0x0	TFEIS Transmit FIFO Empty Interrupt Status 1'b0: spi_txe_intr interrupt is not active after masking 1'b1: spi_txe_intr interrupt is active after masking

SPI_RISR

Address: Operational Base + offset (0x0034)

Raw Interrupt Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFRIS Receive FIFO Full Raw Interrupt Status 1'b0: spi_rxf_intr interrupt is not active prior to masking 1'b1: spi_rxf_intr interrupt is full prior to masking
3	RO	0x0	RFORIS Receive FIFO Overflow Raw Interrupt Status 1'b0: spi_rxo_intr interrupt is not active prior to masking 1'b1: spi_rxo_intr interrupt is active prior to masking
2	RO	0x0	RFURIS Receive FIFO Underflow Raw Interrupt Status 1'b0: spi_rxu_intr interrupt is not active prior to masking 1'b1: spi_rxu_intr interrupt is active prior to masking
1	RO	0x0	TFORIS Transmit FIFO Overflow Raw Interrupt Status 1'b0: spi_txo_intr interrupt is not active prior to masking 1'b1: spi_txo_intr interrupt is active prior to masking

Bit	Attr	Reset Value	Description
0	RO	0x1	TFERIS Transmit FIFO Empty Raw Interrupt Status 1'b0: spi_txe_intr interrupt is not active prior to masking 1'b1: spi_txe_intr interrupt is active prior to masking

SPI_ICR

Address: Operational Base + offset (0x0038)

Interrupt Clear

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	WO	0x0	CTFOI Clear Transmit FIFO Overflow Interrupt Write 1 to Clear Transmit FIFO Overflow Interrupt
2	WO	0x0	CRFOI Clear Receive FIFO Overflow Interrupt Write 1 to Clear Receive FIFO Overflow Interrupt
1	WO	0x0	CRFUI Clear Receive FIFO Underflow Interrupt Write 1 to Clear Receive FIFO Underflow Interrupt
0	WO	0x0	CCI Clear Combined Interrupt Write 1 to Clear Combined Interrupt

SPI_DMACR

Address: Operational Base + offset (0x003c)

DMA Control

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	TDE Transmit DMA Enable 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
0	RW	0x0	RDE Receive DMA Enable 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled

SPI_DMATDLR

Address: Operational Base + offset (0x0040)

DMA Transmit Data Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the <code>dma_tx_req</code> signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and Transmit DMA Enable (DMACR[1]) = 1.

SPI_DMARDLR

Address: Operational Base + offset (0x0044)

DMA Receive Data Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, <code>dma_rx_req</code> is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and Receive DMA Enable(DMACR[0])=1.

SPI_TXDR

Address: Operational Base + offset (0x0400~0x07fc)

Transmit FIFO Data

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	WO	0x0000	TXDR Transmit FIFO Data Register. When it is written to, data are moved into the transmit FIFO.

SPI_RXDR

Address: Operational Base + offset (0x0800~0xbfc)

Receive FIFO Data

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RXDR Receive FIFO Data Register. When the register is read, data in the receive FIFO is accessed.

31.5 Interface description

SPI have 3 IOMUX, which is controlled by GRF_SOC_CON1[4:3].

When GRF_SOC_CON1[4:3] is 2'b00. The IOMUX is as follow.

Table 31-1 SPI interface description in master mode

Module Pin	Direction	Pad Name	IOMUX Setting
spi_clk	I/O	IO_SPIclk_UART1ctsn_GPIO1b0	GRF_GPIO1B_IOMUX[1:0]=2'b01
spi_csn0	I/O	IO_SPIcsn0_UART1rtsn_GPIO1b3	GRF_GPIO1B_IOMUX[7:6]=2'b01
spi_txd	O	IO_SPItxd_UART1sout_GPIO1b1	GRF_GPIO1B_IOMUX[3:2]=2'b01
spi_rxd	I	IO_SPIrxn_UART1sin_GPIO1b2	GRF_GPIO1B_IOMUX[5:4]=2'b01
spi_csn1	O	IO_SPIcsn1_GPIO1b4	GRF_GPIO1B_IOMUX[8]=1'b1

Note: *spi0_csn1*, *spi1_csn1* can only be used in master mode

Table 31-2 SPI interface description in slave mode

Module Pin	Direction	Pad Name	IOMUX Setting
spi_clk	I/O	IO_SPIclk_UART1ctsn_GPIO1b0	GRF_GPIO1B_IOMUX[1:0]=2'b01
spi_csn0	I/O	IO_SPIcsn0_UART1rtsn_GPIO1b3	GRF_GPIO1B_IOMUX[7:6]=2'b01
spi_txd	O	IO_SPItxd_UART1sout_GPIO1b1	GRF_GPIO1B_IOMUX[3:2]=2'b01
spi_rxd	I	IO_SPIrxn_UART1sin_GPIO1b2	GRF_GPIO1B_IOMUX[5:4]=2'b01

When GRF_SOC_CON1[4:3] is 2'b01. The IOMUX is as follow.

Table 31-3 SPI interface description in master mode

Module Pin	Direction	Pad Name	IOMUX Setting
spi_clk	I/O	IO_NANDale_SP11clk_GPIO2a0	GRF_GPIO2A_IOMUX[1:0]=2'b10
spi_csn0	I/O	IO_NANDd6_EM_MCd6_SPI1csn0_GPIO1d6	GRF_GPIO1D_IOMUX[13:12]=2'b11
spi_txd	O	IO_NANDd5_EM_MCd5_SPI1txd1_GPIO1d5	GRF_GPIO1D_IOMUX[11:10]=2'b11
spi_rxd	I	IO_NANDd4_EM_MCd4_SPI1rxn1_GPIO1d4	GRF_GPIO1D_IOMUX[9:8]=2'b11
spi_csn1	O	IO_NANDd7_EM_MCd7_SPI1csn1_GPIO1d7	GRF_GPIO1D_IOMUX[15:14]=2'b11

Note: *spi0_csn1*, *spi1_csn1* can only be used in master mode

Table 31-4 SPI interface description in slave mode

Module Pin	Direction	Pad Name	IOMUX Setting
spi_clk	I/O	IO_NANDale_SPI1clk_GP IO2a0	GRF_GPIO2A_IOMUX[1:0]=2'b10
spi_csn0	I/O	IO_NANDd6_EMMCd6_S PI1csn0_GP IO1d6	GRF_GPIO1D_IOMUX[13:12]=2'b11
spi_txd	O	IO_NANDd5_EMMCd5_S PI1txd1_GPI O1d5	GRF_GPIO1D_IOMUX[11:10]=2'b11
spi_rxd	I	IO_NANDd4_EMMCd4_S PI1rxd1_GPI O1d4	GRF_GPIO1D_IOMUX[9:8]=2'b11

When GRF_SOC_CON1[4:3] is 2'b10. The IOMUX is as follow.

Table 31-5 SPI interface description in master mode

Module Pin	Direction	Pad Name	IOMUX Setting
spi_clk	I/O	IO_I2S1sclk_SP Iclkm_GPIO0b1	GRF_GPIO0B_IOMUX[3:2]=2'b10
spi_csn0	I/O	IO_I2S1sdi_SPI csn0m_GPIO0b6	GRF_GPIO0B_IOMUX[13:12]=2'b10
spi_txd	O	IO_I2S1lrckrx_S PItxdm_GPIO0b3	GRF_GPIO0B_IOMUX[7:6]=2'b10
spi_rxd	I	IO_I2S1sdo_SPI rxdm_GPIO0b5	GRF_GPIO0B_IOMUX[11:10]=2'b10

Table 31-6 SPI interface description in slave mode

Module Pin	Direction	Pad Name	IOMUX Setting
spi_clk	I/O	IO_I2S1sclk_SPIclkm_G PIO0b1	GRF_GPIO0B_IOMUX[3:2]=2'b10
spi_csn0	I/O	IO_I2S1sdi_SPIcsn0m_G PIO0b6	GRF_GPIO0B_IOMUX[13:12]=2'b10
spi_txd	O	IO_I2S1lrckrx_SPItxdm_G GPIO0b3	GRF_GPIO0B_IOMUX[7:6]=2'b10
spi_rxd	I	IO_I2S1sdo_SPIrxdm_G PIO0b5	GRF_GPIO0B_IOMUX[11:10]=2'b10

31.6 Application Notes

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock

(sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the $F_{spi_clk} \geq 2 \times (\text{maximum } F_{sclk_out})$

When SPI Controller works as slave, the $F_{spi_clk} \geq 6 \times (\text{maximum } F_{sclk_in})$

Master Transfer Flow

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the sclk_out line. When the SPI is disabled (`SPI_ENR = 0`), no serial transfers can occur and sclk_out is held in "inactive" state, as defined by the serial protocol under which it operates.

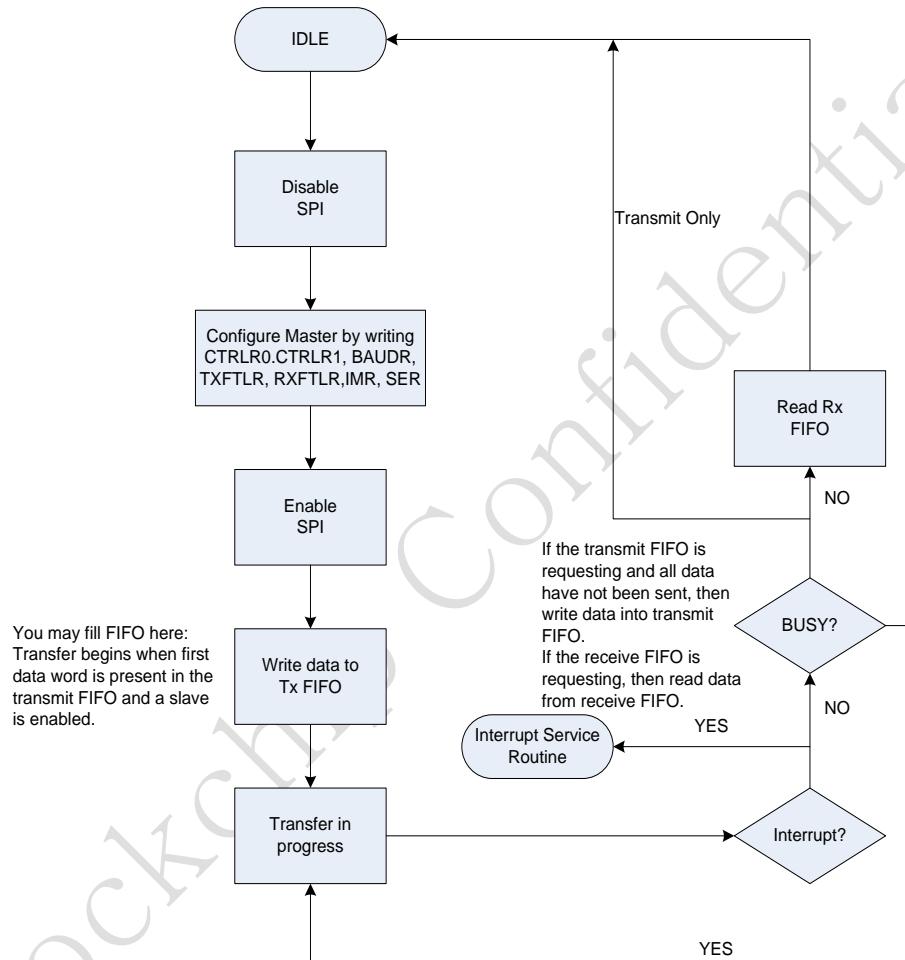


Fig.31-7SPI Master transfer flow diagram

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.

When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

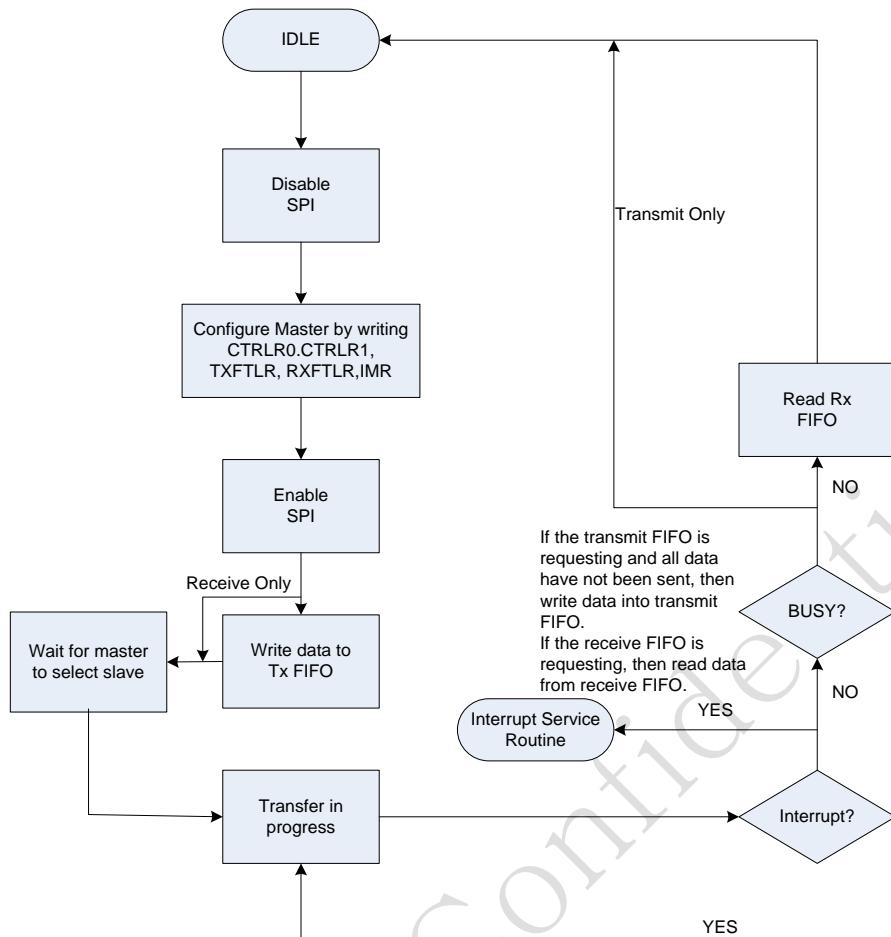


Fig.31-8SPI Slave transfer flow diagram

Chapter 32 UART

32.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

32.1.1 Features

UART Controller supports the following features:

- AMBA APB interface – Allows for easy integration into a Synthesizable Components for AMBA 2 implementation
- Support interrupt interface to interrupt controller
- UART1/UART2 contain two 32Bytes FIFOs for data receive and transmit, UART0's two embedded FIFOs are both 64Bytes for BT transfer
- Programmable serial data baud rate as calculated by the following: baud rate = (serial clock frequency)/(16×divisor)
- UART0 / UART1 / UART2 support auto flow-control
- UART0 / UART1 / UART2 are in peripheral subsystem

32.2 Block Diagram

This section provides a description about the functions and behavior under various conditions. The UART Controller comprises with:

- AMBA APB interface
- FIFO controllers
- Register block
- Modem synchronization block and baud clock generation block
- Serial receiver and serial transmitter

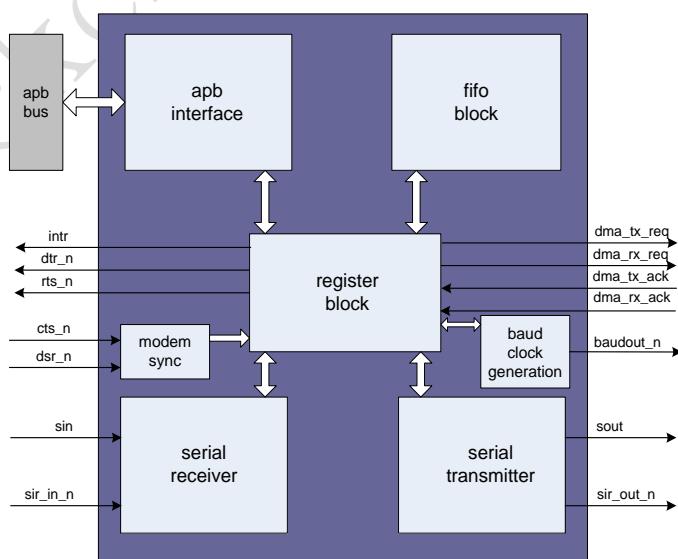


Fig. 32-1UART Architecture

APB INTERFACE

The host processor accesses data, control, and status information on the UART through the APB interface. The UART supports APB data bus widths of 8, 16, and 32 bits.

Register block

Be responsible for the main UART functionality including control, status and interrupt generation.

Modem Synchronization block

Synchronizes the modem input signal.

FIFO block

Be responsible for FIFO control and storage (when using internal RAM) or signaling to control external RAM (when used).

Baud Clock Generator

Generate the transmitter and receiver baud clock along with the output reference clock signal (baudout_n).

Serial Transmitter

Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. This makeup of serial data, referred to as a character can exit the block in two forms, either serial UART format or IrDA 1.0 SIR format.

Serial Receiver

Converts the serial data character (as specified by the control register) received in either the UART or IrDA 1.0 SIR format to parallel form. Parity error detection, framing error detection and line break detection is carried out in this block.

32.3 Function description

UART (RS232) Serial Protocol

Because the serial communication is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to perform simple error checking on the received data, as shown in Figure.

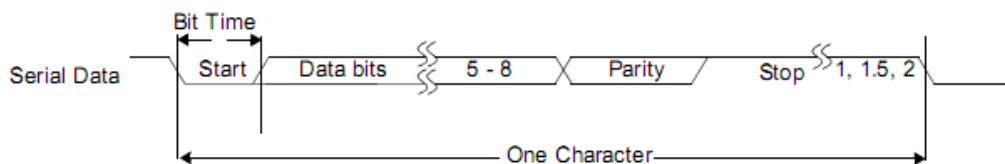


Fig. 32-2UART Serial protocol

Baud Clock

The baud rate is controlled by the serial clock (sclk or pclk in a single clock implementation) and the Divisor Latch Register (DLH and DLL). As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid-point for sampling is not difficult, that is every 16 baud clocks after the mid point sample of the start bit.

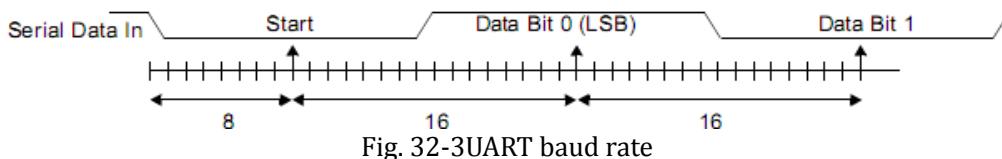


Fig. 32-3UART baud rate

FIFO Support

1. NONE FIFO MODE

If FIFO support is not selected, then no FIFOs are implemented and only a single receive data byte and transmit data byte can be stored at a time in the RBR and THR.

2. FIFO MODE

The FIFO depth of UART1/UART2/UART3 is 32bytes and the FIFO depth of UART0 is 64bytes. The FIFO mode of all the UART is enabled by register FCR[0].

Interrupts

The following interrupt types can be enabled with the IER register.

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE Interrupt mode)
- Modem Status

DMA Support

The uart supports DMA signaling with the use of two output signals (dma_tx_req_n and dma_rx_req_n) to indicate when data is ready to be read or when the transmit FIFO is empty.

The dma_tx_req_n signal is asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode.
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled.
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

The dma_rx_req_n signal is asserted under the following conditions:

- When there is a single character available in the Receive Buffer Register in non-FIFO mode.
- When the Receiver FIFO is at or above the programmed trigger level in FIFO mode.

Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control mode has been selected, it can be enabled with the Modem Control Register (MCR[5]). Following figure shows a block diagram of the Auto Flow Control functionality.

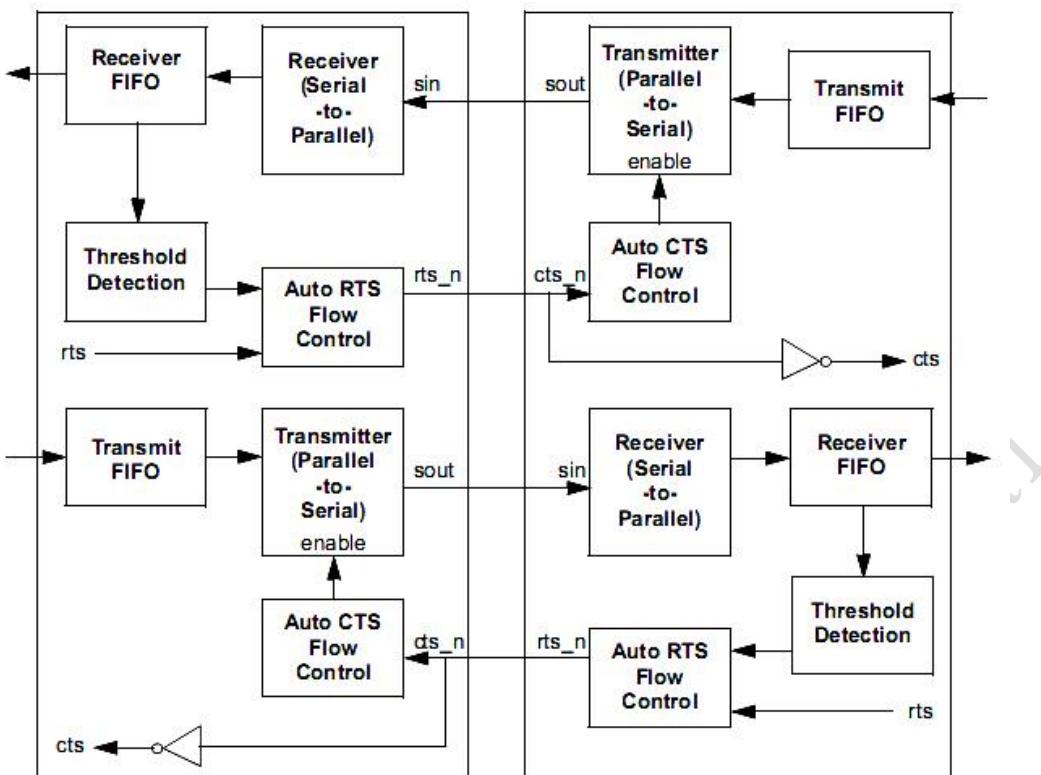


Fig. 32-4UART Auto flow control block diagram

Auto RTS – Becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0]) bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

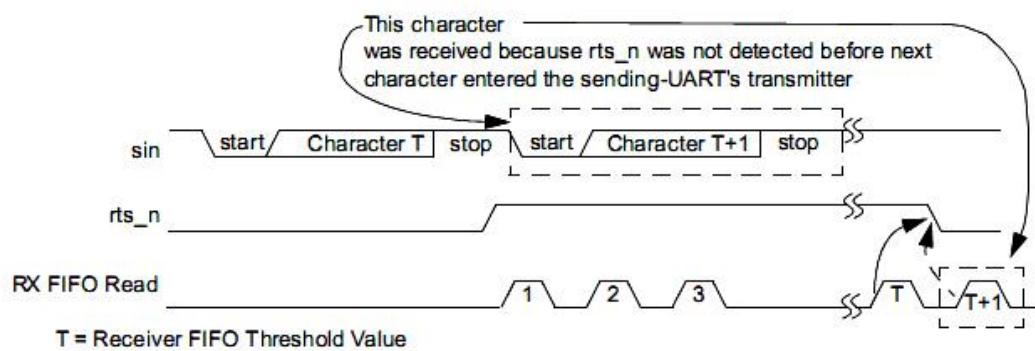


Fig. 32-5UART AUTO RTS TIMING

Auto CTS – becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- AFCE (MCR[5] bit is set)
- FIFOs are enabled through FIFO Control Register FCR[0] bit
- SIR mode is disabled (MCR[6] bit is not set)

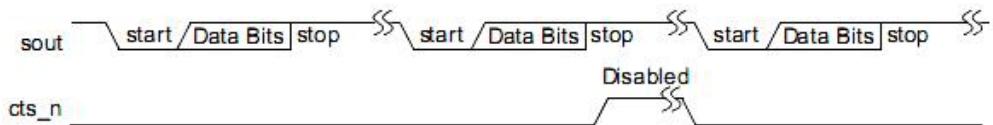


Fig. 32-6UART AUTO CTS TIMING

32.4 Register Description

This section describes the control/status registers of the design. There are 4 UARTs in RKAudi, and each one has its own base address.

32.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_THR	0x0000	W	0x00000000	Transmit Holding Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch (Low)
UART_DLH	0x0004	W	0x00000000	Divisor Latch (High)
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_IIR	0x0008	W	0x00000001	Interrupt Identification Register
UART_FCR	0x0008	W	0x00000000	FIFO Control Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000060	Line Status Register
UART_MSR	0x0018	W	0x00000000	Modem Status Register
UART_SCR	0x001c	W	0x00000000	Scratchpad Register
UART_SRBR	0x0030~0x006c	W	0x00000000	Shadow Receive Buffer Register
UART_STHR	0x0030~0x006c	W	0x00000000	Shadow Transmit Holding Register
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO Write
UART_USR	0x007c	W	0x00000006	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO Level
UART_RFL	0x0084	W	0x00000000	Receive FIFO Level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO Enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
UART_STET	0x00a0	W	0x00000000	Shadow TX Empty Trigger
UART_HTX	0x00a4	W	0x00000000	Halt TX
UART_DMASA	0x00a8	W	0x00000000	DMA Software Acknowledge
UART_CPR	0x00f4	W	0x00000000	Component Parameter Register
UART_UCV	0x00f8	W	0x3330382a	UART Component Version
UART_CTR	0x00fc	W	0x44570110	Component Type Register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

32.4.2 Detail Register Description

UART_RBR

Address: Operational Base + offset (0x0000)
Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	<p>data_input Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set.</p> <p>If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error.</p> <p>If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.</p>

UART_THR

Address: Operational Base + offset (0x0000)
Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>data_output Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten.</p> <p>If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p>

UART_DLL

Address: Operational Base + offset (0x0000)

Divisor Latch (Low)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	<p>baud_rate_divisor_L Lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock frequency) / (16 * divisor).</p> <p>Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.</p>

UART_DLH

Address: Operational Base + offset (0x0004)

Divisor Latch (High)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	baud_rate_divisor_H Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.

UART_IER

Address: Operational Base + offset (0x0004)

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	prog_thre_int_en Programmable THRE Interrupt Mode Enable This is used to enable/disable the generation of THRE Interrupt. 1'b0: disabled 1'b1: enabled
6:4	RO	0x0	reserved
3	RW	0x0	modem_status_int_en Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 1'b0: disabled 1'b1: enabled
2	RW	0x0	receive_line_status_int_en Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 1'b0: disabled 1'b1: enabled
1	RW	0x0	trans_hold_empty_int_en Enable Transmit Holding Register Empty Interrupt.
0	RW	0x0	receive_data_available_int_en Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 1'b0: disabled 1'b1: enabled

UART_IIR

Address: Operational Base + offset (0x0008)

Interrupt Identification Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	RO	0x0	fifos_en FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 2'b00: disabled 2'b11: enabled
5:4	RO	0x0	reserved
3:0	RO	0x1	int_id Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types: 4'b0000: modem status 4'b0001: no interrupt pending 4'b0010: THR empty 4'b0100: received data available 4'b0110: receiver line status 4'b0111: busy detect 4'b1100: character timeout

UART_FCR

Address: Operational Base + offset (0x0008)

FIFO Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	WO	0x0	rcvr_trigger RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 2'b00: 1 character in the FIFO 2'b01: FIFO 1/4 full 2'b10: FIFO 1/2 full 2'b11: FIFO 2 less than full
5:4	WO	0x0	tx_empty_trigger TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 2'b00: FIFO empty 2'b01: 2 characters in the FIFO 2'b10: FIFO 1/4 full 2'b11: FIFO 1/2 full

Bit	Attr	Reset Value	Description
3	WO	0x0	dma_mode DMA Mode This determines the DMA signalling mode used for the dma_tx_req_n anddma_rx_req_n output signals when additional DMA handshaking signals are not selected . 1'b0: mode 0 1'b1: mode 11100 = character timeout.
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are select. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	WO	0x0	fifo_en FIFO Enable. FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

UART_LCR

Address: Operational Base + offset (0x000c)

Line Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	div_lat_access Divisor Latch Access Bit. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.

Bit	Attr	Reset Value	Description
6	RW	0x0	<p>break_ctrl Break Control Bit.</p> <p>This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If MCR[6] set to one, the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to thereceiver and the sir_out_n line is forced low.</p>
5	RO	0x0	reserved
4	RW	0x0	<p>even_parity_sel Even Parity Select.</p> <p>Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.</p>
3	RW	0x0	<p>parity_en Parity Enable.</p> <p>Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively.</p> <p>1'b0: parity disabled 1'b1: parity enabled</p>
2	RW	0x0	<p>stop_bits_num Number of stop bits.</p> <p>Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits select, the receiver checks only the first stop bit.</p> <p>1'b0: 1 stop bit 1'b1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit.</p>

Bit	Attr	Reset Value	Description
1:0	RW	0x0	<p>data_length_sel Data Length Select. Writable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows:</p> <ul style="list-style-type: none"> 2'b00: 5 bits 1'b01: 6 bits 1'b10: 7 bits 1'b11: 8 bits

UART_MCR

Address: Operational Base + offset (0x0010)

Modem Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	<p>sir_mode_en SIR Mode Enable. SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode.</p> <ul style="list-style-type: none"> 1'b0: IrDA SIR Mode disabled 1'b1: IrDA SIR Mode enabled
5	RW	0x0	<p>auto_flow_ctrl_en Auto Flow Control Enable. 1'b0: Auto Flow Control Mode disabled 1'b1: Auto Flow Control Mode enabled</p>
4	RW	0x0	<p>loopback LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes.</p>
3	RW	0x0	<p>out2 OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is:</p> <ul style="list-style-type: none"> 1'b0: out2_n de-asserted (logic 1) 1'b1: out2_n asserted (logic 0)
2	RW	0x0	<p>out1 OUT1</p>
1	RW	0x0	<p>req_to_send Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data.</p>

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>data_terminal_ready Data Terminal Ready.</p> <p>This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is:</p> <p>1'b0: dtr_n de-asserted (logic 1) 1'b1: dtr_n asserted (logic 0)</p>

UART_LSR

Address: Operational Base + offset (0x0014)

Line Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	<p>receiver_fifo_error Receiver FIFO Error bit.</p> <p>This bit is relevant if FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO.</p> <p>1'b0: no error in RX FIFO 1'b1: error in RX FIFO</p>
6	RO	0x1	<p>trans_empty Transmitter Empty bit.</p> <p>Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.</p>
5	RO	0x1	<p>trans_hold_reg_empty Transmit Holding Register Empty bit.</p> <p>If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty.</p> <p>This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If IER[7] set to one and FCR[0] set to one respectively, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.</p>
4	RO	0x0	<p>break_int Break Interrupt bit.</p> <p>This is used to indicate the detection of a break sequence on the serial input data.</p>

Bit	Attr	Reset Value	Description
3	RO	0x0	framing_error Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.
2	RO	0x0	parity_error Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.
1	RO	0x0	overrun_error Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read.
0	RO	0x0	data_ready Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 1'b0: no data ready 1'b1: data ready

UART_MSR

Address: Operational Base + offset (0x0018)

Modem Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	data_carrier_detect Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n.
6	RO	0x0	ring_indicator Ring Indicator. This is used to indicate the current state of the modem control line ri_n.
5	RO	0x0	data_set_ready Data Set Ready. This is used to indicate the current state of the modem control line dsr_n.
4	RO	0x0	clear_to_send Clear to Send. This is used to indicate the current state of the modem control line cts_n.
3	RO	0x0	delta_data_carrier_detect Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.

Bit	Attr	Reset Value	Description
2	RO	0x0	trailing_edge_ring_indicator Trailing Edge of Ring Indicator. Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.
1	RO	0x0	delta_data_set_ready Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.
0	RO	0x0	delta_clear_to_send Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.

UART_SCR

Address: Operational Base + offset (0x001c)

Scratchpad Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	temp_store_space This register is for programmers to use as a temporary storage space.

UART_SRBR

Address: Operational Base + offset (0x0030~0x006c)

Shadow Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	<p>shadow_rbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set.</p> <p>If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error.</p> <p>If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.</p>

UART_STHR

Address: Operational Base + offset (0x0030~0x006c)

Shadow Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	<p>shadow_thr This is a shadow register for the THR.</p>

UART_FAR

Address: Operational Base + offset (0x0070)

FIFO Access Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	<p>fifo_access_test_en This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not enabled it allows the RBR to be written by the master and the THR to be read by the master.</p> <p>1'b0: FIFO access mode disabled 1'b1: FIFO access mode enabled</p>

UART_TFR

Address: Operational Base + offset (0x0074)

Transmit FIFO Read

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	trans_fifo_read Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO.

UART_RFW

Address: Operational Base + offset (0x0078)

Receive FIFO Write

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	receive_fifo_framing_error Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
8	WO	0x0	receive_fifo_parity_error Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
7:0	WO	0x00	receive_fifo_write Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs not enabled, the data that is written to the RFWD is pushed into the RBR.

UART_USR

Address: Operational Base + offset (0x007c)

UART Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	receive_fifo_full Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 1'b0: Receive FIFO not full 1'b1: Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.

Bit	Attr	Reset Value	Description
3	RO	0x0	receive_fifo_not_empty Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 1'b0: Receive FIFO is empty 1'b1: Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	RO	0x1	trans_fifo_empty Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty
1	RO	0x1	trans_fifo_not_full Transmit FIFO Not Full. This is used to indicate that the transmit FIFO is not full. 1'b0: Transmit FIFO is full 1'b1: Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	RO	0x0	uart_busy UART Busy. UART Busy. This indicates that a serial transfer is in progress, when cleared indicates that the UART is idle or inactive. 1'b0: UART is idle or inactive 1'b1: UART is busy (actively transferring data)

UART_TFL

Address: Operational Base + offset (0x0080)

Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	trans_fifo_level Transmit FIFO Level. This indicates the number of data entries in the transmit FIFO.

UART_RFL

Address: Operational Base + offset (0x0084)

Receive FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x00	receive_fifo_level Receive FIFO Level. This indicates the number of data entries in the receive FIFO.

UART_SRR

Address: Operational Base + offset (0x0088)

Software Reset Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]).
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]).
0	WO	0x0	uart_reset UART Reset. This asynchronously resets the UART and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

UART_SRTS

Address: Operational Base + offset (0x008c)

Shadow Request to Send

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_req_to_send Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR.

UART_SBCR

Address: Operational Base + offset (0x0090)

Shadow Break Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_break_ctrl Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR.

UART_SDMAM

Address: Operational Base + offset (0x0094)

Shadow DMA Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	shadow_dma_mode Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]).

UART_SFE

Address: Operational Base + offset (0x0098)

Shadow FIFO Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_fifo_en Shadow FIFO Enable. Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).

UART_SRT

Address: Operational Base + offset (0x009c)

Shadow RCVR Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
1:0	RW	0x0	shadow_rcvr_trigger Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).

UART_STET

Address: Operational Base + offset (0x00a0)

Shadow TX Empty Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
1:0	RW	0x0	shadow_tx_empty_trigger Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).

UART_HTX

Address: Operational Base + offset (0x00a4)

Halt TX

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	halt_tx_en This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 1'b0: Halt TX disabled 1'b1: Halt TX enabled

UART_DMASA

Address: Operational Base + offset (0x00a8)

DMA Software Acknowledge

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	dma_software_ack This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.

UART_UCV

Address: Operational Base + offset (0x00f8)

UART Component Version

Bit	Attr	Reset Value	Description
31:0	RO	0x3330382a	ver ASCII value for each number in the version

UART_CTR

Address: Operational Base + offset (0x00fc)

Component Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	peripheral_id This register contains the peripherals identification code.

32.5 Interface description

Table 32-1UART Interface Description

Module pin	Dir.	Pad name	IOMUX
UART0 Interface			
uart0_sin	I	IO_SCclk_URT0sin_GPIO2d3	GRF_GPIO2D_IOMUX [7:6]=2'b10
uart0_sout	O	IO_SCrst_URT0sout_GPIO2d2	GRF_GPIO2D_IOMUX [5:4]=2'b10
uart0_cts_n	I	IO_SCdetect_URT0Ctsn_GPIO2d5	GRF_GPIO2D_IOMUX [11:10]=2'b10
uart0_rts_n	O	IO_Scio_URT0rtsn_GPIO0c1	GRF_GPIO0C_IOMUX

			[3:2]=2'b10
UART1 Interface			
uart1_sin	I	IO_SPIrxd_UART1sin_GPIO1b2	GRF_GPIO1B_IOMUX [5:4]=2'b10
uart1_sout	O	IO_SPItxd_UART1sout_GPIO1b1	GRF_GPIO1B_IOMUX [3:2]=2'b10
uart1_cts_n	I	IO_SPIclk_UART1ctsn_GPIO1b0	GRF_GPIO1B_IOMUX [1:0]=2'b10
uart1_rts_n	O	IO_SPIcsn0_UART1rtsn_GPIO1b3	GRF_GPIO1B_IOMUX [7:6]=2'b10
UART2 Interface			
uart2_sin	I	IO_MMCD0d1_UART2rx_GPIO1c3	GRF_GPIO1C_IOMUX [7:6]=2'b10
uart2_sout	O	IO_MMCD0d0_UART2tx_GPIO1c2	GRF_GPIO1C_IOMUX [5:4]=2'b10

32.6 Application Notes

32.6.1 None FIFO Mode Transfer Flow

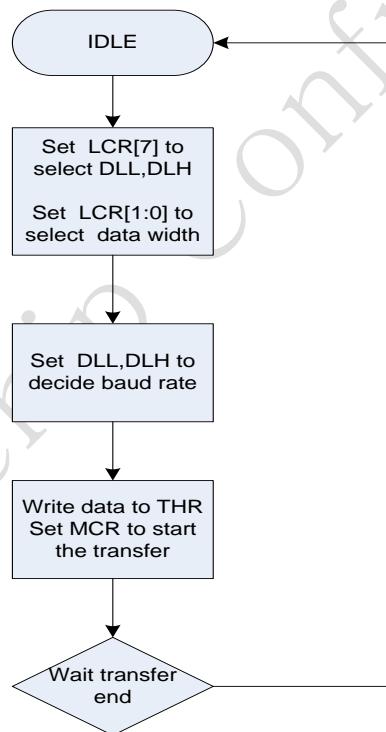


Fig. 32-7UART none fifo mode

32.6.2 FIFO Mode Transfer Flow

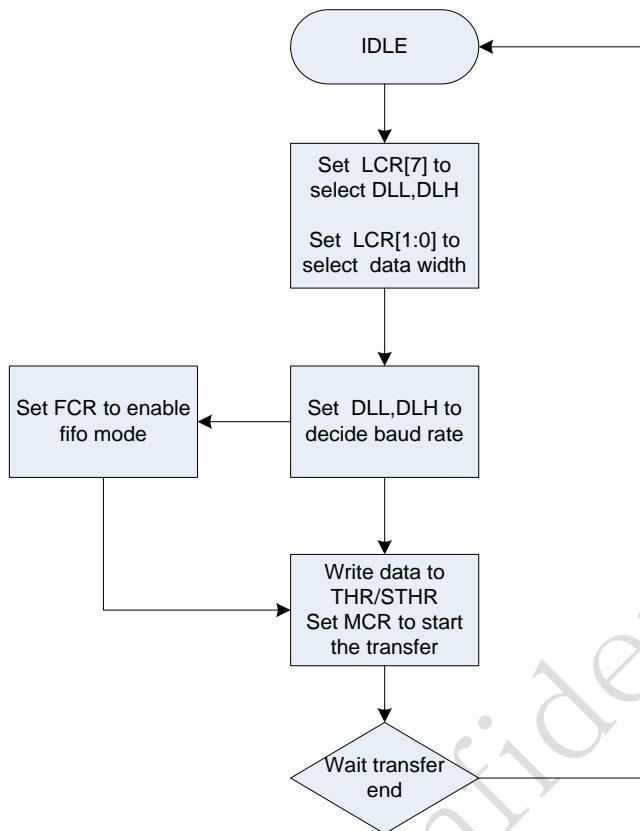


Fig. 32-8UART fifo mode

The UART is an APB slave performing:

Serial-to-parallel conversion on data received from a peripheral device.

Parallel-to-serial conversion on data transmitted to the peripheral device.

The CPU reads and writes data and control/status information through the APB interface. The transmitting and receiving paths are buffered with internal FIFO memories enabling up to 32-bytes to be stored independently in both transmit and receive modes. A baud rate generator can generate a common transmit and receive internal clock input. The baud rates will depend on the internal clock frequency. The UART will also provide transmit, receive and exception interrupts to system. A DMA interface is implemented for improving the system performance.

32.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation.

UART0 source clocks can be selected from four PLL outputs (CODEC PLL/GENERAL PLL/GENERAL PLL D2/USBPHY_480M). UART1 and UART2 source clocks can be selected from three PLL outputs (CODEC PLL/GENERAL PLL/GENERAL PLL D2). UART clocks can be generated by 1 to 64 division of its source clock, or can be fractionally divided again.

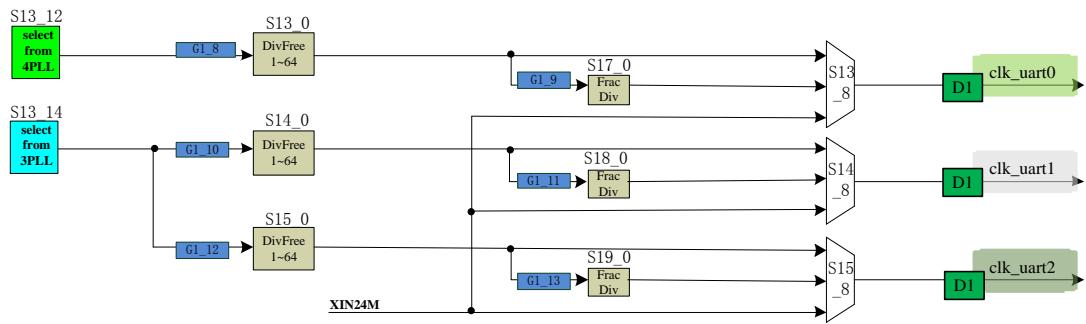


Fig. 32-9UART clock generation

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 32-2UART baud rate configuration

Baud Rate	Reference Configuration
115.2 Kbps	Configure GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Config UART_DLL to 8.
460.8 Kbps	Configure GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 2.
921.6 Kbps	Configure GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 1.
1.5 Mbps	Choose GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 16 to get 24MHz clock; Configure UART_DLL to 1
3 Mbps	Choose GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 8 to get 48MHz clock; Configure UART_DLL to 1
4 Mbps	Configure GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 6 to get 64MHz clock; Configure UART_DLL to 1

Chapter 33 I2C Interface

33.1 Overview

The Inter-Integrated Circuit (I2C) is a two wired (SCL and SDA), bi-directional serial bus that provides an efficient and simple method of information exchange between devices. This I2C bus controller supports master mode acting as a bridge between AMBA protocol and generic I2C bus system.

33.1.1 Features

I2C Controller supports the following features:

- Item Compatible with I2C-bus
- AMBA APB slave interface
- Supports master mode of I2C bus
- Software programmable clock frequency and transfer rate up to 400Kbit/sec
- Supports 7 bits and 10 bits addressing modes
- Interrupt or polling driven multiple bytes data transfer
- Clock stretching and wait state generation
- I2C0 / I2C1 / I2C2 /I2C3 are in peripheral sub-system

33.2 Block Diagram

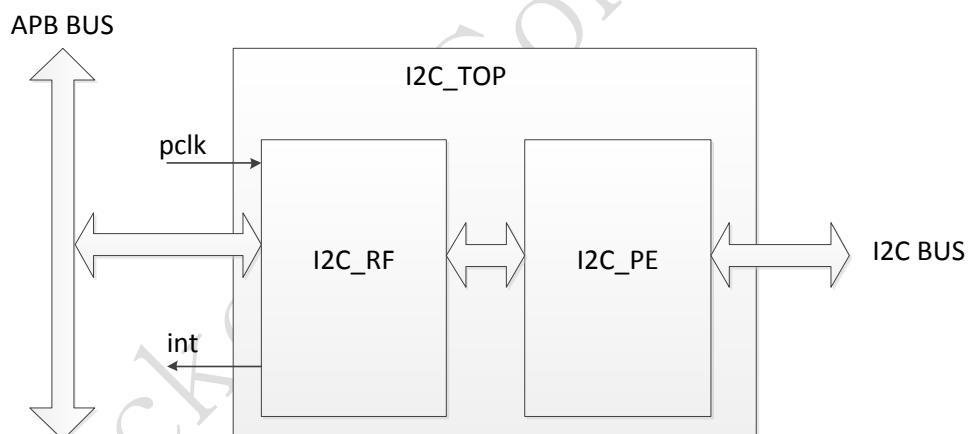


Fig.33-1I2C architecture

I2C_RF

I2C_RF module is used to control the I2C controller operation by the host with APB interface. It implements the register set and the interrupt functionality. The CSR component operates synchronously with the pclk clock.

I2C_PE

I2C_PE module implements the I2C master operation for transmit data to and receive data from other I2C devices. The I2C master controller operates synchronously with the pclk.

I2C_TOP

I2C_TOP module is the top module of the I2C controller.

33.3 Function description

This chapter provides a description about the functions and behavior under various conditions.

The I2C controller supports only Masterfunction. It supports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 400Kbit/sec.

The operations of I2C controller is divided to 2 parts and described separately: initialization and master mode programming.

33.3.1 Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting & configuration must be conformed, which includes:

- I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.
- I2C Clock Rate: The I2C controller uses the APB clock as the system clock so the APB clock will determine the I2C bus clock. The correct register setting is subject to the system requirement.

33.3.2 Master Mode Programming

1. SCL Clock: When the I2C controller is programmed in Master mode, the SCL frequency is determined by I2C_CLKDIV register. The SCL frequency is calculated by the following formula:

$$\text{SCL Divisor} = 8 * ((\text{CLKDIVL} + 1) + (\text{CLKDIVH} + 1))$$

$$\text{SCL} = \text{PCLK} / \text{SCLK Divisor}$$

2. Data Receiver Register Access

When the i2c controller received MRXCNT bytes data, CPU can get the datas through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 byte data in one transaction.

When MRXCNT register is written, the I2C controller will start to drive SCL to receive data.

3. Transmit Trasmitter Register

Data to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 byte data in one transaction. The lower byte will be transmitted first.

When MTXCNT register is written, the I2C controller will start to transmit data.

4. Start Command

Write 1 to I2C_CON[3], the controller will send I2C start command.

5. Stop Command

Write 1 to I2C_CON[4], the controller will send I2C stop command

6. I2C Operation mode

There are four i2c operation modes.

When I2C_CON[2:1] is 2'b00, the controller transmit all valid data in

TXDATA0~TXDATA7 byte by byte. The controller will transmit lower byte first.

When I2C_CON[2:1] is 2'b01, the controller will transmit device address in MRXADDR first (Write/Read bit = 0) and then transmit device register address in MRXRADDR. After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.

When I2C_CON[2:1] is 2'b10, the controller is in receive mode, it will triggered clock to read MRXCNT byte data.

When I2C_CON[2:1] is 2'b11, the controller will transmit device address in MRXADDR first (Write/Read bit = 1) and then transmit device register address in MRXRADDR . After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.

7. Read/Write Command

When I2C_OPMODE(I2C_CON[2:1]) is 2'b01 or 2'b11, the Read/Write command bit is decided by controller itself.

In RX only mode (I2C_CON[2:1] is 2'b10), the Read/Write command bit is decided by MRXADDR[0].

In TX only mode (I2C_CON[[2:1] is 2'b00), the Read/Write command bit is decided by TXDATA[0].

8. Master Interrupt Condition

There are 7 interrupt bits in I2C_ISR register related to master mode.

Byte transmitted finish interrupt (Bit 0): The bit is asserted when Master complete transmitting a byte.

Byte received finish interrupt (Bit 1): The bit is asserted when Master complete receiving a byte.

MTXCNT bytes data transmitted finish interrupt (Bit 2): The bit is asserted when Master complete transmitting MTXCNT bytes.

MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master complete receiving MRXCNT bytes.

Start interrupt(Bit 4): The bit is asserted when Master finish asserting start command to I2C bus.

Stop interrupt (Bit 5): The bit is asserted when Master finish asserting stop command to I2C bus.

NAK received interrupt (Bit 6): The bit is asserted when Master received a NAK handshake.

9. Last byte acknowledge control

If I2C_CON[5] is 1, the I2C controller will transmit NAK handshake to slave when the last byte received in RX only mode.

If I2C_CON[5] is 0, the I2C controller will transmit ACK handshake to slave when the last byte received in RX only mode.

10. How to handle nak handshake received

If I2C_CON[6] is 1, the I2C controller will stop all transactions when NAK handshake received. And the software should take responsibility to handle the problem.

If I2C_CON[6] is 0, the I2C controller will ignore all NAK handshake received.

11. I2C controller data transfer waveform

● Bit transferring

(a) Data Validity

The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.

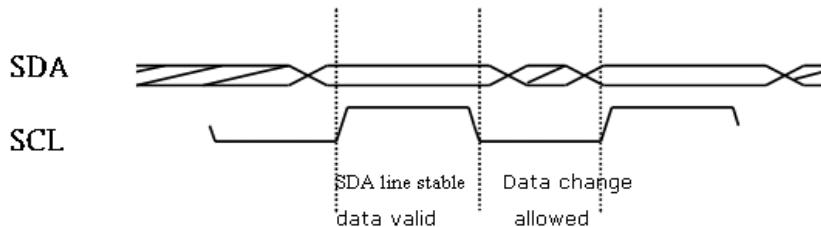


Fig.33-2I2C DATA Validity

(b) START and STOP conditions

START condition occurs when SDA goes low while SCL is in high period. STOP condition is generated when SDA line goes high while SCL is in high state.

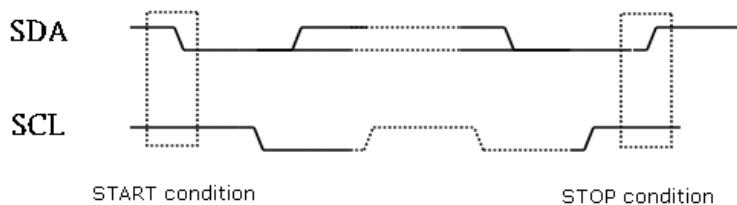


Fig.33-3I2C Start and stop conditions

● Data transfer

(a) Acknowledge

After a byte of data transferring (clocks labeled as 1~8), in 9th clock the receiver must assert an ACK signal on SDA line, if the receiver pulls SDA line to low, it means "ACK", on the contrary, it's "NOT ACK".

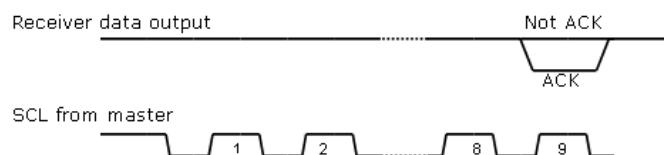


Fig.33-4I2C Acknowledge

(b) Byte transfer

The master own I2C bus might initiate multi byte to transfer to a slave. The transfer starts from a "START" command and ends in a "STOP"command. After every byte transfer, the receiver must reply an ACK to transmitter.

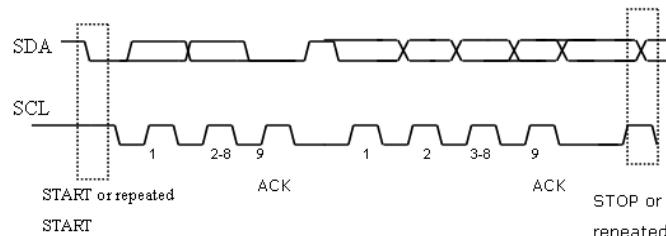


Fig.33-5I2C byte transfer

33.4 Register Description

This section describes the control/status registers of the design.

33.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2C_CON	0x0000	W	0x00000000	control register
I2C_CLKDIV	0x0004	W	0x00060006	Clock divisor register
I2C_MRXCADDR	0x0008	W	0x00000000	the slave address accessed for master receive mode
I2C_MRXRADDR	0x000c	W	0x00000000	the slave register address accessed for master receive mode
I2C_MTXCNT	0x0010	W	0x00000000	master transmit count
I2C_MRXCNT	0x0014	W	0x00000000	master receive count
I2C_IEN	0x0018	W	0x00000000	interrupt enable register
I2C_IPD	0x001c	W	0x00000000	interrupt pending register
I2C_FCNT	0x0020	W	0x00000000	finished count
I2C_TXDATA0	0x0100	W	0x00000000	I2C transmit data register 0
I2C_TXDATA1	0x0104	W	0x00000000	I2C transmit data register 1
I2C_TXDATA2	0x0108	W	0x00000000	I2C transmit data register 2
I2C_TXDATA3	0x010c	W	0x00000000	I2C transmit data register 3
I2C_TXDATA4	0x0110	W	0x00000000	I2C transmit data register 4
I2C_TXDATA5	0x0114	W	0x00000000	I2C transmit data register 5
I2C_TXDATA6	0x0118	W	0x00000000	I2C transmit data register 6
I2C_TXDATA7	0x011c	W	0x00000000	I2C transmit data register 7
I2C_RXDATA0	0x0200	W	0x00000000	I2C receive data register 0
I2C_RXDATA1	0x0204	W	0x00000000	I2C receive data register 1
I2C_RXDATA2	0x0208	W	0x00000000	I2C receive data register 2
I2C_RXDATA3	0x020c	W	0x00000000	I2C receive data register 3
I2C_RXDATA4	0x0210	W	0x00000000	I2C receive data register 4
I2C_RXDATA5	0x0214	W	0x00000000	I2C receive data register 5
I2C_RXDATA6	0x0218	W	0x00000000	I2C receive data register 6
I2C_RXDATA7	0x021c	W	0x00000000	I2C receive data register 7

Notes:
Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

33.4.2 Detail Register Description

I2C_CON

Address: Operational Base + offset (0x0000)
 control register

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	act2nak operation when NAK handshake is received 1'b0: ignored 1'b1: stop transaction
5	RW	0x0	ack last byte acknowledge control last byte acknowledge control in master receive mode . 1'b0: ACK 1'b1: NAK
4	W1C	0x0	stop stop enable when this bit is written to 1, I2C will generate stop signal. It cleared itself when stop operation ends.
3	W1C	0x0	start start enable when this bit is written to 1, I2C will generate start signal. It cleared itself when start operation ends.
2:1	RW	0x0	i2c_mode 2'b00: transmit only 2'b01: transmit address (device + register address) --> restart -->transmit address ->receive only 2'b10:receive only 2'b11: transmit address (device + register address, write/read bit is 1) --> restart -->transmit address (device address) -->receive data
0	RW	0x0	i2c_en i2c module enable 1: i2c is enabled. 0: i2c is disabled.

I2C_CLKDIV

Address: Operational Base + offset (0x0004)

Clock divisor register

Bit	Attr	Reset Value	Description
31:16	RW	0x0006	CLKDIVH SCL high level clock count $T(SCL_HIGH) = T(PCLK) * (CLKDIVH+1) * 8$
15:0	RW	0x0006	CLKDIVL SCL low level clock count $T(SCL_LOW) = T(PCLK) * (CLKDIVL+1) * 8$

I2C_MRXADDR

Address: Operational Base + offset (0x0008)

the slave address accessed for master receive mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	addhvld address high byte valid
25	RW	0x0	addmvld address middle byte valid
24	RW	0x0	addlvld address low byte valid
23:0	RW	0x000000	saddr master address register the lowest bit indicate write or read

I2C_MRXRADDR

Address: Operational Base + offset (0x000c)

the slave register address accessed for master receive mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	sraddhvld address high byte valid
25	RW	0x0	sraddmvld address middle byte valid
24	RW	0x0	sraddlvld address low byte valid
23:0	RW	0x000000	sraddr slave register address accessed

I2C_MTXCNT

Address: Operational Base + offset (0x0010)

master transmit count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mtxcnt master transmit count

I2C_MRXCNT

Address: Operational Base + offset (0x0014)

masterreceive count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mrxcnt master receive count

I2C_IEN

Address: Operational Base + offset (0x0018)

interrupt enable register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	nakrcvien NAK handshake received interrupt enable

Bit	Attr	Reset Value	Description
5	RW	0x0	stopien stop operation finished interrupt enable
4	RW	0x0	startien start operation finished interrupt enable
3	RW	0x0	mbrfien MRXCNT data received finished interrupt enable
2	RW	0x0	mbtfien MTXCNT data transmit finished interrupt enable
1	RW	0x0	brfien byte receive finished interrupt enable
0	RW	0x0	btfien byte transmit finished interrupt enable

I2C_IPD

Address: Operational Base + offset (0x001c)
interrupt pending register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	nakrcvipd NAK handshake received interrupt pending bit
5	RW	0x0	stopipd stop operation finished interrupt pending bit
4	RW	0x0	startipd start operation finished interrupt pending bit
3	RW	0x0	mbrfipd MRXCNT data received finished interrupt pending bit
2	RW	0x0	mbtfipd MTXCNT data transmitfinished interrupt pending bit
1	RW	0x0	brfipd byte receivefinished interrupt pending bit
0	RW	0x0	btfipd byte transmitfinished interrupt pending bit

I2C_FCNT

Address: Operational Base + offset (0x0020)
finished count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	fcnt finished count the count of data which has been transmitted or received for debug purpose

I2C_TXDATA0

Address: Operational Base + offset (0x0100)

I2C transmit data register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0

I2C_TXDATA1

Address: Operational Base + offset (0x0104)

I2C transmit data register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1

I2C_TXDATA2

Address: Operational Base + offset (0x0108)

I2C transmit data register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2

I2C_TXDATA3

Address: Operational Base + offset (0x010c)

I2C transmit data register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3

I2C_TXDATA4

Address: Operational Base + offset (0x0110)

I2C transmit data register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4

I2C_TXDATA5

Address: Operational Base + offset (0x0114)

I2C transmit data register 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5

I2C_TXDATA6

Address: Operational Base + offset (0x0118)

I2C transmit data register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6

I2C_TXDATA7

Address: Operational Base + offset (0x011c)

I2C transmit data register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7

I2C_RXDATA0

Address: Operational Base + offset (0x0200)

I2C receive data register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata0

I2C_RXDATA1

Address: Operational Base + offset (0x0204)

I2C receive data register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata1

I2C_RXDATA2

Address: Operational Base + offset (0x0208)

I2C receive data register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata2

I2C_RXDATA3

Address: Operational Base + offset (0x020c)

I2C receive data register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata3

I2C_RXDATA4

Address: Operational Base + offset (0x0210)

I2C receive data register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata4

I2C_RXDATA5

Address: Operational Base + offset (0x0214)

I2C receive data register 5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata5

I2C_RXDATA6

Address: Operational Base + offset (0x0218)

I2C receive data register 6

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata6

I2C_RXDATA7

Address: Operational Base + offset (0x021c)

I2C receive data register 7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata7

33.5 Interface description

Table 33-1 I2C Interface Description

Module pin	Direction	Pad name	IOMUX
I2C0 Interface			
i2c0_sda	I/O	IO_I2C0pm_usda_GPIO0a1	GRF_GPIO0A_IOMUX[3:2]=2'b01
i2c0_scl	I/O	IO_I2C0pm_uscl_GPIO0a0	GRF_GPIO0A_IOMUX[1:0]=2'b01
I2C1 Interface			
i2c1_sda	I/O	IO_I2C1tps_da_GPIO0a3	GRF_GPIO0A_IOMUX[7:6]=2'b01
i2c1_scl	I/O	IO_I2C1tpsc_l_GPIO0a2	GRF_GPIO0A_IOMUX[5:4]=2'b01
I2C2 Interface			
i2c2_sda	I/O	IO_LCD0d18_EBCgdrI2C2sda_GPIO2c4	GRF_GPIO2C_IOMUX[9:8]=2'b11
i2c2_scl	I/O	IO_LCD0d19_EBCsdshrI2C2scl_GPIO2c5	GRF_GPIO2C_IOMUX[11:10]=2'b11
I2C3 Interface			
i2c3_sda	I/O	IO_I2C3cifs_da_HDMIddcsda_GPIO0a7	GRF_GPIO0A_IOMUX[15:14]=2'b01
i2c3_scl	I/O	IO_I2C3cifs_cl_HDMIddcscl_GPIO0a6	GRF_GPIO0A_IOMUX[13:12]=2'b01

33.6 Application Notes

The I2C controller core operation flow chart below is to describe how the software configures and performs an I2C transaction through this I2C controller core. Descriptions are divided into 3 sections, transmit only mode, receive only mode, and mix mode. Users are strongly advised to following.

- Transmit only mode (I2C_CON[1:0]=2'b00)

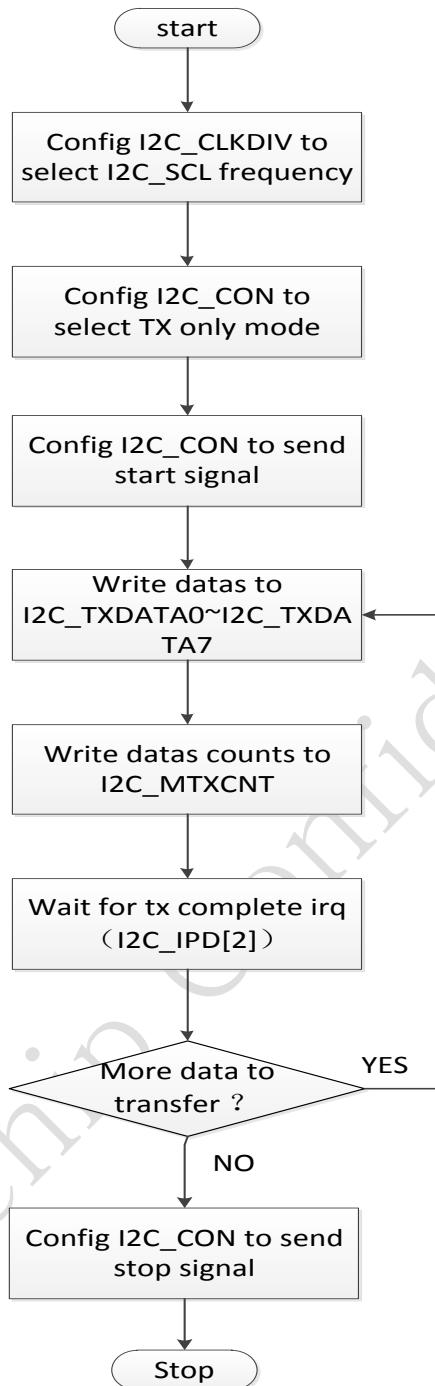


Fig.33-6I2C Flow chat for transmit only mode

- Receive only mode ($I2C_CON[1:0]=2'b10$)

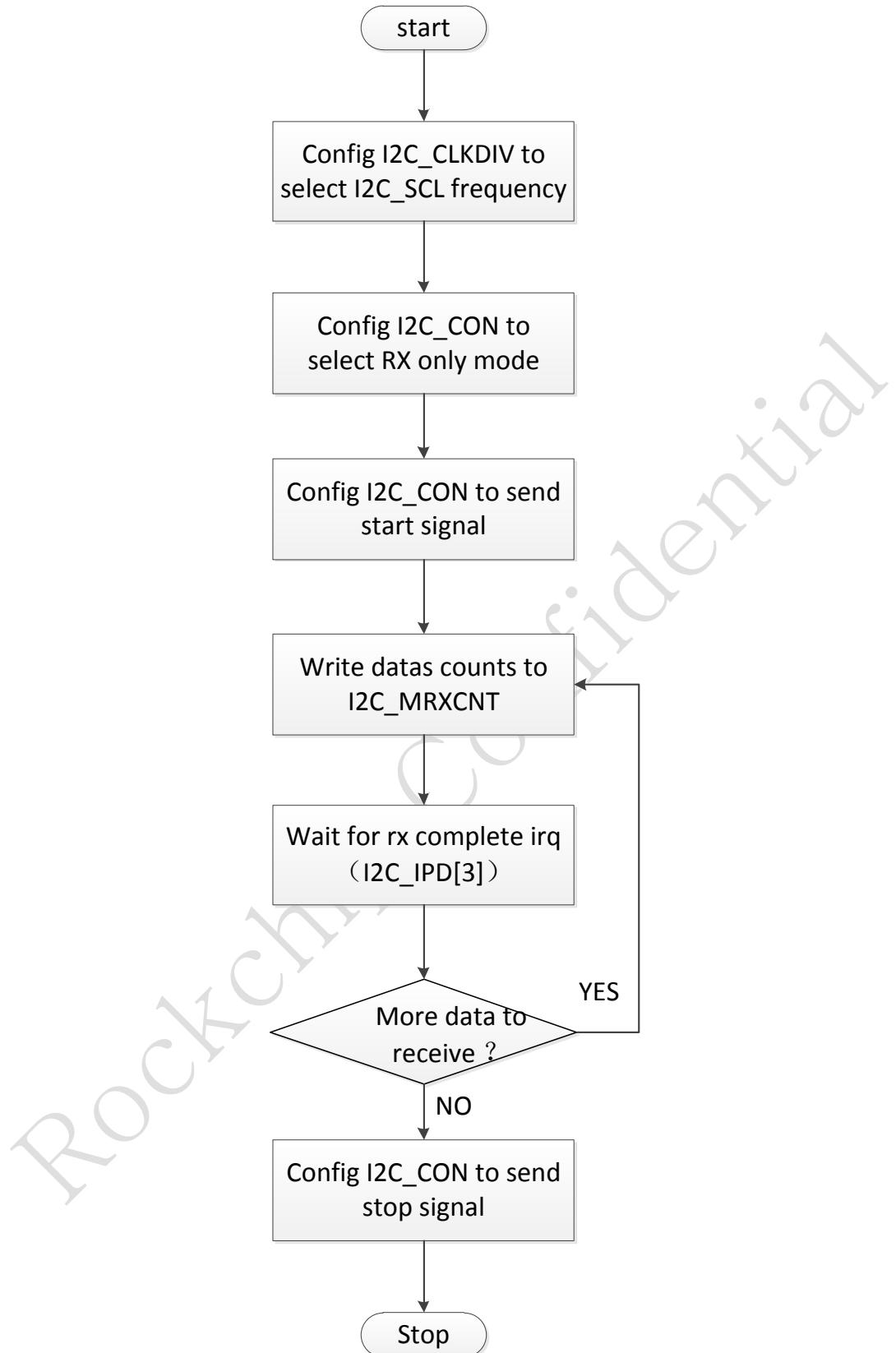


Fig.33-7I2C Flow chat for receive only mode

- Mix mode ($I2C_CON[1:0]=2'b01$ or $I2C_CON[1:0]=2'b11$)

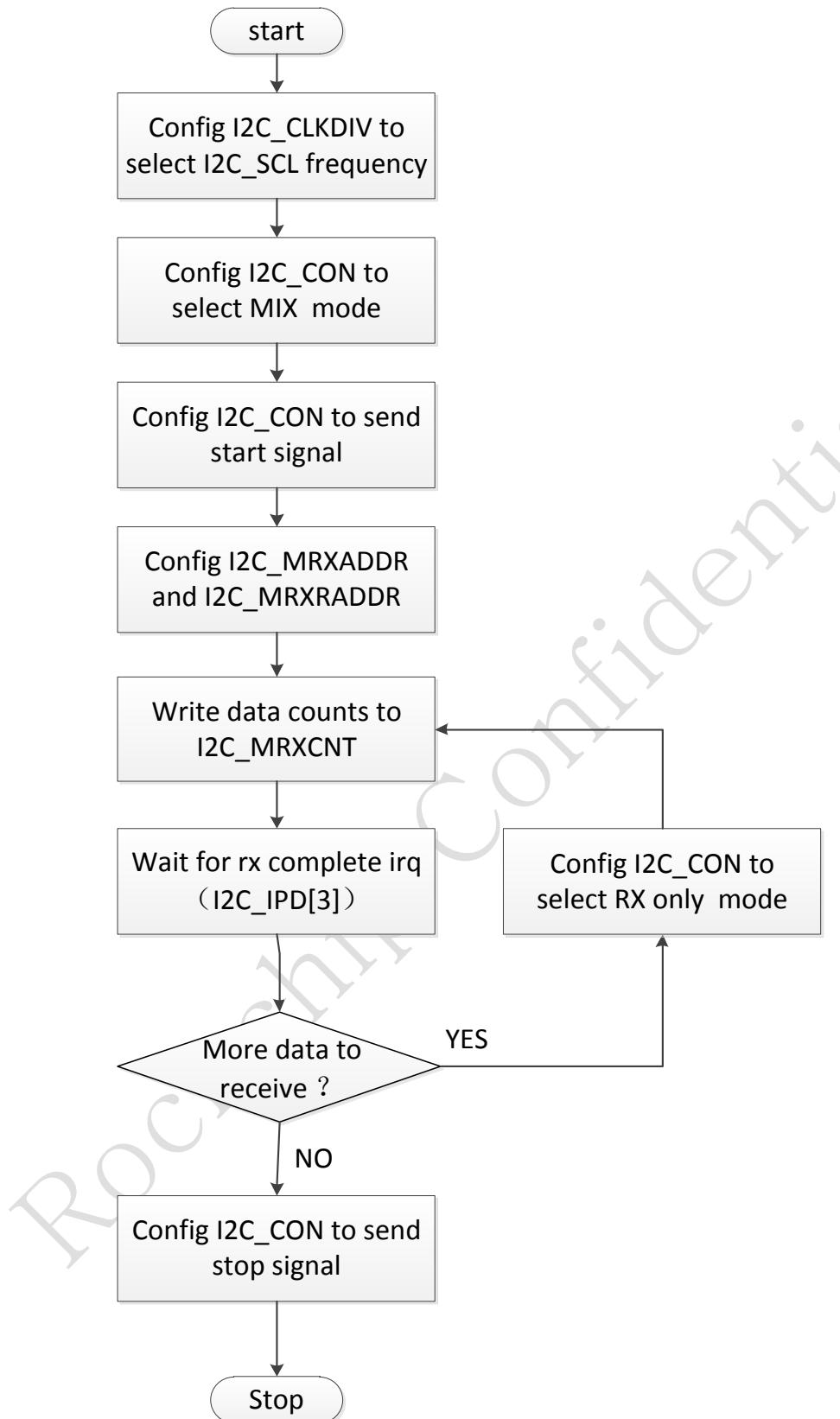


Fig.33-8I2C Flow chat for mix mode

Chapter 34 General-Purpose I/O Ports(GPIO)

34.1 Overview

GPIO is a programmable General Purpose Programming I/O peripheral. This component is a APB slave device. GPIO controls the output data and direction of external I/O pads. It also can read back the data on external pads using memory-mapped registers.

34.1.1 Features

GPIO supports the following features:

- 32 bits APB bus width
- 32 independently configurable signals
- Separate data registers and data direction registers for each signal
- Software control for each signal, or for each bit of each signal
- Configurable interrupt mode for Port A
- Port A has 32 bits

Notes: Port A 32bits are corresponding to port A/B/C/D 8bits in Chapter1

34.2 Block Diagram

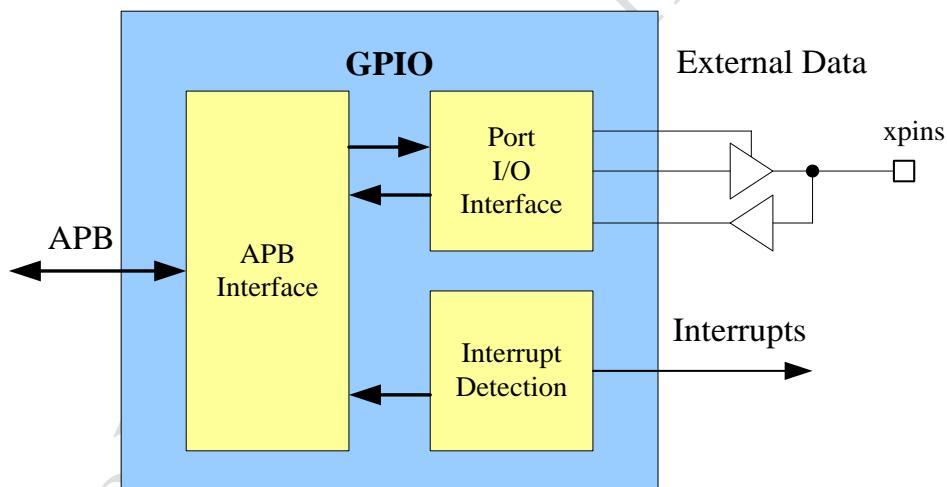


Fig.34-1GPIO block diagram

Block descriptions:

APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

Port I/O Interface

External data Interface to or from I/O pads.

Interrupt Detection

Interrupt interface to or from interrupt controller.

34.3 Function description

34.3.1 Operation

Control Mode(software)

Under software control, the data and direction control for the signal are sourced from the data register (GPIO_SWPORTA_DR) and direction control register (GPIO_SWPORTA_DDR).

The direction of the external I/O pad is controlled by a write to the Porta datadirection register (GPIO_SWPORTA_DDR). The data written to this memory-mapped register gets mapped onto an output signal, GPIO_PORTA_DDR, of the GPIO peripheral. This output signal controls the direction of an external I/O pad.

The data written to the Porta data register (GPIO_SWPORTA_DR) drives the output buffer of the I/O pad. External data are input on the external data signal, GPIO_EXT_PORTA. Reading the external signal register(GPIO_EXT_PORTA) shows the value on the signal, regardless of the direction. This register is read-only, meaning that it cannot be written from the APB software interface.

Reading External Signals

The data on the GPIO_EXT_PORTA external signal can always be read. The data on the external GPIO signal is read by an APB read of the memory-mapped register, GPIO_EXT_PORTA.

An APB read to the GPIO_EXT_PORTA register yields a value equal to that which is on the GPIO_EXT_PORTA signal.

Interrupts

Port A can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge

The interrupts can be masked by programming the GPIO_INTMASK register. The interrupt status can be read before masking (called raw status) and after masking.

The interrupts are combined into a single interrupt output signal, which has the same polarity as the individual interrupts. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.

Whenever Port A is configured for interrupts, the data direction must be set to Input. If the data direction register is reprogrammed to Output, then any pending interrupts are not lost. However, no new interrupts are generated.

For edge-detected interrupts, the ISR can clear the interrupt by writing a 1 to the GPIO_PORTA_EOI register for the corresponding bit to disable the interrupt. This write also clears the interrupt status and raw status registers. Writing to the GPIO_PORTA_EOI register has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll

the GPIO_INT_RAWSTATUS register until the interrupt source disappears, or it can write to the GPIO_INTMASK register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

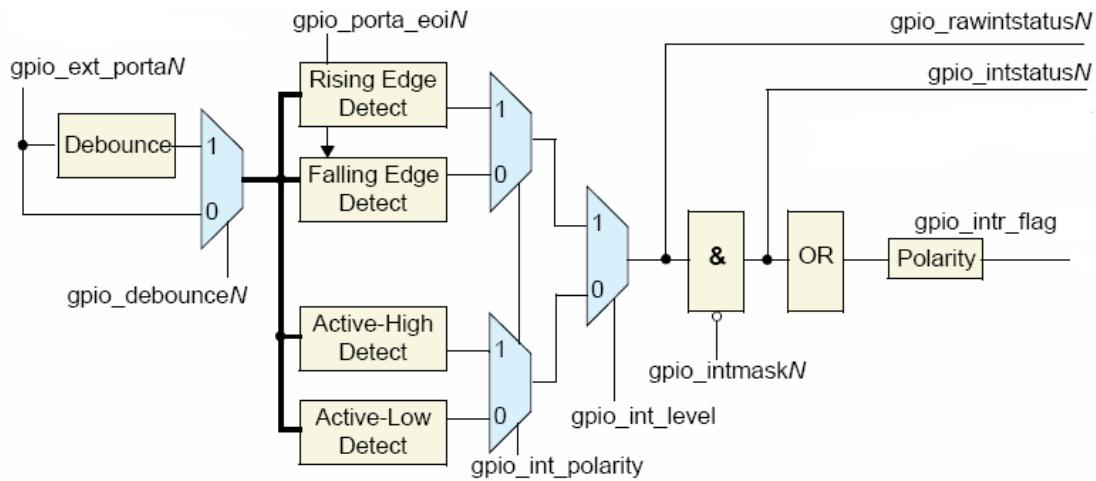


Fig.34-2 GPIO Interrupt RTL Block Diagram

Debounce operation

Port A has been configured to include the debounce capability interrupt feature. The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.

When input interrupt signals are debounced using a debounce clock(pclk), the signals must be active for a minimum of two cycles of the debounce clock to guarantee that they are registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one risingedge, it is not registered.

Synchronization of Interrupt Signals to the System Clock

Interrupt signals are internally synchronized to pclk. Synchronization topclk must occur for edge-detect signals. With level-sensitive interrupts, synchronization is optional and under software control(GPIO_LS_SYNC).

34.3.2 Programming

Programming Considerations

- Reading from an unused location or unused bits in a particular register always returns zeros. There is no error mechanism in the APB.
- Programming the GPIO registers for interrupt capability, edge-sensitive or level-sensitive interrupts, and interrupt polarity should be completed prior to enabling the interrupts on PortA in order to prevent spurious glitches on the interrupt lines to the interrupt controller.
- Writing to the interrupt clear register clears an edge-detected interrupt and has no effect on a level-sensitive interrupt.

4 GPIOs' hierarchy in the chip

GPIO1, GPIO2 are in cpu subsystem, GPIO3 is in peripheral subsystem, and GPIO0 is in alive subsystem.

34.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses. There are 4 GPIOs (GPIO0 ~ GPIO3), and each of them has same register group. Therefore, 4 GPIOs' register groups have 4 different base address.

34.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORTA_DR	0x0000	W	0x00000000	Port A data register
GPIO_SWPORTA_DDR	0x0004	W	0x00000000	Port A data direction register
GPIO_INTEN	0x0030	W	0x00000000	Interrupt enable register
GPIO_INTMASK	0x0034	W	0x00000000	Interrupt mask register
GPIO_INTPOLTYPE_LEVEL	0x0038	W	0x00000000	Interrupt level register
GPIO_INT_POLARITY	0x003c	W	0x00000000	Interrupt polarity register
GPIO_INT_STATUS	0x0040	W	0x00000000	Interrupt status of port A
GPIO_INT_RAWSTATUS	0x0044	W	0x00000000	Raw Interrupt status of port A
GPIO_DEBOUNCE	0x0048	W	0x00000000	Debounce enable register
GPIO_PORTA_EOI	0x004c	W	0x00000000	Port A clear interrupt register
GPIO_EXT_PORTA	0x0050	W	0x00000000	Port A external port register
GPIO_LS_SYNC	0x0060	W	0x00000000	Level_sensitive synchronization enable register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

34.4.2 Detail Register Description

GPIO_SWPORTA_DR

Address: Operational Base + offset (0x0000)

Port A data register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_dr Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORTA_DDR

Address: Operational Base + offset (0x0004)

Port A data direction register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_ddr Values written to this register independently control the direction of the corresponding data bit in Port A. 1'b0: Input (default) 1'b1: Output

GPIO_INTEN

Address: Operational Base + offset (0x0030)

Interrupt enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_en Allows each bit of Port A to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output. 1'b0: Configure Port A bit as normal GPIO signal (default) 1'b1: Configure Port A bit as interrupt

GPIO_INTMASK

Address: Operational Base + offset (0x0034)

Interrupt mask register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_mask Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. 1'b0: Interrupt bits are unmasked (default) 1'b1: Mask interrupt

GPIO_INTPTYPE_LEVEL

Address: Operational Base + offset (0x0038)

Interrupt level register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_inttype_level Controls the type of interrupt that can occur on Port A. 1'b0: Level-sensitive (default) 1'b1: Edge-sensitive

GPIO_INT_POLARITY

Address: Operational Base + offset (0x003c)

Interrupt polarity register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_polarity Controls the polarity of edge or level sensitivity that can occur on input of Port A. 1'b0: Active-low (default) 1'b1: Active-high

GPIO_INT_STATUS

Address: Operational Base + offset (0x0040)

Interrupt status of port A

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_status Interrupt status of Port A

GPIO_INT_RAWSTATUS

Address: Operational Base + offset (0x0044)

Raw Interrupt status of port A

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_rawstatus Raw interrupt of status of Port A (premasking bits)

GPIO_DEBOUNCE

Address: Operational Base + offset (0x0048)

Debounce enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_debounce Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 1'b0: No debounce (default) 1'b1: Enable debounce

GPIO_PORTA_EOI

Address: Operational Base + offset (0x004c)

Port A clear interrupt register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	gpio_porta_eoi Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 1'b0: No interrupt clear (default) 1'b1: Clear interrupt

GPIO_EXT_PORTA

Address: Operational Base + offset (0x0050)

Port A external port register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_ext_porta When Port A is configured as Input, then reading this location reads the values on the signal. When the datadirection of Port A is set as Output, reading this location reads the data register for Port A.

GPIO_LS_SYNC

Address: Operational Base + offset (0x0060)

Level_sensitive synchronization enable register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	gpio_ls_sync Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 1'b0: No synchronization to pclk_intr (default) 1'b1: Synchronize to pclk_intr

34.5 Interface description

Table 34-1 GPIO interface description

Module Pin	Dir	Pad Name	IOMUX Setting
GPIO0 Interface			
gpio0_porta[7:0]	I/O	GPIO0_A[7:0]	GRF_GPIO0A_IOMUX[15:0] =16'h0000
gpio0_porta[15:8]	I/O	GPIO0_B[7:0]	GRF_GPIO0B_IOMUX[15:0] =16'h0000
gpio0_porta[23:16]	I/O	GPIO0_C[7:0]	GRF_GPIO0C_IOMUX[15:0] =16'h0000
gpio0_porta[31:24]	I/O	GPIO0_D[7:0]	GRF_GPIO0D_IOMUX[15:0] =16'h0000
GPIO1 Interface			
gpio1_porta[7:0]	I/O	GPIO1_A[7:0]	GRF_GPIO1A_IOMUX[15:0] =16'h0000
gpio1_porta[15:8]	I/O	GPIO1_B[7:0]	GRF_GPIO1B_IOMUX[15:0]

			=16'h0000
gpio1_porta[23:16]	I/O	GPIO1_C[7:0]	GRF_GPIO1C_IOMUX[15:0] =16'h0000
gpio1_porta[31:24]	I/O	GPIO1_D[7:0]	GRF_GPIO1D_IOMUX[15:0] =16'h0000
GPIO2 Interface			
gpio2_porta[7:0]	I/O	GPIO2_A[7:0]	GRF_GPIO2A_IOMUX[15:0] =16'h0000
gpio2_porta[15:8]	I/O	GPIO2_B[7:0]	GRF_GPIO2B_IOMUX[15:0] =16'h0000
gpio2_porta[23:16]	I/O	GPIO2_C[7:0]	GRF_GPIO2C_IOMUX[7:0]= 8'h00 GRF_GPIO2C_IOMUX2[15:0] =16'h0000
gpio2_porta[31:24]	I/O	GPIO2_D[7:0]	GRF_GPIO2D_IOMUX[15:0] =16'h0000
GPIO3 Interface			
gpio3_porta[7:0]	I/O	GPIO3_A[7:0]	GRF_GPIO3A_IOMUX[15:0] =16'h0000
gpio3_porta[15:8]	I/O	GPIO3_B[7:0]	GRF_GPIO3B_IOMUX[15:0] =16'h0000
gpio3_porta[23:16]	I/O	GPIO3_C[7:0]	GRF_GPIO3C_IOMUX[15:0] =16'h0000
gpio3_porta[31:24]	I/O	GPIO3_D[7:0]	GRF_GPIO3D_IOMUX[15:0] =16'h0000

34.6 Application Notes

Steps to set GPIO's direction

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction and Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Default GPIO's direction is input direction.

Steps to set GPIO's level

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction.
- Write GPIO_SWPORT_DR[x] as v to set this GPIO's value.

Steps to get GPIO's level

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Read from GPIO_EXT_PORT[x] to get GPIO's value

Steps to set GPIO as interrupt source

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Write GPIO_INTTYPE_LEVEL[x] as v1 and write GPIO_INT_POLARITY[x] as v2 to set interrupt type
- Write GPIO_INTEN[x] as 1 to enable GPIO's interrupt

Note: Please switch iomux to GPIO mode first!

Chapter 35 Timer

35.1 Overview

Timer is a programmable timer peripheral. This component is an APBslave device. Timer count down from a programmed value and generate an interrupt when the count reaches zero.

35.1.1 Features

Timers supports the following features:

- One APB timers in the soc system, include six programmable 64 bits timer channel, acts as TIMER0~TIMERS
- Two operation modes: free-running and user-defined count.
- Maskable for each individual interrupt.

35.2 Block Diagram

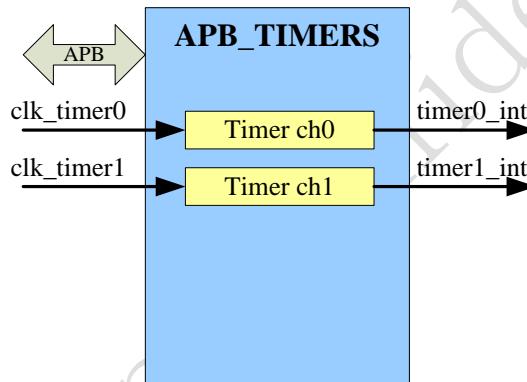


Fig. 35-1 Timers Block Diagram

The above figure shows the architecture of the APB timers (include two programmable timer channel) that in the peripheral subsystem.

35.3 Function description

35.3.1 Timer clock

TIMER0~5 are in the PERI subsystem, using timer ch0 ~ ch5 respectively. The timer clock is 24MHz OSC.

35.3.2 Programming sequence

1. Initialize the timer by the TIMERn_CONTROLREG ($0 \leq n \leq 1$)register:
 - Disable the timer by writing a “0” to the timer enable bit (bit 0). Accordingly, the timer_enoutput signal is de-asserted.
 - Program the timer mode—user-defined or free-running—by writing a “0” or “1” respectively, to the timer mode bit (bit 1).
 - Set the interrupt mask as either masked or not masked by writing a “0” or “1” respectively, to the timer interrupt mask bit (bit 2).
2. Load the timer count value into the TIMERn_LOAD_COUNT1 ($0 \leq n \leq 5$) and TIMERn_LOAD_COUNT0 ($0 \leq n \leq 5$)register.

3. Enable the timer by writing a “1” to bit 0 of TIMERn_CONTROLREG ($0 \leq n \leq 5$).

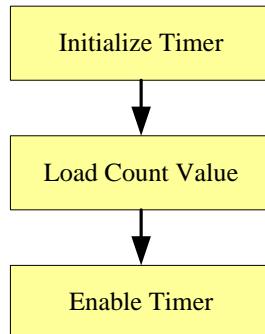


Fig. 35-2 Timer Usage Flow

35.3.3 Loading a timer countvalue

The initial value for each timer—that is, the value from which it counts down—is loaded into the timer using the load count register (TIMERn_LOAD_COUNT1 ($0 \leq n \leq 1$) and TIMERn_LOAD_COUNT0 ($0 \leq n \leq 1$)). Two events can cause a timer to load the initial value from its load count register:

- Timer is enabled after reset or disabled.
- Timer counts down to 0, when timer is configured into free-running mode.

35.3.4 Timer mode selection

- User-defined count mode – Timer loads TIMERn_LOAD_COUNT1($0 \leq n \leq 1$) and TIMERn_LOAD_COUNT0($0 \leq n \leq 1$) register as initial value. Timer will not automatically load the count register, when timer counts down to 0. User need to disable timer firstly and follow the programming sequence to make timer work again.
- Free-running mode – Timer loads the TIMERn_LOAD_COUNT1($0 \leq n \leq 1$) and TIMERn_LOAD_COUNT0($0 \leq n \leq 1$) register as initial value. Timer will automatically load the count register, when timer counts down to 0.

35.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses.

35.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TIMER0_LOAD_COUNT0	0x0000	W	0x00000000	Timer0 Load Count Register
TIMER0_LOAD_COUNT1	0x0004	W	0x00000000	Timer0 Load Count Register
TIMER0_CURRENT_VALUE0	0x0008	W	0x00000000	Timer0 Current Value Register
TIMER0_CURRENT_VALUE1	0x000C	W	0x00000000	Timer0 Current Value Register
TIMER0_CONTROLREG	0x0010	W	0x00000000	Timer0 Control Register

TIMER0_INTSTATUS	0x0018	W	0x00000000	Timer0 Interrupt Status Register
TIMER1_LOAD_COUNT0	0x0020	W	0x00000000	Timer1 Load Count Register
TIMER1_LOAD_COUNT1	0x0024	W	0x00000000	Timer1 Load Count Register
TIMER1_CURRENT_VALUE0	0x0028	W	0x00000000	Timer1 Current Value Register
TIMER1_CURRENT_VALUE1	0x002c	W	0x00000000	Timer1 Current Value Register
TIMER1_CONTROLREG	0x0030	W	0x00000000	Timer1 Control Register
TIMER1_INTSTATUS	0x0038	W	0x00000000	Timer1 Interrupt Status Register
TIMER2_LOAD_COUNT0	0x0020	W	0x00000000	Timer1 Load Count Register
TIMER2_LOAD_COUNT1	0x0024	W	0x00000000	Timer1 Load Count Register
TIMER2_CURRENT_VALUE0	0x0028	W	0x00000000	Timer1 Current Value Register
TIMER2_CURRENT_VALUE1	0x002c	W	0x00000000	Timer1 Current Value Register
TIMER2_CONTROLREG	0x0030	W	0x00000000	Timer1 Control Register
TIMER2_INTSTATUS	0x0038	W	0x00000000	Timer1 Interrupt Status Register
TIMER3_LOAD_COUNT0	0x0020	W	0x00000000	Timer1 Load Count Register
TIMER3_LOAD_COUNT1	0x0024	W	0x00000000	Timer1 Load Count Register
TIMER3_CURRENT_VALUE0	0x0028	W	0x00000000	Timer1 Current Value Register
TIMER3_CURRENT_VALUE1	0x002c	W	0x00000000	Timer1 Current Value Register
TIMER3_CONTROLREG	0x0030	W	0x00000000	Timer1 Control Register
TIMER3_INTSTATUS	0x0038	W	0x00000000	Timer1 Interrupt Status Register
TIMER4_LOAD_COUNT0	0x0020	W	0x00000000	Timer1 Load Count Register
TIMER4_LOAD_COUNT1	0x0024	W	0x00000000	Timer1 Load Count Register
TIMER4_CURRENT_VALUE0	0x0028	W	0x00000000	Timer1 Current Value Register
TIMER4_CURRENT_VALUE1	0x002c	W	0x00000000	Timer1 Current Value Register
TIMER4_CONTROLREG	0x0030	W	0x00000000	Timer1 Control Register
TIMER4_INTSTATUS	0x0038	W	0x00000000	Timer1 Interrupt Status Register
TIMER5_LOAD_COUNT0	0x0020	W	0x00000000	Timer1 Load Count Register
TIMER5_LOAD_COUNT1	0x0024	W	0x00000000	Timer1 Load

				Count Register
TIMER5_CURRENT_VALUE0	0x0028	W	0x00000000	Timer1 Current Value Register
TIMER5_CURRENT_VALUE1	0x002c	W	0x00000000	Timer1 Current Value Register
TIMER5_CONTROLREG	0x0030	W	0x00000000	Timer1 Control Register
TIMER5_INTSTATUS	0x0038	W	0x00000000	Timer1 Interrupt Status Register

Notes: Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** – WORD (32 bits) access

35.4.2 Detail Register Description

TIMERn_LOAD_COUNT0

Address: Operational Base + offset(0x00+n*0x20)

Timer n Load Count Register($0 \leq n \leq 1$)

Bit	Attr	Reset Value	Description
31:0	RW	0x0	Low 32 bits Value to be loaded into Timer n. This is the value from which counting commences.

TIMERn_LOAD_COUNT1

Address: Operational Base + offset(0x04+n*0x20)

Timer n Load Count Register($0 \leq n \leq 1$)

Bit	Attr	Reset Value	Description
31:0	RW	0x0	High 32 bits Value to be loaded into Timer n. This is the value from which counting commences.

TIMERn_CURRENT_VALUE0

Address: Operational Base + offset(0x08+n*0x20)

Timer n Current Value Register($0 \leq n \leq 1$)

Bit	Attr	Reset Value	Description
31:0	R	0x0	Low 32 bits of Current Value of Timer n.

TIMERn_CURRENT_VALUE1

Address: Operational Base + offset(0x0c+n*0x20)

Timer n Current Value Register($0 \leq n \leq 1$)

Bit	Attr	Reset Value	Description
31:0	R	0x0	High 32 bits of Current Value of Timer n.

TIMERn_CONTROLREG

Address: Operational Base + offset(0x10+n*0x20)

Timer n Control Register($0 \leq n \leq 1$)

Bit	Attr	Reset Value	Description
31:3	-	-	Reserved
2	RW	0x0	Timer interrupt mask. 1'b0: mask 1'b1: not mask
1	RW	0x0	Timer mode. 1'b0: free-running mode 1'b1: user-defined count mode
0	RW	0x0	Timer enable.

			1'b0: disable 1'b1: enable
--	--	--	-------------------------------

TIMERn_INTSTATUS

Address: Operational Base + offset(0x18+n*0x20)

Timer n Interrupt Status Register($0 \leq n \leq 1$)

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	This register contains the interrupt status for timer n Write 1 to this register will clear the interrupt

Notes: Attr: **RW** – Read/writable, **R** – read only, **W** – write only**35.5 Application Notes**

In the chip, the timer_clk is from 24MHz OSC, asynchronous to the pclk. When user disable the timer enable bit (bit 0 of TIMERn_CONTROLREG($0 \leq n \leq 5$)), the timer_en output signal is de-asserted, and timer_clk will stop. When user enable the timer, the timer_en signal is asserted and timer_clk will start running.

The application is only allowed to re-config registers when timer_en is low.

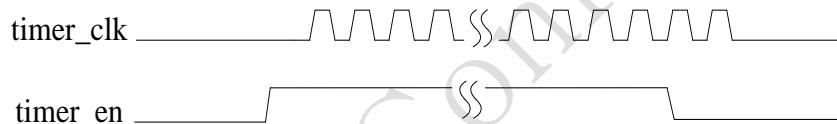


Fig. 35-3 Timing between timer_en and timer_clk

Please refer to funciton description section for the timer usage flow.

Chapter 36 Pulse Width Modulation(PWM)

36.1 Overview

The pulse-width modulator (PWM) feature is very common in embedded systems. It provides a way to generate a pulse periodic waveform for motor control or can act as a digital-to-analog converter with some external components.

36.1.1 Features

The PWMModule supports the following features:

- 4-built-in PWM channels
- Configurable to operate in capture mode
 - Measures the high/low polarity effective cycles of this input waveform
 - Generates a single interrupt at the transition of input waveform polarity
 - 32-bit high polarity capture register
 - 32-bit low polarity capture register
 - 32-bit current value register
- Configurable to operate in continuous mode or one-shot mode
 - 32-bit period counter
 - 32-bit duty register
 - 32-bit current value register
 - Configurable PWM output polarity in inactive state and duty period pulse polarity
 - Period and duty cycle are shadow buffered. Change takes effect when the end of the effective period is reached or when the channel is disabled
 - Programmable center or left aligned outputs, and change takes effect when the end of the effective period is reached or when the channel is disabled
 - 8-bit repeat counter for one-shot operation. One-shot operation will produce $N + 1$ periods of the waveform, where N is the repeat counter value, and generates a single interrupt at the end of operation
 - Continuous mode generates the waveform continuously, and do not generates any interrupts
- pre-scaled operation to bus clock and then further scaled
- Available low-power mode to reduce power consumption when the channel is inactive.

36.2 Block Diagram

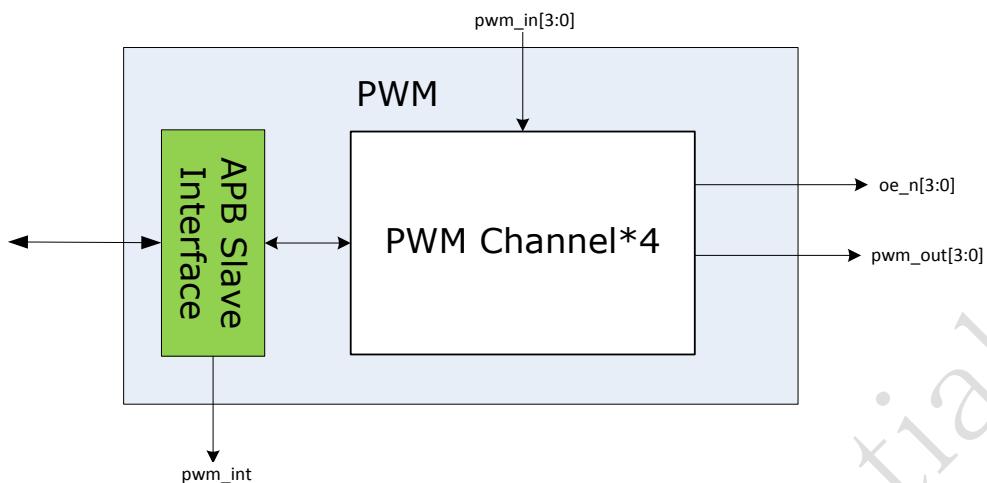


Fig.36-1 PWM architecture

36.2.1 PWM APB Slave Interface

The host processor gets access to PWM Register Block through the APB slave interface with 32-bit bus width, and asserts the active-high level interrupt.

36.2.2 PWM Channels

This is the control logic of PWM module, and controls the operation of PWM module according to the configured working mode.

36.3 Function description

The PWM supports three operation modes: reference mode, one-shot mode and continuous mode. For the one-shot mode and the continuous mode, the PWM output can be configured as the left-aligned mode or the center-aligned mode.

36.3.1 Reference mode

The reference mode is used to measure the PWM channel input waveform high/low effective cycles with the PWM channel clock, and asserts an interrupt when the polarity of the input waveform changes. The number of the high effective cycles is recorded in the `PWMx_PERIOD_HPC` register, while the number of the low effective cycles is recorded in the `PWMx_DUTY_LPC` register.

Note: the PWM input waveform is doubled buffered when the PWM channel is working in order to filter unexpected shot-time polarity transition, and therefore the interrupt is asserted several cycles after the input waveform polarity changes, and so does the change of the values of PWMx_PERIOD_HPC and PWMx_DUTY_LPC.

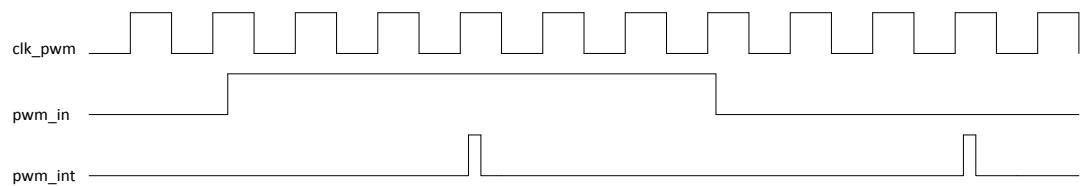


Fig.36-2PWM Reference Mode

36.3.2 Continuous Mode

The PWM channel generates a series of the pulses continuously as expected once the channel is enabled with continuous mode.

In the continuous mode, the PWM output waveforms can be in one form of the two output mode: left-aligned mode or center-aligned mode.

For the left-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. After the period number (PWMx_PERIOD_HPC) is reached, the output is again switched to the opposite polarity to start another period of desired pulse.

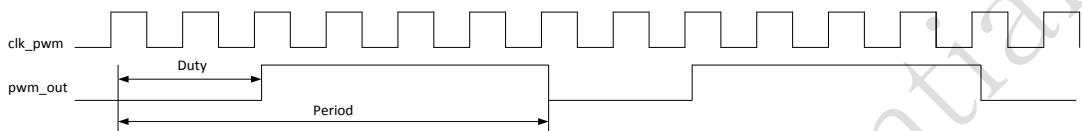


Fig.36-3PWM Left-aligned Output Mode

For the center-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once one half of duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. Then if there is one half of duty cycle left for the whole period , the output is again switched to the opposite polarity. Finally after the period number (PWMx_PERIOD_HPC) is reached, the output is again switched to the opposite polarity to start another period of desired pulse.

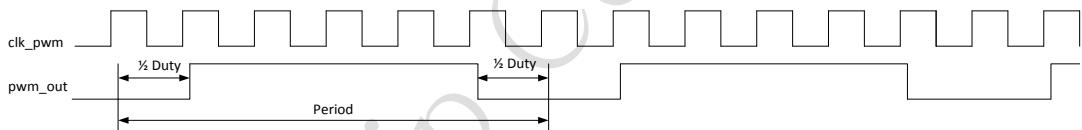


Fig.36-4PWM Center-aligned Output Mode

Disable the PWM channel, the channel stops generating the output waveforms and output polarity is fixed as the configured inactive polarity (PWMx_CTRL.inactive_pol).

36.3.3 One-shot Mode

Unlike the continuous mode, the PWM channel generates the output waveforms within the configured periods (PWM_CTRL.rpt + 1), and then stops. At the same times, an interrupt is asserted to inform that the operation has been finished.

There are also two output modes for the one-shot mode: the left-aligned mode and the center-aligned mode.

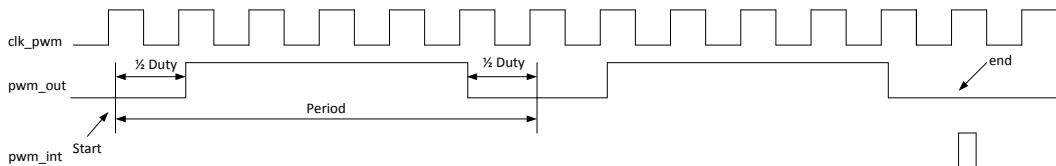


Fig.36-5PWM Center-aligned Output Mode

36.4 Register description

36.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
PWM_PWM0_CNT	0x0000	W	0x00000000	PWM Channel 0 Counter Register
PWM_PWM0_PERIOD_HPR	0x0004	W	0x00000000	PWM Channel 0 Period Register/High Polarity Capture Register
PWM_PWM0_DUTY_LPR	0x0008	W	0x00000000	PWM Channel 0 Duty Register/Low Polarity Capture Register
PWM_PWM0_CTRL	0x000c	W	0x00000000	PWM Channel 0 Control Register
PWM_PWM1_CNT	0x0010	W	0x00000000	PWM Channel 1 Counter Register
PWM_PWM1_PERIOD_HPR	0x0014	W	0x00000000	PWM Channel 1 Period Register/High Polarity Capture Register
PWM_PWM1_DUTY_LPR	0x0018	W	0x00000000	PWM Channel 1 Duty Register/Low Polarity Capture Register
PWM_PWM1_CTRL	0x001c	W	0x00000000	PWM Channel 1 Control Register
PWM_PWM2_CNT	0x0020	W	0x00000000	PWM Channel 2 Counter Register
PWM_PWM2_PERIOD_HPR	0x0024	W	0x00000000	PWM Channel 2 Period Register/High Polarity Capture Register
PWM_PWM2_DUTY_LPR	0x0028	W	0x00000000	PWM Channel 2 Duty Register/Low Polarity Capture Register
PWM_PWM2_CTRL	0x002c	W	0x00000000	PWM Channel 2 Control Register
PWM_PWM3_CNT	0x0030	W	0x00000000	PWM Channel 3 Counter Register

Name	Offset	Size	Reset Value	Description
PWM_PWM3_PERIOD_HPR	0x0034	W	0x00000000	PWM Channel 3 Period Register/High Polarity Capture Register
PWM_PWM3_DUTY_LPR	0x0038	W	0x00000000	PWM Channel 3 Duty Register/Low Polarity Capture Register
PWM_PWM3_CTRL	0x003c	W	0x00000000	PWM Channel 3 Control Register
PWM_INTSTS	0x0040	W	0x00000000	Interrupt Status Register
PWM_INT_EN	0x0044	W	0x00000000	Interrupt Enable Register

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

36.4.2 Detail Register Description

PWM_PWM0_CNT

Address: Operational Base + offset (0x0000)

PWM Channel 0 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 0 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_PERIOD_HPR

Address: Operational Base + offset (0x0004)

PWM Channel 0 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERIOD_LPR</p> <p>Output Waveform Period/Input Waveform High Polarity Cycle</p> <p>If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0.</p> <p>If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock.</p> <p>The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM0_DUTY_LPR

Address: Operational Base + offset (0x0008)

PWM Channel 0 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>DUTY_LPR</p> <p>Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle</p> <p>If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account.</p> <p>If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM0_CTRL

Address: Operational Base + offset (0x000c)

PWM Channel 0 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This fields defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	<p>output_mode PWM Output mode 0: left aligned mode 1: center aligned mode</p>
4	RW	0x0	<p>inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive</p>
3	RW	0x0	<p>duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive</p>
2:1	RW	0x0	<p>pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt . 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved</p>
0	RW	0x0	<p>pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked the one-shot mode, this bit will be cleared at the end of operation</p>

PWM_PWM1_CNT

Address: Operational Base + offset (0x0010)

PWM Channel 1 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 1 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to ($2^{32}-1$).

PWM_PWM1_PERIOD_HPR

Address: Operational Base + offset (0x0014)

PWM Channel 1 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_LPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).

PWM_PWM1_DUTY_LPR

Address: Operational Base + offset (0x0018)

PWM Channel 1 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle</p> <p>If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account.</p> <p>If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM1_CTRL

Address: Operational Base + offset (0x001c)

PWM Channel 1 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	<p>rpt Repeat Counter</p> <p>This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.</p>
23:16	RW	0x00	<p>scale Scale Factor</p> <p>This fields defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2^N. If N is 0, it means that the clock is divided by 512(2^{19}).</p>
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	<p>prescale Prescale Factor</p> <p>This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.</p>
11:10	RO	0x0	reserved
9	RW	0x0	<p>clk_sel Clock Source Select</p> <p>0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source</p> <p>1: scaled clock is selected as PWM clock source</p>
8	RW	0x0	<p>lp_en Low Power Mode Enable</p> <p>0: disabled</p> <p>1: enabled</p> <p>When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.</p>
7:6	RO	0x0	reserved
5	RW	0x0	<p>output_mode PWM Output mode</p> <p>0: left aligned mode</p> <p>1: center aligned mode</p>
4	RW	0x0	<p>inactive_pol Inactive State Output Polarity</p> <p>This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled.</p> <p>0: negative</p> <p>1: positive</p>

Bit	Attr	Reset Value	Description
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM2_CNT

Address: Operational Base + offset (0x0020)

PWM Channel 2 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 2 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_PERIOD_HPR

Address: Operational Base + offset (0x0024)

PWM Channel 2 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERIOD_LPR Output Waveform Period/Input Waveform High Polarity Cycle</p> <p>If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0.</p> <p>If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM2_DUTY_LPR

Address: Operational Base + offset (0x0028)

PWM Channel 2 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle</p> <p>If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account.</p> <p>If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM2_CTRL

Address: Operational Base + offset (0x002c)

PWM Channel 2 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This fields defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source

Bit	Attr	Reset Value	Description
8	RW	0x0	<p>lp_en Low Power Mode Enable 0: disabled 1: enabled</p> <p>When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.</p>
7:6	RO	0x0	reserved
5	RW	0x0	<p>output_mode PWM Output mode 0: left aligned mode 1: center aligned mode</p>
4	RW	0x0	<p>inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive</p>
3	RW	0x0	<p>duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive</p>

Bit	Attr	Reset Value	Description
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM3_CNT

Address: Operational Base + offset (0x0030)

PWM Channel 3 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 3 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_PERIOD_HPR

Address: Operational Base + offset (0x0034)

PWM Channel 3 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERIOD_LPR</p> <p>Output Waveform Period/Input Waveform High Polarity Cycle</p> <p>If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0.</p> <p>If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM3_DUTY_LPR

Address: Operational Base + offset (0x0038)

PWM Channel 3 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>DUTY_LPR</p> <p>Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle</p> <p>If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account.</p> <p>If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM3_CTRL

Address: Operational Base + offset (0x003c)

PWM Channel 3 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This fields defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	<p>output_mode PWM Output mode 0: left aligned mode 1: center aligned mode</p>
4	RW	0x0	<p>inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive</p>
3	RW	0x0	<p>duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive</p>
2:1	RW	0x0	<p>pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved</p>
0	RW	0x0	<p>pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked the one-shot mode, this bit will be cleared at the end of operation</p>

PWM_INTSTS

Address: Operational Base + offset (0x0040)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11	RO	0x0	<p>CH3_Pol Channel 3 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM3_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM3_PERIOD_HPR to know the effective low cycle of Channel 3 input waveform. Write 1 to CH3_IntSts will clear this bit.</p>
10	RO	0x0	<p>CH2_Pol Channel 2 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM2_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM2_PERIOD_HPR to know the effective low cycle of Channel 2 input waveform. Write 1 to CH2_IntSts will clear this bit.</p>
9	RO	0x0	<p>CH1_Pol Channel 1 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM1_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM1_PERIOD_HPR to know the effective low cycle of Channel 1 input waveform. Write 1 to CH1_IntSts will clear this bit.</p>

Bit	Attr	Reset Value	Description
8	RO	0x0	<p>CH0_Pol Channel 0 Interrupt Polarity Flag</p> <p>This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM0_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM0_PERIOD_HPR to know the effective low cycle of Channel 0 input waveform. Write 1 to CH0_IntSts will clear this bit.</p>
7:4	RO	0x0	reserved
3	RW	0x0	<p>CH3_IntSts Channel 3 Interrupt Status</p> <p>0: Channel 3 Interrupt not generated 1: Channel 3 Interrupt generated</p>
2	RW	0x0	<p>CH2_IntSts Channel 2 Interrupt Status</p> <p>0: Channel 2 Interrupt not generated 1: Channel 2 Interrupt generated</p>
1	RW	0x0	<p>CH1_IntSts Channel 1 Interrupt Status</p> <p>0: Channel 1 Interrupt not generated 1: Channel 1 Interrupt generated</p>
0	RW	0x0	<p>CH0_IntSts Channel 0 Raw Interrupt Status</p> <p>0: Channel 0 Interrupt not generated 1: Channel 0 Interrupt generated</p>

PWM_INT_EN

Address: Operational Base + offset (0x0044)

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	CH3_Int_en Channel 3 Interrupt Enable 0: Channel 3 Interrupt disabled 1: Channel 3 Interrupt enabled
2	RW	0x0	CH2_Int_en Channel 2 Interrupt Enable 0: Channel 2 Interrupt disabled 1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en Channel 1 Interrupt Enable 0: Channel 1 Interrupt disabled 1: Channel 1 Interrupt enabled
0	RW	0x0	CH0_Int_en Channel 0 Interrupt Enable 0: Channel 0 Interrupt disabled 1: Channel 0 Interrupt enabled

36.5 Interface Description

Table 36-1 PWM Interface Description

Module Pin	IO	Pad Name	IOMUX Setting
pwm3	O	PWMMirin_GPIO3d2	GPIO3D_IOMUX[4]= 1'b1
pwm2	O	PWM2_GPIO0d4	GPIO0D_IOMUX[8]= 1'b1
pwm1	O	PWM1_GPIO0d3	GPIO0D_IOMUX[6]= 1'b1
pwm0	O	PWM0_GPIO0d2	GPIO0D_IOMUX[4]= 1'b1

Notes: 1. I=Input, O=Output, I/O=Input/Output, bidirectional

2. There are two sets of IOs for each PWM channel

36.6 Application Notes

36.6.1 PWM Reference Mode Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Configure the channel to work in the reference mode.

4. Enable the INT_EN.chx_int_en to enable the interrupt generation.
5. Enable the channel by writing '1' to PWMx_CTRL.pwm_en bit to start the channel.
6. When an interrupt is asserted, refer to INTSTS register to know the raw interrupt status. If the corresponding polarity flag is set, turn to PWMx_PERIOD_HPC register to know the effective high cycles of input waveforms, otherwise turn to PWMx_DUTY_LPC register to know the effective low cycles.
7. Write '0' to PWMx_CTRL.pwm_en to disable the channel.

36.6.2 PWM One-shotMode/Continuous Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Choose the output mode by setting PWMx_CTRL.output_mode, and set the duty polarity and inactive polarity by programming PWMx_CTRL.duty_pol and PWMx_CTRL.inactive_pol.
4. Set the PWMx_CTRL.rpt if the channel is desired to work in the one-shot mode;
5. Configure the channel to work in the one-shot mode or the continuous mode.
6. Enable the INT_EN.chx_int_en to enable the interrupt generation if the channel is desired to work in the one-shot mode;
7. If the channel is working in the one-shot mode, an interrupt is asserted after the end of operation, and the PWMx_CTRL.pwm_en is automatically cleared. Whatever mode the channel is working, write '0' to PWMx_CTRL.pwm_en bit to disable the PWM channel.

36.6.3 Low-power mode

Setting PWMx_CTRL.ip_en to '1' makes the channel enter the low-power mode. When the PWM channel is inactive, the APB bus clock to the PWM channel is gated in order to reduce the power consumption. It is recommended to disable the channel before entering the low-power mode, and quit the low-power mode before enabling the channel.

36.6.4 Other notes

When the channel is active to produce waveforms, it is free to programming the PWMx_PERIOD_HPC and PWMx_DUTY_LPC register. The change will not take effect immediately until the current period ends.

An active channel can be changed to another operation mode without disable the PWM channel. However, during the transition of the operation mode there may be some irregular output waveforms.

If the PWM operational frequency is desired to changed, it is recommended to disable the channel first, and then make the channel enter the low-power mode to gate the clock. It is free to change clock setting. After clock setting is changed, quit the lower-power mode and enable the channel to take the change into effect.

Chapter 37 WatchDog

37.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that may be caused by conflicting parts or programs in a SoC. The WDT would generate interrupt or reset signal when its counter reaches zero, then a reset controller would reset the system.

WDT supports the following features:

- 32 bits APB bus width
- WDT counter's clock is pclk
- 32 bits WDT counter width
- Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
- WDT can perform two types of operations when timeout occurs:
 - Generate a system reset
 - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
- Programmable reset pulse length
- Total 16 defined-ranges of main timeout period

37.2 Block Diagram

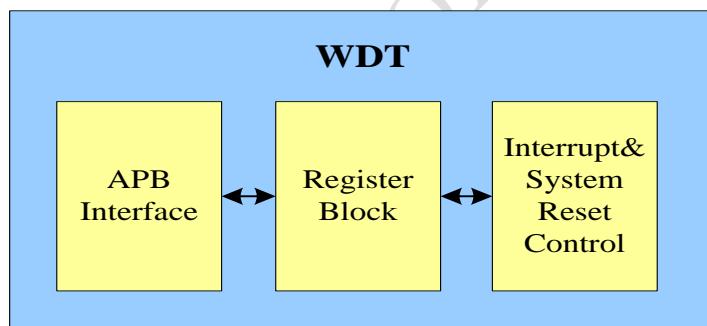


Fig. 37-1 WDT block diagram

Block Descriptions:

APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

Register Block

A register block that handles coherence for the current count register.

Interrupt & system reset control

An interrupt/system reset generation block comprising of a decrementing counter and control logic.

37.3 Function description

37.3.1 Operation

Counter

The WDT counts from a preset (timeout) value in descending order to zero. When the counter reaches zero, depending on the output response mode selected, either a system reset or an interrupt occurs. When the counter reaches zero, it wraps to the selected timeout value and continues decrementing. The user can restart the counter to its initial value. This is programmed by writing to the restart register at any time. The process of restarting the watchdog counter is sometimes referred as kicking the dog. As a safety feature to prevent accidental restarts, the value 0x76 must be written to the Current Counter Value Register (WDT_CRR).

Interrupts

The WDT can be programmed to generate an interrupt (and then a system reset) when a timeout occurs. When a 1 is written to the response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates an interrupt. If it is not cleared by the time a second timeout occurs, then it generates a system reset. If a restart occurs at the same time the watchdog counter reaches zero, an interrupt is not generated.

System Resets

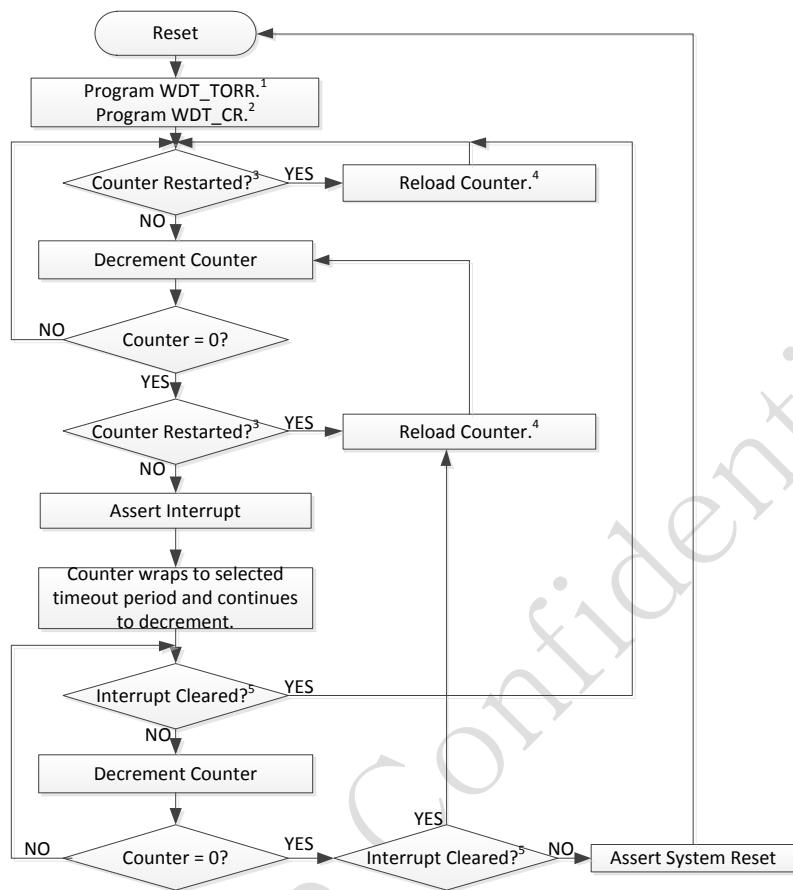
When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates a system reset when a timeout occurs.

Reset Pulse Length

The reset pulse length is the number of pclk cycles for which a system reset is asserted. When a system reset is generated, it remains asserted for the number of cycles specified by the reset pulse length or until the system is reset. A counter restart has no effect on the system reset once it has been asserted.

37.3.2 Programming sequence

Operation Flow Chart (Response mode=1)



1. Select required timeout period.
2. Set reset pulse length, response mode, and enable WDT.
3. Write 0x76 to WDT_CRR.
4. Starts back to selected timeout period.
5. Can clear by reading WDT_EOI or restarting (kicking) the counter by writing 0x76 to WDT_CRR.

Fig. 37-2 WDT Operation Flow

37.4 Register Description

This section describes the control/status registers of the design.

37.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout range Register
WDT_CCVR	0x0008	W	0x00000000	Current counter value Register
WDT_CRR	0x000c	W	0x00000000	Counter restart Register
WDT_STAT	0x0010	W	0x00000000	Interrupt status Register
WDT_EOI	0x0014	W	0x00000000	Interrupt clear Register

Notes: **S**- Size: **B**- Byte (8 bits) access, **H**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

37.4.2 Detail Register Description

WDT_CR

Address: Operational Base + offset (0x0000)

Control Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:2	RW	0x2	<p>rst_pluse_lenth Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted.</p> <p>3'b000: 2 pclk cycles 3'b001: 4 pclk cycles 3'b010: 8 pclk cycles 3'b011: 16 pclk cycles 3'b100: 32 pclk cycles 3'b101: 64 pclk cycles 3'b110: 128 pclk cycles 3'b111: 256 pclk cycles</p>
1	RW	0x1	<p>resp_mode Response mode. Selects the output response generated to a timeout.</p> <p>1'b0: Generate a system reset 1'b1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset</p>
0	RW	0x0	<p>wdt_en WDT enable</p> <p>1'b0: WDT disabled 1'b1: WDT enabled</p>

WDT_TORR

Address: Operational Base + offset (0x0004)

Timeout range Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	<p>timeout_period Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for a 32-bit watchdog counter are:</p> <p>4'b0000: 0x0000ffff 4'b0001: 0x0001ffff 4'b0010: 0x0003ffff 4'b0011: 0x0007ffff 4'b0100: 0x000fffff 4'b0101: 0x001fffff 4'b0110: 0x003fffff 4'b0111: 0x007fffff 4'b1000: 0x00ffffff 4'b1001: 0x01ffffff 4'b1010: 0x03ffffff 4'b1011: 0x07ffffff 4'b1100: 0x0fffffff 4'b1101: 0x1fffffff 4'b1110: 0x3fffffff 4'b1111: 0x7fffffff</p>

WDT_CCVR

Address: Operational Base + offset (0x0008)

Current counter value Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>cur_cnt Current counter value This register, when read, is the current value of the internal counter. This value is read coherently when ever it is read</p>

WDT_CRR

Address: Operational Base + offset (0x000c)

Counter restart Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	W1C	0x00	<p>cnt_restart Counter restart This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.</p>

WDT_STAT

Address: Operational Base + offset (0x0010)

Interrupt status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_status This register shows the interrupt status of the WDT. 1'b1: Interrupt is active regardless of polarity. 1'b0: Interrupt is inactive.

WDT_EOI

Address: Operational Base + offset (0x0014)

Interrupt clear Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_int_clr Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.

37.5 Application Notes

Please refer to the function description section.

Chapter 38 SAR-ADC

38.1 Overview

38.1.1 Features

The ADC is a 4-channel (0/1/2/6) signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as its reference which avoid the use of any external reference. It converts the analog input signal into 10-bit binary digital codes at a maximum conversion rate of 100KSPS with 1MHz A/D converter clock.

38.2 Block Diagram

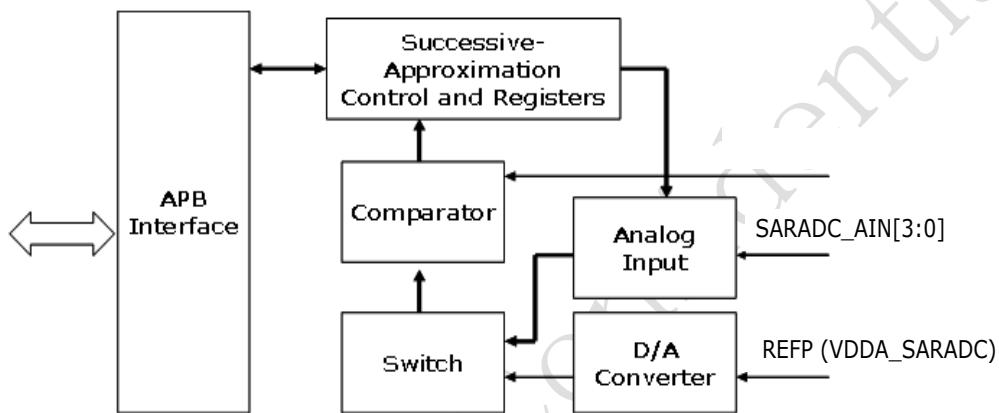


Fig.38-1RKAudiSAR-ADC block diagram

Successive-Approximate Register and Control Logic Block

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.

Comparator Block

This block compares the analog input `SARADC_AIN[3:0]` with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

38.3 Function description

In RKAudi, SAR-ADC works at single-sample operation mode.

This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC. The detailed timing diagram will be shown in the following.

38.4 Register Description

This section describes the control/status registers of the design.

38.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion.
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter.
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	delay between power up and start command

Notes: **Size:** **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

38.4.2 Detail Register Description

SARADC_DATA

Address: Operational Base + offset (0x0000)

This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RO	0x000	adc_data A/D value of the last conversion (DOUT[9:0]).

SARADC_STAS

Address: Operational Base + offset (0x0004)

The status register of A/D Converter.

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	adc_status ADC status (EOC) 1'b0: ADC stop 1'b1: Conversion in progress

SARADC_CTRL

Address: Operational Base + offset (0x0008)

The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	int_status Interrupt status. This bit will be set to 1 when endofconversion. Set 0 to clear the interrupt.

Bit	Attr	Reset Value	Description
5	RW	0x0	int_en Interrupt enable. 1'b0: Disable 1'b1: Enable
4	RO	0x0	reserved
3	RW	0x0	adc_power_ctrl ADC power down control bit 1'b0: ADC power down 1'b1: ADC power up and reset start signal will be asserted (DLY_PU_SOC+2) sclk clock period later after power up
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 3'b000: Input source 0 (SARADC_AIN[0]) 3'b001: Input source 1 (SARADC_AIN[1]) 3'b010: Input source 2 (SARADC_AIN[2]) 3'b110: Input source 3 (SARADC_AIN[3]) Others : Reserved

SARADC_DLY_PU_SOC

Address: Operational Base + offset (0x000c)
delay between power up and start command

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x08	DLY_PU_SOC delay between power up and start command The start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up

38.5 Timing Diagram

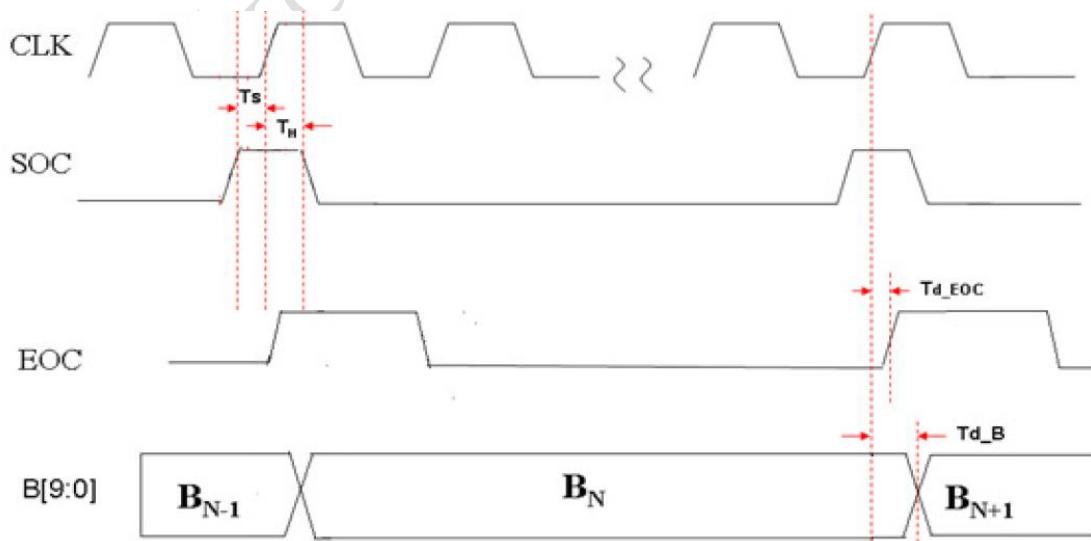


Fig.38-2SAR-ADC timing diagram in single-sample conversion mode

The following table has shows the detailed value for timing parameters in the

above diagram.

Table 38-1RKAudi SAR-ADC timing parameters list

Timing	Symbol	Value			Unit	Description
		Min	Typ	Max		
Data Setup	T _S	0			ns	Setup time for SOC
Data Hold	T _H	10			ns	Hold time for SOC
EOC delay	T _{D_EOC}			10.9	ns	EOC delay from rising edge of CLK
Data delay	T _{d_B}			11.1	ns	B[9:0] delay from rising edge of CLK
CLK period		90			ns	CLK period

38.6 Application Notes

Steps of adc conversion:

- Write SARADC_CTRL[3] as 0 to power down adc converter.
- Write SARADC_CTRL[2:0] as n to select adc channel(n).
- Write SARADC_CTRL[5] as 1 to enable adc interrupt.
- Write SARADC_CTRL[3] as 1 to power up adc converter.
- Wait for adc interrupt or poll SARADC_STAS register to assert whether the conversion is completed
- Read the conversion result from SARADC_DATA[9:0]

Note: The A/D converter was designed to operate at maximum 1MHZ.

Chapter 39 eFuse

39.1 Overview

In RKaudi, eFuse is a parallel-in/parallel-out Electrical Fuse Macro IP which has 512 bits internal nonvolatile one-time programmable EFUSE storage. With a serial interface, 1-bit can be programmed each time in program mode and 8-bit can be read at one time in read mode.

39.1.1 Features

The main features are as follows:

- One-time programmable nonvolatile EFUSE storage cells organized as 64x8 bit
- 1.1V typical core voltage (DVDD)
- The accumulative total time of $1.21V < AVDD \leq 2.75V$ must NOT exceed 1s.
- AVDD is NOT allowed to exceed 2.75V, otherwise there will be reliability issue
- Operating junction temperature range of -40°C to 125°C (for reading)
- Burning requirements:
 - 2.5V typical burning voltage (AVDD), AVDD must be high during PGM mode. AVDD must be low or floating during READ mode and inactive mode
 - 2us burning pulse width
 - Ambient temperature range of 10~40°C
 - Burning at wafer, package, or field level

39.2 Block Diagram

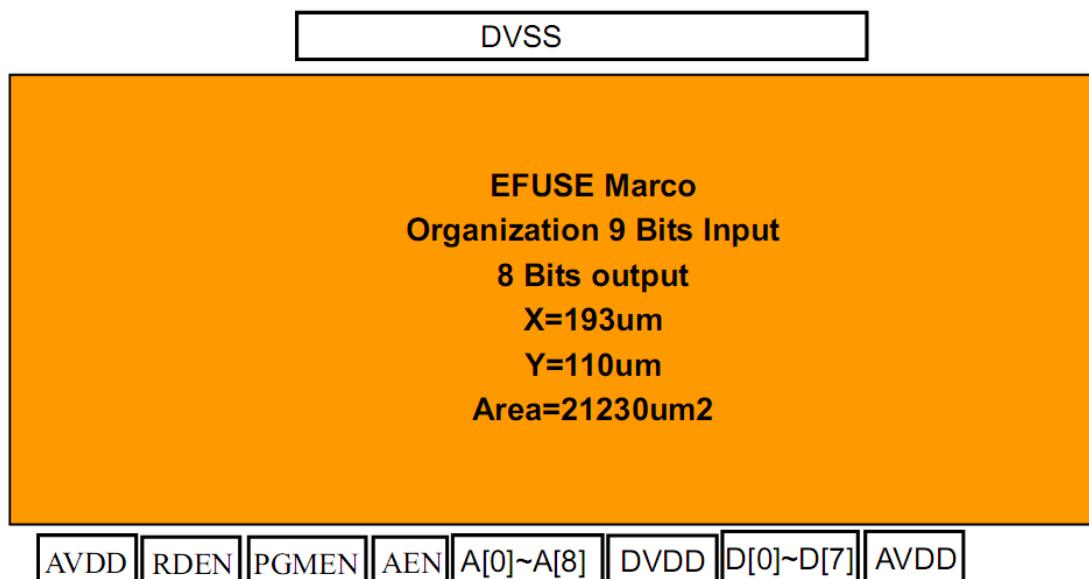


Fig. 39-1RKaudi eFuse block diagram

39.3 Function description

eFuse has three operation modes. They are defined as read, program and inactive. The EFUSE macro enters one of the three modes of operation

determine by the logic level of read select signal (RDEN) and program select (PGMEN). See the below table for the corresponding logic levels and operating modes.

Mode	PGMEN	RDEN	AVDD	DVDD
PGM mode	H	L	H	H
Read mode	L	H	L or Floating	H
Inactive mode 1)	L	L	L or Floating	H

Exceptions and Constraints

- PGMEN and RDEN are not allowed to be H at the same time
- AEN high is not allowed except in PGM mode or READ mode
- A port (Address) toggle is not allowed when AEN is high in PGM mode or READ mode
- Neither READ mode transitioning to PGM mode directly nor PGM mode transitioning to READ mode directly is allowed
- For PGM mode converting to inactive mode, AVDD must change from high to low or floating.

Program (PGM) Mode

Before burning, initial Fuse output is “0”, and written to “1” after burning. When program mode (RDEN=L, PGMEN=H), EFUSE bit specified by address A[8:0] will be burnt by high pulse of AEN. D[7:0] are undefined in PROGRAM mode.

Read Mode

The EFUSE enters read mode if RDEN=H, PGMEN=L. Address signals A[8]~A[6] are invalid

A[5:0]	D[0]	D[1]	D[2]			D[6]	D[7]
000000	Fuse[0]	Fuse[64]	Fuse[64]			Fuse[384]	Fuse[448]
000001	Fuse[1]	Fuse[65]	Fuse[65]			Fuse[385]	Fuse[449]
000010	Fuse[2]	Fuse[66]	Fuse[66]			Fuse[386]	Fuse[450]
.
.
.
111111	Fuse[63]	Fuse[127]	Fuse[191]	Fuse[447]	Fuse[511]

Fig. 39-2 RKaudi eFuse Q[0]~Q[7] corresponds with 256 fuse cells

Inactive Mode

The EFUSE enters inactive mode if neither program mode nor read mode is active. The preferred standby conditions in inactive mode are AEN=L, RDEN=L, PGMEN=L. D[7:0] are undefined in inactive mode.

39.4 Register Description

This section describes the control/status registers of the design.

39.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
EFUSE_con0	0x0000	W	0x00000000	efuse 0 control register
EFUSE_data0	0x0004	W	0x00000000	efuse 0 data out register
EFUSE_con1	0x0008	W	0x00000000	efuse 1 control register
EFUSE_data1	0x000c	W	0x00000000	efuse 1 data out register

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

39.4.2 Detail Register Description

EFUSE_con0

Address: Operational Base + offset (0x0000)

Efuse 0 control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	address address input.
7:6	RO	0x0	reserved
5	RW	0x0	efuse_pd power down enable. active high.
4	RW	0x0	efuse_ps high to pass high program voltage to internal for program.
3	RW	0x0	efuse_pgenb program enable, low enable.
2	RW	0x0	efuse_load high to turn on sense amplifier and load data into latch. Active-High (1.1v) for read mode; when program mode, must be set "low"(0v).
1	RW	0x0	efuse_strobe high to trun on the array for read or program access.
0	RW	0x0	efuse_csb chip select, low active, to put in low power standby mode.

EFUSE_data0

Address: Operational Base + offset (0x0004)

Efuse 0 data out register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	efuse0_dout efuse 0 data out.

EFUSE_con1

Address: Operational Base + offset (0x0008)

efuse 1 control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	address address input.
7:6	RO	0x0	reserved
5	RW	0x0	efuse_pd power down enable. active high.
4	RW	0x0	efuse_ps high to pass high program voltage to internal for program.
3	RW	0x0	efuse_pgenb program enable, low enable.
2	RW	0x0	efuse_load high to turn on sense amplifier and load data into latch. Active-High (1.1v) for read mode; when program mode, must be set "low"(0v).
1	RW	0x0	efuse_strobe high to trun on the array for read or program access.
0	RW	0x0	efuse_csb chip select, low active, to put in low power standby mode.

EFUSE_data1

Address: Operational Base + offset (0x000c)

efuse 1 data out

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	efuse1_dout efuse 1 data out.

39.5 Timing Diagram

- When efuse is in program(PGM) mode.

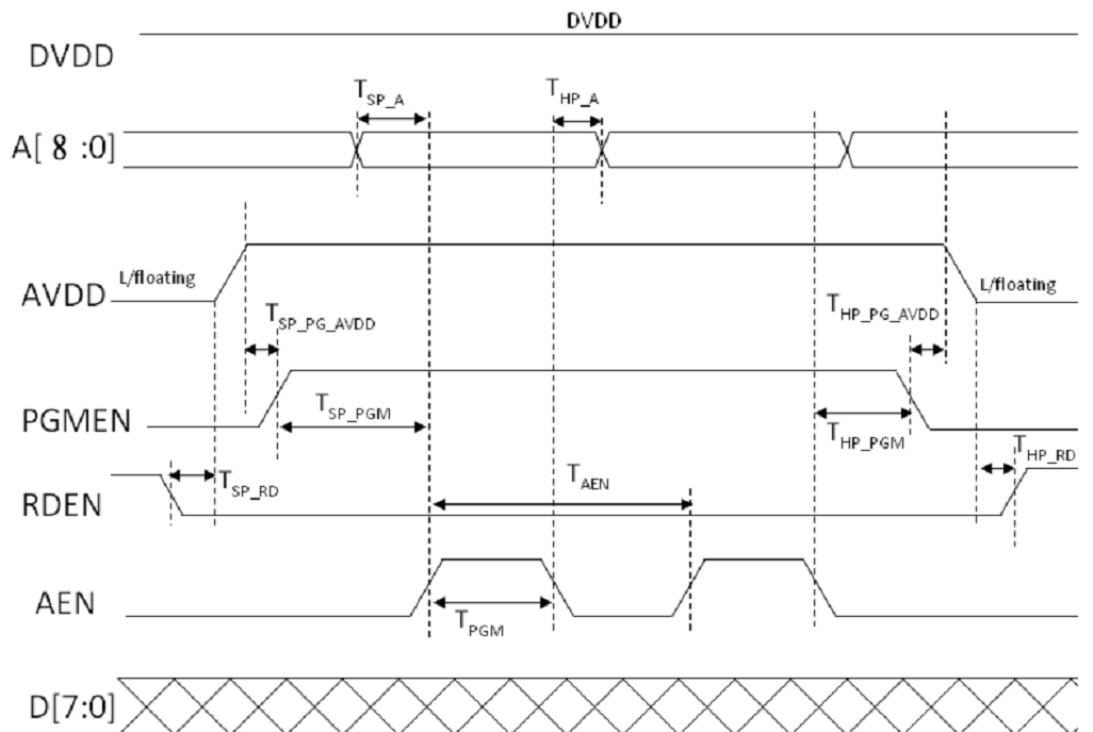


Fig. 39-2RKaudi efuse timing diagram in program mode

Table 39-1RKaudi eFuse program timing parameters list

Parameter	symbol	value			unit
		min	typ	max	
Burning time	T_{PGM}	2000	2000	10000	ns
Address enable cycle time	T_{AEN}	$T_{PGM}+1900$			ns
Address to AEN setup time	T_{SP_A}	50			ns
Address hold time from AEN	T_{HP_A}	50			ns
PGMEN signal to AEN setup time	T_{SP_PGM}	100			ns
AEN to PGMEN signal hold time	T_{HP_PGM}	100			ns
RDEN signal to AVDD setup time	T_{SP_RD}	150			ns
AVDD to RDEN signal hold time	T_{HP_RD}	150			ns
AVDD to PGMEN setup time	$T_{SP_PG_AVDD}$	1000			ns
PGMEN to AVDD hold time	$T_{HP_PG_AVDD}$	1000			ns

Note

4. The accumulative total time of $1.21V < AVDD \leq 2.75V$ must NOT exceed 1s
 5. AVDD is NOT allowed to exceed 2.75V, otherwise there will be reliability issue
 6. Burning time is checked with SMIC on the released date but might be changed because of the process later. Please check with SMIC if you have the concern.
 7. Other values are validated by simulation for the preliminary version
- **When efuse is in read mode.**

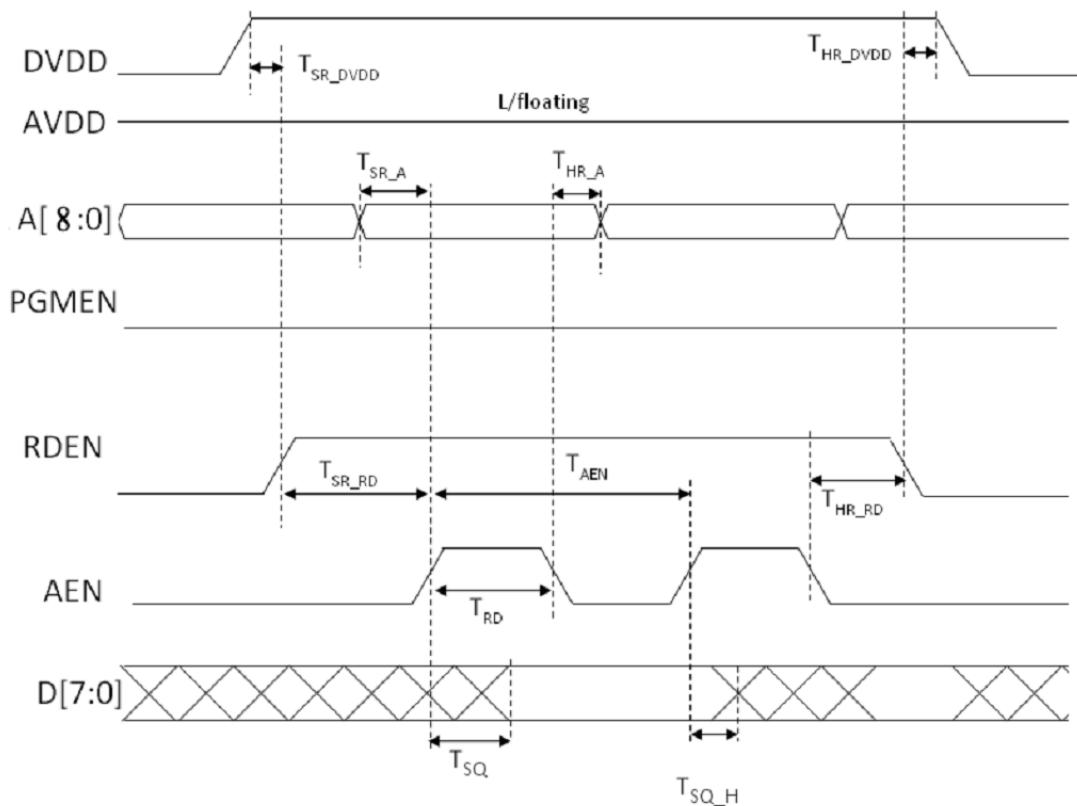


Fig. 39-3RKaudi eFuse timing diagram in read mode

The following table has shows the detailed value for timing parameters in the above diagram.

Table 39-2 RKaudi eFuse read timing parameters list

Parameter	symbol	value			unit
		min	typ	max	
Read time	T_{RD}	40			ns
Address enable cycle time	T_{AEN}	$T_{RD}+35$			ns
Address to AEN setup time	T_{SP_A}	10			ns
AEN to Address hold time	T_{HR_A}	10			ns
DVDD to RDEN setup time	T_{SR_DVDD}	150			ns
RDEN to DVDD hold time	T_{HR_DVDD}	150			ns
RDEN signal to AEN setup time	T_{SR_RD}	100			ns
Output data steady time with 0 loading	T_{SQ}			45	ns
Output data hold time	T_{SQ_H}	0			ns
AEN to RDEN signal hold time	T_{HR_RD}	100			ns

39.6 Application Notes

39.6.1 Operating and Burning Conditions

Table 39-3 Operating and Burning Conditions

Parameter	Symbol	Values			Unit	Note / Condition
		Min.	Typ.	Max.		
Core voltage (DVDD pin)	DVDD	0.99	1.1	1.21	V	-
Burn voltage(AVDD pin)	AVDD	2.25	2.5	2.75	V	
Read operating junction temperature	T _{READ}	-40	25	125	°C	-
Burn operating ambient temperature	T _{BURN}	10	25	40	°C	-

39.6.2 Signal Capacitance Requirements

Table 39-4 Signal Capacitance Requirements

Parameter	Symbol	Values			Unit	Note / Condition
		Min	Typ.	Max.		
RDEN Signal Capacitance	C _{RDE}	-	-	1.4	fF	-
PGMEN Signal Capacitance	C _{PGMEN}	-	-	1.4	fF	-
A[12:0] Signal Capacitance	C _D	-	-	3.2	fF	-
AEN Signal Capacitance	C _{AEN}	-	-	1.6	fF	-

39.6.3 Standby and Active Current

Table 39-5 Standby and Active Current

Parameter	Symbol	Values			Unit
		Min.	Typ.	Max.	
Standby current	I _{STANDBY}	-	1	-	uA
Peak burning current	I _{PROG}	-	15	-	mA
Current during normal reading (10 MHz)	I _{ACTIVE}	-	4	-	mA

Chapter 40 Process-Voltage-Temperature Monitor(PVTM)

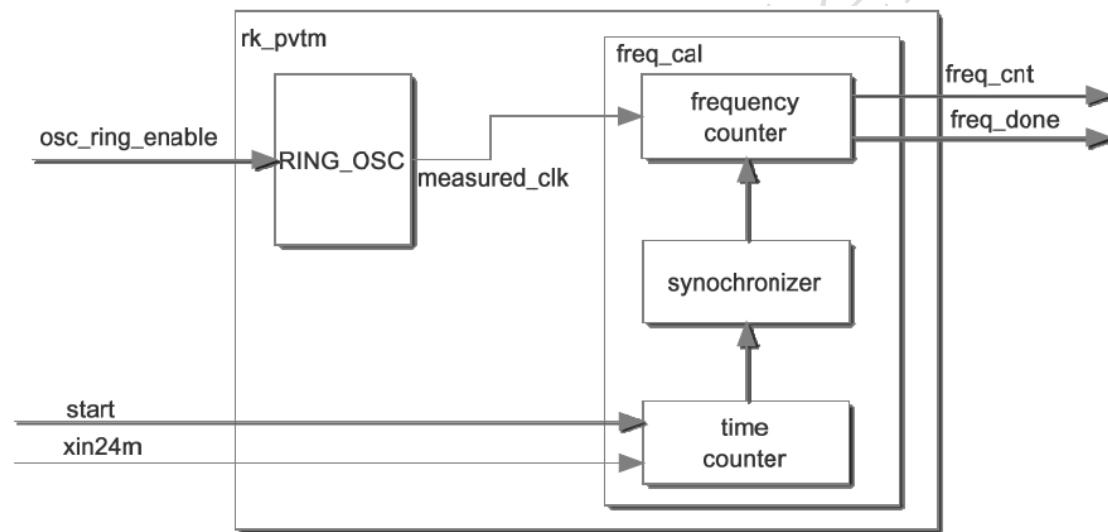
40.1 Overview

The Process-Voltage-Temperature Monitor (PVTM) is used to monitor the chip performance variance caused by chip process, voltage and temperature.

PVTM supports the following features:

- a clock oscillation ring is integrated and used to generate a clock like signal, the frequency of this clock is determined by the cell delay value of clock oscillation ring circuit
- a frequency counter is used to measure the frequency of the clock oscillation ring.

40.2 Block Diagram



The PVTM include two main blocks

- RING_OSC, it is composed with inverters with odd number, which is used to generate a clock
- Freq_cal, it is used to measure the frequency of clock which generated from the RING_OSC block

Frequency Calculation

A frequency fixed clock(24MH) is used to calculate the clock cycles of RING_OSC generated clock. Suppose the time period is 1s, then the clock period of RING_OSC clock is $T = 1/2 * \text{clock_counter}(s)$

Chapter 41 Serial Flash Controller(SFC)

41.1 Overview

The serial flash controller (SFC) is used to control the data transfer between the chip system and the serial nor/nand flash device.

41.1.1 Features

The SFC supports the following features:

- Support AHB slave interface to configure register and read/write serial flash
- Support AHB master interface to transfer data from/to spi flash device
- Support ahb burst with incr4x32bits, or incr x32bits
- Support two independent clock domain: AHB clock and SPI clock
- Support x1,x2,x4 data bits mode
- Support up to 4 chip select
- Support interrupt output, interrupt maskable
- Support Spansion, MXIC,Gigadevice...vendor's nor flash memory.

41.2 Block Diagram

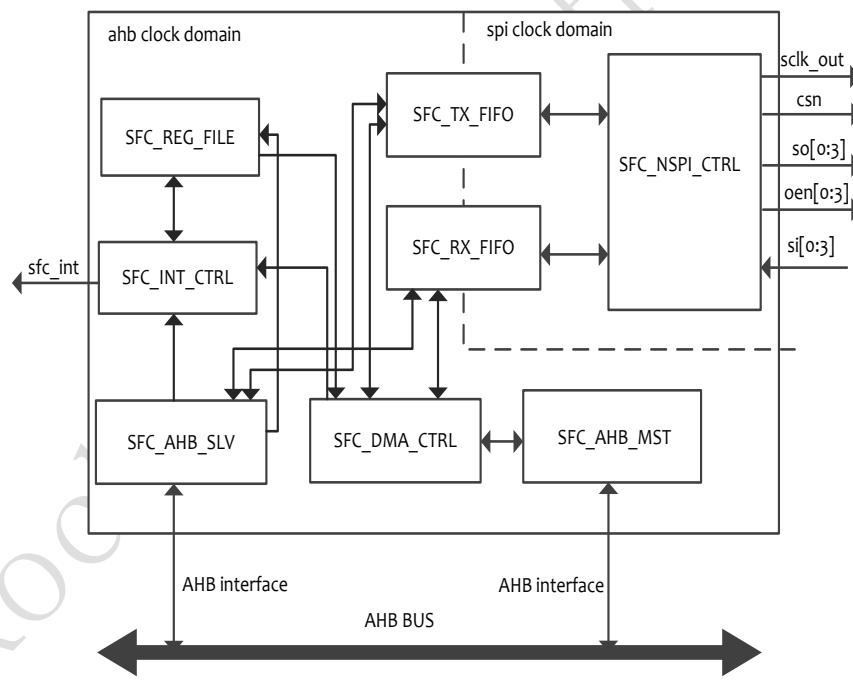


Fig.41-1 SFC architecture

41.3 Function description

41.3.1 SFC ahb slave

The ahb slave is used to configure the register, and also write to/read from the serial nor/nand flash device.

The SFC_CTRL register is a global control register, when the controller is in busy state(SFC_SR), SFC_CTRL cannot be set. The field sclk_idle_level_cycles of

this register is used to configure the idle level cycles of sclk before read the first bit of the read command.

Like the following picture shows: the red line of the sclk is the idle cycles, during these cycles, the chip pad is switched to output. When sclk_idle_level_cycles=0, it means there will be not such idle level.

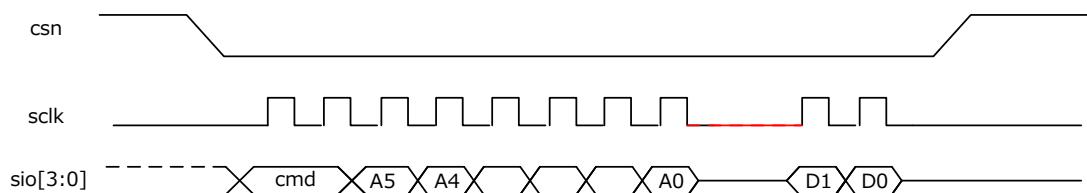


Fig.41-2 idle cycles

When the field spi mode is set, the transfer waveform will like following, and switch to mode3.

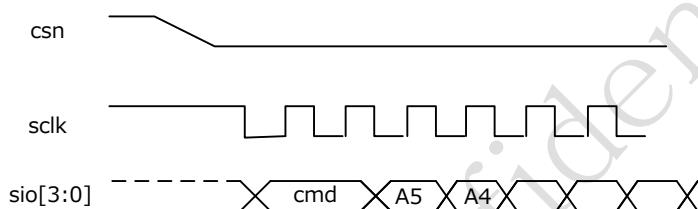


Fig.41-3spi mode

41.4 Register Description

41.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
SFC_CTRL	0x0000	W	0x00000000	SFC systemcontrolregister
SFC_IMR	0x0004	W	0x00000000	SFC status register
SFC_ICLR	0x0008	W	0x00000000	Interrupt mask register
SFC_FTLR	0x000c	W	0x00000000	FIFO threshold level register
SFC_RCVR	0x0010	W	0x00000000	Interrupt clear register
SFC_AX	0x0014	W	0x00000000	Serial flash enhanced mode value
SFC_ABIT	0x0018	W	0x00000000	Serial flash address bits
SFC_ISR	0x001c	W	0x00000000	SFC interrupt status
SFC_FSR	0x0020	W	0x00000000	SFC fifo status
SFC_SR	0x0024	W	0x00000000	SFC status
SFC_RISR	0x0024	W	0x00000000	SFC raw interrupt status
SFC_DMA_TRIGGER	0x0080	W	0x00000000	Ahb master trigger
SFC_DMA_ADDR	0x0084	W	0x00000000	Ahb master source/destination address
SFC_CMD	0x0100	W	0x00000000	Command register
SFC_ADDR	0x0104	W	0x00000000	Address register
SFC_DATA	0x0108	W	0x00000000	Data register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

41.4.2 Detail Register Description

SFC_CTRL

Address: Operational Base + offset (0x000)

bit	Attr	Reset Value	Description
13:12	RW	0x0	Data bits number 2'b00: 1bits 2'b01: 2bits 2'b10: 4bits
11:10	RW	0x0	Addr bits number 2'b00: 1bits 2'b01: 2bits 2'b10: 4bits
9:8	RW	0x0	Cmd bits number 2'b00: 1bits 2'b01: 2bits 2'b10: 4bits
7:4	RW	0x0	sclk_idle_level_cycles 4'b0: idle hold is disabled 4'b1: hold the sclk in idle for two cycles when switch to shift in ...
1	RW	0x0	Shift in phase selection 0: shift in the flash data at posedge sclk_out 1: shift in the flash data at negedge sclk_out
0	RW	0x0	spi mode select

SFC_IMR

Address: Operational Base + offset (0x004)

bit	Attr	Reset Value	Description
7	R/W	1	dma interrupt mask
6	R/W	1	Nspi error interrupt mask
5	R/W	1	Ahb bus error interrupt mask
4	R/W	1	Transfer finish interrupt mask
3	R/W	1	Txfifo empty interrupt mask
2	R/W	1	Txfifo overflow interrupt mask
1	R/W	1	Rx fifo underflow interrupt mask
0	R/W	1	Rx fifo full interrupt mask 1 mask 0 not mask

SFC_ICLR

Address: Operational Base + offset (0x008)

bit	Attr	Reset Value	Description

7	W1C	0	Write to clear dma interrupt
6	W1C	0	Write to clear the error nspi protocol interrupt
5	W1C	0	Write to clear the ahb error response interrupt
4	W1C	0	Write to clear the transfer finish interrupt
3	W1C	0	Write to clear the txfifo empty interrupt
2	W1C	0	Write to clear thetxfifo overflow interrupt
1	W1C	0	Write to clear therxfifo underflow interrupt
0	W1C	0	Write to clear the rxfifo full interrupt

SFC_FTLR

Address: Operational Base + offset (0x008)

bit	Attr	Reset Value	Description
12:8	RW	0x10	Rx fifo threshold level 当RX FIFO中的数据数大等于该水线的时候，会触发 rxfifo full interrupt
4:0	RW	0x10	Txfifo threshold level 当TX FIFO中的数据个数小等于该水线的时候，会触发 txfifo empty interrupt

SFC_RCVR

Address: Operational Base + offset (0x010)

bit	Attr	Reset Value	Description
0	W	0	Reset the FSM, FIFO

SFC_AX

Address: Operational Base + offset (0x014)

bit	Attr	Reset Value	Description
7:0	RW	0xa5	Mobe bit register value when in continuous read mode

SFC_ABIT

Address: Operational Base + offset (0x018)

bit	Attr	Reset Value	Description
4:0	RW	0x00	Serial flash address bits number 0x0: 1bit flash address 0x1: 2bits flash address ... 0x1f:32bits flash address

SFC_ISR

Address: Operational Base + offset (0x01c)

bit	Attr	Reset Value	Description
7	R	0	dma interrupt status 1: active 0: in-active

			Example: When the FSM in idle state , the txfifo send out a data but a command
6	R	0	Error nspi protocol interrupt status 1: active 0: in-active
5	R	0	Ahb error interrupt status 1: active 0: in-active
4	R	0	Transfer finish interrupt status 1: active 0: in-active
3	R	0	Txfifo empty interrupt status 1: active 0: in-active
2	R	0	Txfifo overflow interrupt status 1: active 0: in-active
1	R	0	Rx fifo underflow interrupt status 1: active 0: in-active
0	R	0	Rx fifo full interrupt status 1: active 0: in-active

SFC_FSR

Address: Operational Base + offset (0x020)

bit	Attr	Reset Value	Description
20:16	R	0	rxfifowaterlevel 0x0: fifo is empty 0x1: one entry is taken ... 0x10: 16 entry is taken fifo is full
12:8	R	0	txfifowaterlevel: 0x0: fifo is full 0x1:left 1 entry ... 0x10: left 16 entry,fifo is empty
3	R	0	rxfifo full 0: rxfifo is not full 1: rxfifo is full
2	R	0	rxfifo empty 0: rxfifo is not empty 1: rxfifo is empty
1	R	0	txfifoempty 0: txfifo is not empty 1: txfifo is empty
0	R	0	txfifo full 0: txfifo is not full 1: txfifo is full

SFC_SR

Address: Operational Base + offset (0x024)

bit	Attr	Reset Value	Description
0	R	0	sfc busy flag 0: sfc is idle 1: sfc is busy. When busy , don't try to set the control register

SFC_RISR

Address: Operational Base + offset (0x028)

bit	Attr	Reset Value	Description
7	R	0	dma raw interrupt status 1: active 0: in-active Example: When the FSM in idle state , the txfifo send out a data but a command
6	R	0	Error nsipi protocol raw interrupt status 1: active 0: in-active
5	R	0	Ahb error raw interrupt status 1: active 0: in-active
4	R	0	Transfer finish raw interrupt status 1: active 0: in-active
3	R	0	Txfifo empty raw interrupt status 1: active 0: in-active
2	R	0	Txfifo overflow raw interrupt status 1: active 0: in-active
1	R	0	Rx fifo underflow raw interrupt status 1: active 0: in-active
0	R	0	Rx fifo full raw interrupt status 1: active 0: in-active

SFC_DMA_TRIGGER

Address: Operational Base + offset (0x080)

bit	Attr	Reset Value	Description
0	W1C	0	Dma start trigger signal. Auto cleared after write

SFC_DMA_ADDR

Address: Operational Base + offset (0x084)

bit	Attr	Reset Value	Description
31:0	RW	0	The source address/destination address for dma

			to transfer
--	--	--	-------------

SFC_CMD

Address: Operational Base + offset (0x100)

bit	Attr	Reset Value	Description
31:30	RW	0	Chip select
29:16	RW	0	Transferred bytes number
15:14	RW	0	Address bits number 2'b00 : 0 2'b01: 24bits 2'b10: 32bits 2'b11: from register setting(SFC_ADDR_BITS)
13	RW	0	Continuous read mode
12	RW	0	0:read, 1: write
11:8	RW	0	Dummy bits number
7:0	RW	0	Command that will send to Serial Flash At the same time with writting the SFC_CMD,TX FIFO will also be written with SFC_CMD.

SFC_ADDR

Address: Operational Base + offset(0x104)

bit	Attr	Reset Value	Description
31:0	RW	0	Address that will send to serial flash.

SFC_DATA

Address: Operational Base + offset(0x108)

bit	Attr	Reset Value	Description
31:0	RW	0	Currently data written to serial flash

41.5 Interface Description

Table 41-1 IOMUX Settings for SFC

Module Pin	Direction	Pad Name	IOMUX Setting
sfc_clk	O	IO_NANDrdy_EMMCCmd1_S FCclk_GPIO2a4	GRF_GPIO2A_IOMUX[9:8]=2'b 11
sfc_csn0	O	IO_NANDwrn_SFCCsn0_GPI O2a2	GRF_GPIO2A_IOMUX[5:4]=2'b 10
sfc_csn1	O	IO_NANDrdn_SFCCsn1_GPI O2a3	GRF_GPIO2A_IOMUX[7:6]=2'b 10
sfc_sio0	I/O	IO_NANDd0_EMMCd0_SFCd 0_GPIO1d0	GRF_GPIO1D_IOMUX[1:0]=2'b 11
sfc_sio1	I/O	IO_NANDd1_EMMCd1_SFCd 1_GPIO1d1	GRF_GPIO1D_IOMUX[3:2]=2'b 11
sfc_sio2	I/O	IO_NANDd2_EMMCd2_SFCd 2_GPIO1d2	GRF_GPIO1D_IOMUX[5:4]=2'b 11
sfc_sio3	I/O	IO_NANDd3_EMMCd3_SFCd 3_GPIO1d3	GRF_GPIO1D_IOMUX[7:6]=2'b 11

Notes: Direction: **I**- Input, **O**- Output, **I/O**- Input/Output

41.6 Application Notes

41.6.1 AHB Slave write flash flow

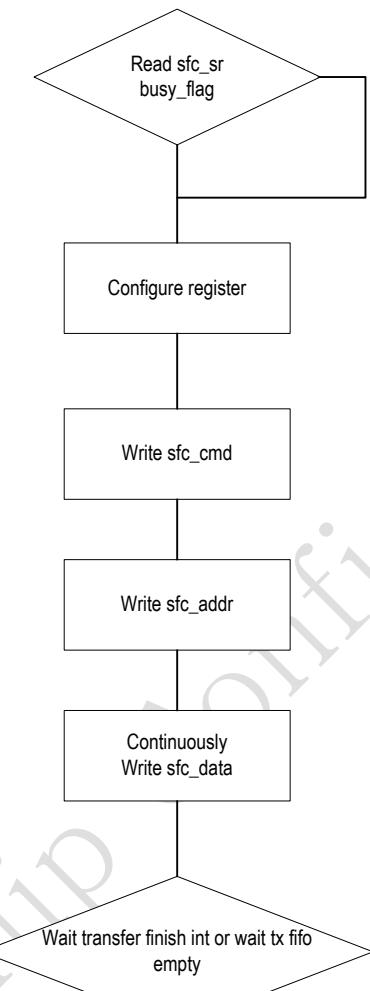


Fig.41-4 slave mode write

41.6.2 AHB Slave read flash flow

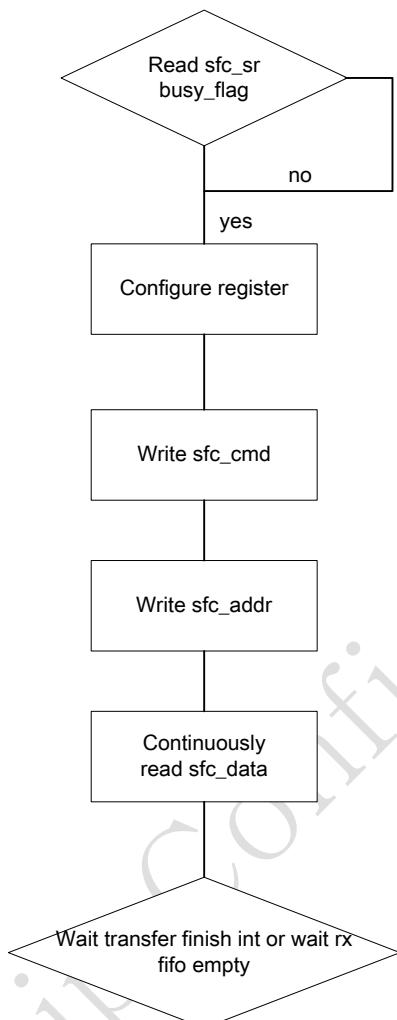


Fig.41-5 slave mode read

41.6.3 AHB dma transfer flow

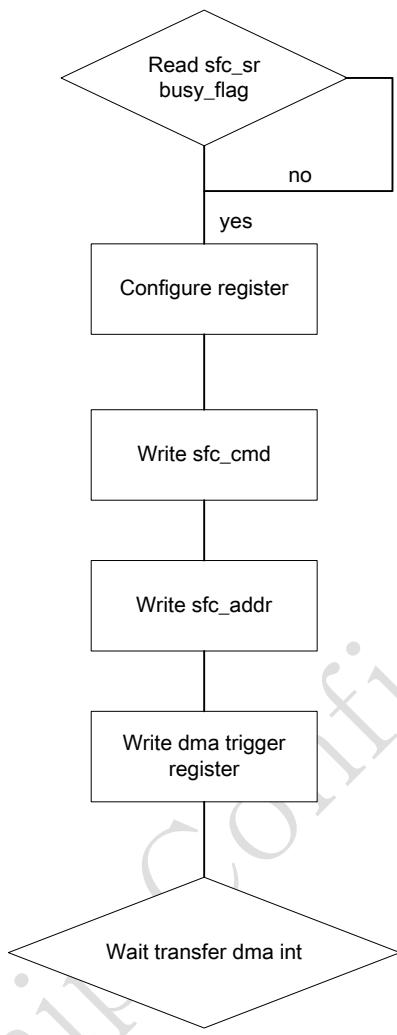


Fig.41-6 master mode flow

41.6.4 Software notice

The sfcsclk need to be kept under 200MHZ. It's better to soft reset the sfc before data transfer.

Chapter 42 TSP(Transport Stream Processing Module)

42.1 Overview

The Transport Stream Processing Module(TSP) is designed for processing Transport Stream Packets, including receiving TS packets, PID filtering, TS descrambling, De-multiplexing and TS outputting. Processed data are transferred to memory buffer which are continued to be processing by software.

TPS supports the following features:

- Supports two TS input channels and one TS output channel
- Supports 4 TS Input Mode: sync/valid mode in the case of serial TS input; nosync/valid mode, sync/valid, sync/burst mode in the case of parallel TS input
- Supports serial and parallel output mode with PCR adjustment, and lsb-msb or msb-lsb bit ordering can be chosen in the serial output mode
- Supports 2 TS sources: demodulators and local memory
- Supports 2 Built-in PTIs(Programmable Transport Interface) to process TS simultaneously
- Supports 1 PVR(Personal Video Recording) output channel
- 1 built-in multi-channel DMA Controller
- DMAC supports:
 - Word alignment transfer
 - Fixed and incrementing addressing
 - Word size transfer
 - burst modes: Incr4, Incr8, Inc16; burst transfer will be done with INCR mode if the remaining data or address space is not capable to perform a complete burst transfer
 - Hardware/software trigger mode
 - LLP(List Link Programming) Mode
 - DMA done and error interrupt for each PTI channel
- Each PTI supports
 - 64 PID filters
 - TS descrambling with 16 sets of Control Word under CSA v2.0 standard, up to 104Mbps
 - 16 PES/ES filters with PTS/DTS extraction and ES start code detection
 - 4/8 PCR extraction channels
 - 64 Section filters with CRC check, and three interrupt mode: stop per unit, full-stop, recycle mode with version number check
 - PID done and error interrupts for each channel
 - PCR/DTS/PTS extraction interrupt for each channel

42.2 Block Diagram

The TSP comprises of following components:

- AMBA AHB slave interface
- Register block
- PTI
- DMAC
- TS Out Interface

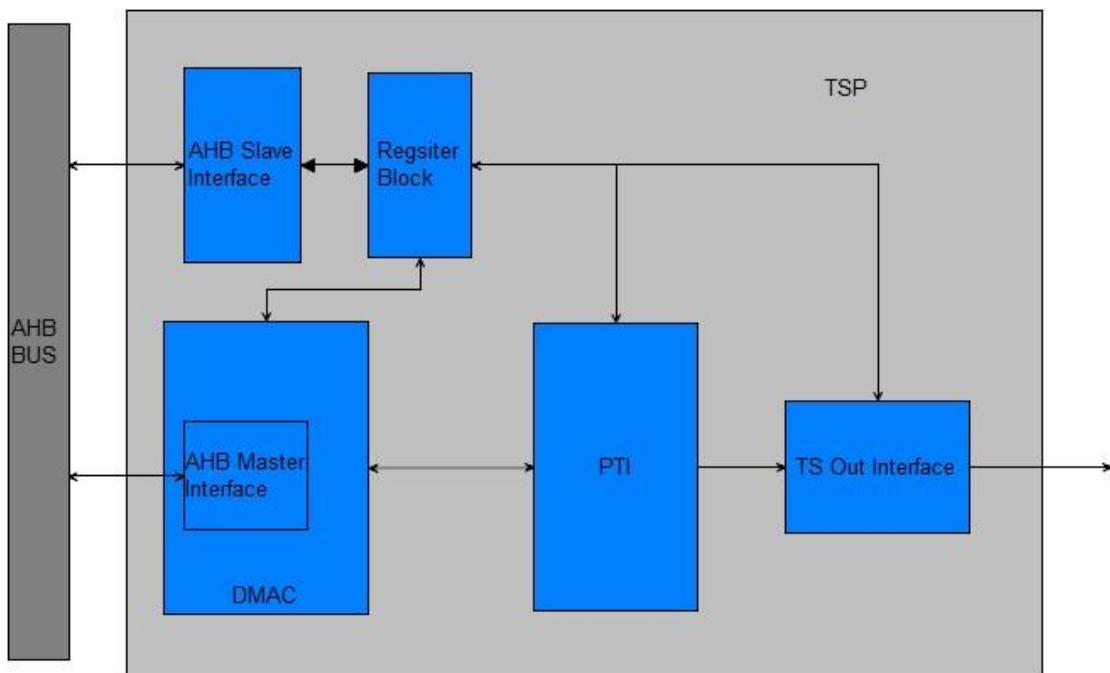


Fig. 42-1 TSP architecture

AHB Slave INTERFACE

The host processor can get access to the register block through AHB slave interface. The slave interface supports 32bit access.

Register block

All registers in the TSP are addressed at 32-bit boundaries to remain consistent with the AHB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

PTI

Most of the TS processing are dealt with PTI. TS packets are re-synchronized, filtered, descrambled and demultiplexing, and the processed packets are transferred to memory buffer to be processed further by software. The embedded TS in interface can receive TS packets by connecting to a compliant TS demodulator. TS stream stored in the local memory is another source to feed into PTI through by using LLP DMA mode.

TS Out Interface

TS out interface can output either PID-filtered or non-PID-filtered TS packets from one PTI channel in a certain stream mode as configured. The TS receiver conforms to the stream mode to receive the TS packets.

DMAC

The DMAC performs all DMA transfers which get access to memory.

42.3 Function Description

42.3.1 TS Stream of TS_IN Interface

TS_IN interface supports 4 input TS stream mode: sync/valid serial mode,

sync/valid parallel mode, sync/burst parallel mode, nosync/valid parallel mode.

A. Sync/Valid Serial Mode

In this mode, TS_IN interface takes use of TSI_SYNC and TSI_VALID clocked with TSI_CLK signal to sample input serial TS packet data.

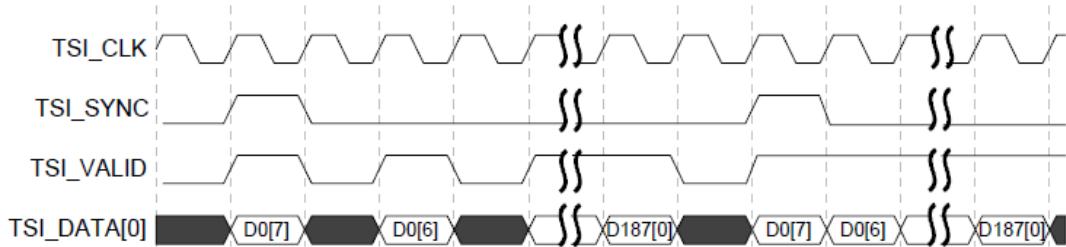


Fig. 42-2 Sync/Valid Serial Mode with Msb-Lsb Bit Ordering

TSI_SYNC must be active high together with TSI_VALID when indicating the first valid bit of a TS packet, and TSI_VALID indicates the 188*8 valid bits of a TS packet. TSI supports both msb-lsb and lsb-msb bit ordering.

B. Sync/Valid Parallel Mode

In this mode, TS_IN interface takes use of TSI_SYNC and TSI_VALID clocked with TSI_CLK signal to sample input parallel TS packet data.

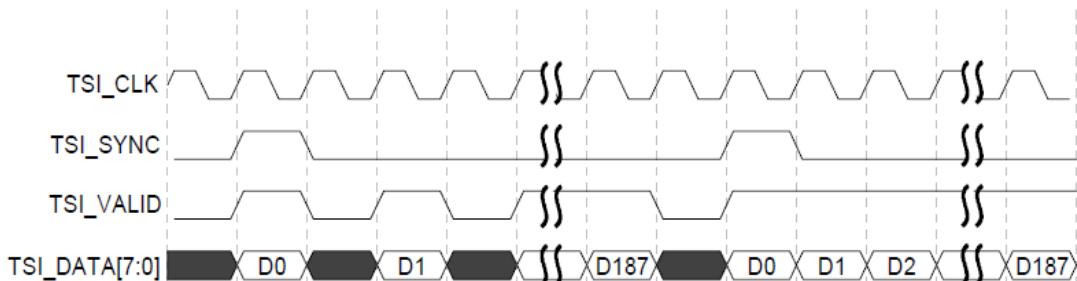


Fig. 42-3 Sync/valid Parallel Mode

TSI_SYNC must be active high together with TSI_VALID when indicating the first valid byte of a TS packet, and TSI_VALID indicates the 188 valid byte of a TS packet.

C. Sync/Burst Parallel Mode

In this mode, TSI only takes use of TSI_SYNC to sample input parallel TS packet data.

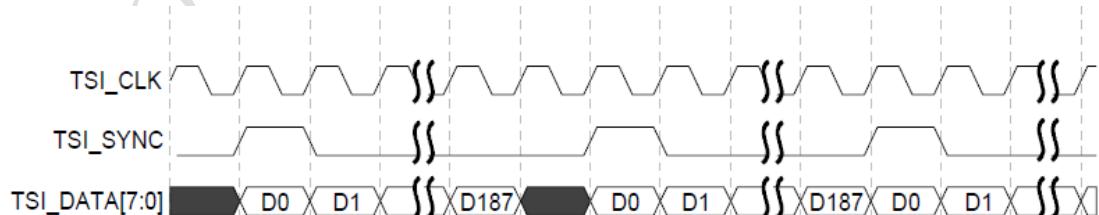


Fig. 42-4 Sync/Burst Parallel Mode

When active high, TSI_SYNC implies the first valid byte of a TS packet and remaining 187 valid bytes of a TS packet are upcoming within the following successive 187 clock cycles.

D. Nosync/Valid Parallel Mode

In this mode, TSI only takes uses of TSI_VALID to sample input parallel TS packet data.

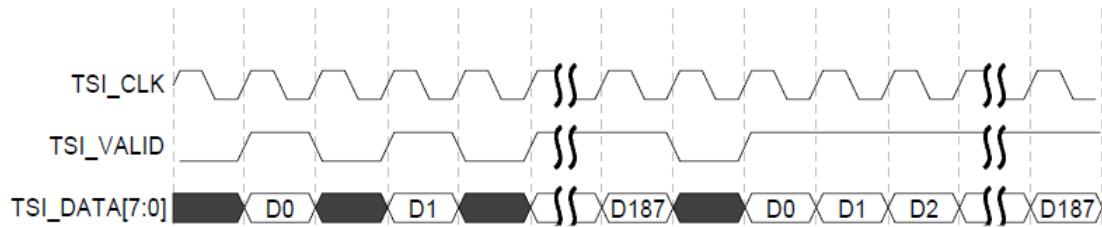


Fig. 42-5 Nosync/Valid Parallel Mode

When active high, TSI_VALID implies a valid byte of a TS packet.

42.3.2 TS output of TS Out Interface

TS out interface transmit the TS data in two mode: serial mode and parallel mode. In the serial mode, the bit order can be lsb-msb or msb-lsb.

The TS_SYNC will be active high when indicating the header of the TS packets, and it only lasts for one cycle. TS_VALID will be active high when the output TS data is valid. The output data is 188 byte TS packet data.

TS out interface also stamp the TS output stream with new PCR value, making PCR adjustment. PCR is used to measure the transport rate.

$$PCR(i) = PCR_base(i) \times 300 + PCR_ext(i)$$

where:

$$PCR_base(i) = ((system_clock_frequency \times t(i)) DIV 300) \% 2^{33}$$

$$PCR_ext(i) = ((system_clock_frequency \times t(i)) DIV 1) \% 300$$

$$transport_rate(i) = \frac{((i' - i'') \times system_clock_frequency)}{PCR(i') - PCR(i'')}$$

Where

i' is the index of the byte containing the last bit of the immediately following program_clock_reference_base field applicable to the program being decoded.

i is the index of any byte in the Transport Stream for $i'' < i < i'$.

i'' is the index of the byte containing the last bit of the most recent program_clock_reference_base field applicable to the program being decoded.

System clock is 27Mhz.

42.3.3 Demux and descrambling

Each PTI has 64 PID channels to deal with demultiplexing and descrambling operation.

The PTI can descramble the TS Packets which are scrambled with CSA v2.0 standard. The TS packets can be scrambled either in TS level or PES level.

The demux module can do the section filtering, pes filtering and es filtering, or directly output TS packets.

42.4 Register Description

42.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
TSP_GCFG	0x0000	W	0x00000000	Global Configuration Register
TSP_PVR_CTRL	0x0004	W	0x00000000	PVR Control Register
TSP_PVR_LEN	0x0008	W	0x00000000	PVR DMA Transaction Length
TSP_PVR_ADDR	0x000c	W	0x00000000	PVR DMA transaction starting address
TSP_PVR_INT_STS	0x0010	W	0x00000000	PVR DMA Interrupt Status Register
TSP_PVR_INT_ENA	0x0014	W	0x00000000	DMA Interrupt Enable Register
TSP_TSOUT_CTRL	0x0018	W	0x00000000	TS Out Control Register
TSP_PTIX_CTRL	0x0100	W	0x00000000	PTI Channel Control Register
TSP_PTIX_LLPCFG	0x0104	W	0x00000000	LLP DMA Control Register
TSP_PTIX_LLPCBASE	0x0108	W	0x00000000	LLP Descriptor BASE Address
TSP_PTIX_LLPPWRIT	0x010c	W	0x00000000	LLP DMA Writing Software Descriptor Counter
TSP_PTIX_LLPREAD	0x0110	W	0x00000000	LLP DMA Reading Hardware Descriptor Counter
TSP_PTIX_PID_STS0	0x0114	W	0x00000000	PTI PID Channel Status 0 Register
TSP_PTIX_PID_STS1	0x0118	W	0x00000000	PTI PID Channel Status 1 Register
TSP_PTIX_PID_STS2	0x011c	W	0x00000000	PTI PID Channel Status 2 Register
TSP_PTIX_PID_STS3	0x0120	W	0x00000000	PTI PID Channel Status 3 Register
TSP_PTIX_PIDINTENA0	0x0124	W	0x00000000	PID Interrupt Enable Register 0
TSP_PTIX_PIDINTENA1	0x0128	W	0x00000000	PID Interrupt Enable Register 1
TSP_PTIX_PIDINTENA2	0x012c	W	0x00000000	PID Interrupt Enable Register 2
TSP_PTIX_PIDINTENA3	0x0130	W	0x00000000	PID Interrupt Enable Register 3
TSP_PTIX_PCRINTSTS	0x0134	W	0x00000000	PTI PCR Interrupt Status Register
TSP_PTIX_PCRINTENA	0x0138	W	0x00000000	PTI PCR Interrupt Enable Register
TSP_PTIX_PCRNCTR	0x013c	W	0x00000000	PID PCR Control Register
TSP_PTIX_PCRNH	0x015c	W	0x00000000	High Order PCR value
TSP_PTIX_PCRNL	0x0160	W	0x00000000	Low Order PCR value
TSP_PTIX_DMASTS	0x019c	W	0x00000000	LLP DMA Interrupt Status Register
TSP_PTIX_DMAENA	0x01a0	W	0x00000000	DMA Interrupt Enable Register

Name	Offset	Size	Reset Value	Description
TSP_PTIx_DATA_FLA_G0	0x01a4	W	0x00000000	PTI_PID_WRITE Flag 0
TSP_PTIx_DATA_FLA_G1	0x01a8	W	0x00000000	PTI_PID_WRITE Flag 1
TSP_PTIx_LIST_FLA_G	0x01ac	W	0x00000000	PTIx_LIST_WRITE Flag
TSP_PTIx_DST_STS0	0x01b0	W	0x00000000	PTI Destination Status Register
TSP_PTIx_DST_STS1	0x01b4	W	0x00000000	PTI Destination Status Register
TSP_PTIx_DST_ENA_0	0x01b8	W	0x00000000	PTI Destination Interrupt Enable Register
TSP_PTIx_DST_ENA_1	0x01bc	W	0x00000000	PTI Destination Interrupt Enable Register
TSP_PTIx_ECWn_H	0x0200	W	0x00000000	The Even Control Word High Order
TSP_PTIx_ECWn_L	0x0204	W	0x00000000	The Even Control Word Low Order
TSP_PTIx_OCWn_H	0x0208	W	0x00000000	The Odd Control Word High Order
TSP_PTIx_OCWn_L	0x020c	W	0x00000000	The Odd Control Word Low Order
TSP_PTIx_PIDn_CTR_L	0x0300	W	0x00000000	PID Channel Control Register
TSP_PTIx_PIDn_BAS_E	0x0400	W	0x00000000	PTI Data Memory Buffer Base Address
TSP_PTIx_PIDn_TOP	0x0404	W	0x00000000	PTI Data Memory Buffer Top Address
TSP_PTIx_PIDn_WRI_TE	0x0408	W	0x00000000	PTI Data Memory Buffer Hardware Writing Address
TSP_PTIx_PIDnREA_D	0x040c	W	0x00000000	PTI Data Memory Buffer Software Reading Address
TSP_PTIx_LISTn_BA_SE	0x0800	W	0x00000000	PTI List Memory Buffer Base Address
TSP_PTIx_LISTn_TO_P	0x0804	W	0x00000000	PTI List Memory Buffer Top Address
TSP_PTIx_LISTn_WRI_TE	0x0808	W	0x00000000	PTI List Memory Buffer Hardware Writing Address
TSP_PTIx_LISTnREA_D	0x080c	W	0x00000000	PTI List Memory Buffer Software Reading Address
TSP_PTIx_PIDn_CFG	0x0900	W	0x00000008	PID Demux Configure Register
TSP_PTIx_PIDn_FILT_0	0x0904	W	0x00000000	Fliter Word 0
TSP_PTIx_PIDn_FILT_1	0x0908	W	0x00000000	Fliter Word 1
TSP_PTIx_PIDn_FILT_2	0x090c	W	0x00000000	Fliter Word 2
TSP_PTIx_PIDn_FILT_3	0x0910	W	0x00000000	Fliter Word 3

Notes: *Size* : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** - WORD (32 bits) access

42.4.2 Detail Register Description

TSP_GCFG

Address: Operational Base + offset (0x0000)

Global Configuration Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x0	arbit_cnt DMA channel arbiter counter This field is used to adjust the priority of DMA channels to prevent one channel holds the highest priority for a long time. The 3-bit field sets the largest times for a DMA channel to hold the highest priority to send the bus request. After requested times reach this limit, the highest priority is passed to next DMA channel in order.
3	RW	0x0	tsout_on TS Output Module Switch 1: TS output module switched on 0: TS output module switched off
2	RW	0x0	pvr_on PVR Module Switch 1: PVR function turned on ; 0: PVR function turned off ;
1	RW	0x0	pti1_on PTI0 channel switch 1: PTI1 channel switched on 0: PTI1 channel switched off
0	RW	0x0	pti0_on PTI0 channel switch 1: PTI0 channel switched on 0: PTI1 channel switched off

TSP_PVR_CTRL

Address: Operational Base + offset (0x0004)

PVR Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	fixaddr_en Fix Address Mode Select 1: fixed address mode; 0: incrementing address mode;

Bit	Attr	Reset Value	Description
5:4	RW	0x0	burst_mode PVR burst mode PVR DMA burst mode 2'b00: INCR4 2'b01: INCR8 2'b10: INCR16 2'b11: Reserverd
3:2	RW	0x0	source PVR Source Select TS source for PVR output. 00: non-PID-filtered TS packets in PTI0; 01: PID filtered TS packets in PTI0; 10: non-PID-filtered TS packets in PTI1; 11: PID-filtered TS packets in PTI1;
1	RWSC	0x0	stop PVR stop Write 1 to stop DMA channel. DMA will complete current burst transfer and then stop. It may takes several cycles. 1: PVR Stop ; 0: no effect ;
0	RWSC	0x0	start PVR start Write 1 to start PVR. This bit will be cleared if PVR is stopped or PVR transaction is completed. 1: start PVR 0: no effect.

TSP_PVR_LEN

Address: Operational Base + offset (0x0008)

PVR DMA Transaction Length

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	len Transaction Length Transaction Length

TSP_PVR_ADDR

Address: Operational Base + offset (0x000c)

PVR DMA transaction starting address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr PVR DMA transaction starting address PVR DMA transaction starting address

TSP_PVR_INT_STS

Address: Operational Base + offset (0x0010)

PVR DMA Interrupt Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	W1C	0x0	pvr_error PVR DMA transaction error 1: error response during PVR DMA transaction; 0: no error response during PVR DMA transaction;
0	W1C	0x0	pvr_done PVR DMA transaction done 1: PVR DMA transaction completed; 0: PVR DMA transaction not completed;

TSP_PVR_INT_ENA

Address: Operational Base + offset (0x0014)

DMA Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	pvr_error_ena PVR DMA Transaction Error Interrupt Enable 1: Error Interrupt Enabled 0: Error Interrupt Disabled
0	RW	0x0	pvr_done_ena PVR DMA Transaction Done Interrupt Enable 1: Done Interrupt Enabled 0: Done Interrupt Disabled

TSP_TSOUT_CTRL

Address: Operational Base + offset (0x0018)

TS Out Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	tso_sdo_sel TS serial data output 1: bit[0] use as serial data output ; 0: bit[7] use as serial data output ;
5	RW	0x0	tso_clk_phase TS output clock phase 0: ts output clock; 1: inverse of ts output clock.
4	RW	0x0	mode TS Output mode Selection Output mode select: 0: Serial Mode 1: Parallel Mode
3	RW	0x0	bit_order ts output serial data byte order Indicates that the output serial data byte order, ignored in the parallel: 0: MSB to LSB 1: LSB to MSB

Bit	Attr	Reset Value	Description
2:1	RW	0x0	source TS Output Source Select TS source for TS out. 00: non-PID-filtered TS packets in PTI0; 01: PID filtered TS packets in PTI0; 10: non-PID-filtered TS packets in PTI1; 11: PID-filtered TS packets in PTI1;
0	RW	0x0	start TS out start 1: to start TS out function ; 0: to stop TS out function;

TSP_PTIx_CTRL

Address: Operational Base + offset (0x0100)

PTI Channel Control Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	tsi_sdi_sel TS Serial Data Input Select 1: bit[0] use as serial input data 0: bit[7] use as serial input data
20:19	RW	0x0	tsi_error_handle TS ERROR Handle 00: don't output 01: set the error indicator to 1 10: don't care
18	RW	0x0	clk_phase_sel ts input clock phase select 1'b0: ts input clock 1'b1: inverse of ts input clock
17:16	RW	0x0	demux_burst_mode Demux DMA Burst Mode Demux DMA Mode 2'b00: INCR4 2'b01: INCR8 2'b10: INCR16 2'b11: Reserved
15	RW	0x0	sync_bypass Bypass mode Selection 1'b1: Bypass mode, indicating that input TS packets will not be resynchronized and directly fed into the following modules; 1'b0: Synchronous mode, default, indicating that input TS packets will be resynchronized;
14	RW	0x0	cw_byteorder Control Word format Configuration 0: Default: first byte of the word is the highest byte 1: first byte of the word is the lowest byte

Bit	Attr	Reset Value	Description
13	RW	0x0	cm_on CSA Conformance Mechanism Configuration CSA Conformance Mechanism 0: CM turned off 1: CM turned on
12:11	RW	0x0	tsi_mode TSI Input Mode Selection Input mode selection: 00: Serial Sync/valid Mode 01: Parallel Sync/valid Mode 10: Parallel Sync/burst Mode 11: Parallel Nosync/valid Mode
10	RW	0x0	tsi_bit_order input serial data order Indicates that the input serial data byte order, ignored in the parallel mode: 0: MSB to LSB 1: LSB to MSB
9	RW	0x0	tsi_sel TS Input Source Select Select input TS source 1'b1: HSADC ; 1'b0: internal memory ;
8	RW	0x0	out_byteswap Output byteswap function When enabled, the word to be transferred to memory buffer "B4B3B2B1" is performed byteswapping to "B1B2B3B4".
7	RW	0x0	in_byteswap Input TS Word Byteswap When enabled, the input TS word "B4B3B2B1" is perfomed byteswapping to "B1B2B3B4".
6:4	RW	0x0	unsync_times TS Header Unsynchronized Times If synchronous mode is selected. This field sets the successive times of TS packet header error to re-lock TS header when TS is in locked status;
3:1	RW	0x0	sync_times TS Header Synchronized Times If synchronous mode is selected. This field sets the successive times of finding TS packet header to lock the TS header when TS is in unlocked status;
0	RWSC	0x0	clear Software clear signal It will reset the core register . It will take several cycles. After reset done, soft_reset will be low. 1. reset; 0. no effect.

TSP_PTIX_LLPCFG

Address: Operational Base + offset (0x0104)

LLP DMA Control Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:8	RW	0x0	<p>threshold LLP Transfer Threshold The depth for LLP descriptors is 64. An interrupt will be asserted when transfer reaches the threshold set if DMA transfer interrupt is enabled.</p> <p>00: 1/1 depth 01: 1/2 depth 10: 1/4 depth 11: 1/8 depth</p>
7:6	RW	0x0	<p>burst_mode LLP DMA Burst Mode LLP DMA Burst Mode 2'b00: INCR4 2'b01: INCR8 2'b10: INCR16 2'b11: Reserved</p>
5	RW	0x0	<p>hw_trigger Hardware Trigger Select 1. hardware trigger; 0. software trigger;</p>
4	RW	0x0	<p>fix_addr_en Fix Address Mode Select 1: fixed address mode; 0: incrementing address mode;</p>
3	W1C	0x0	<p>cfg_done LLP DMA Configuration Done When all descriptors of LLP are configured, write 1 to this bit. The core will clear this bit when llp transaction is finished ;</p>
2	RW	0x0	<p>pause LLP DMA Pause Write 1 to Pause DMA channel . DMA will complete current burst transfer and then pause. All register stay unchanged. If software write 0 later , It will continue to work. It may take several cycles to pause. 1: pause; 0: continue to work ;</p>
1	W1C	0x0	<p>stop LLP DMA Stop Write 1 to stop DMA channel. DMA will complete current burst transfer and then stop. It may take several cycles. 1: stop ; 0: no effect ;</p>

Bit	Attr	Reset Value	Description
0	W1C	0x0	start LLP DMA start Write 1 to start DMA Channel , self clear after 1 cycle. 1: start ; 0: no effect

TSP_PTIX_LL_P_BASE

Address: Operational Base + offset (0x0108)

LLP Descriptor BASE Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr LLP Descriptor BASE Address LLP Descriptor BASE address

TSP_PTIX_LL_P_WRITE

Address: Operational Base + offset (0x010c)

LLP DMA Writing Software Descriptor Counter

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	counter LLP DMA Writing Software Descriptor Counter LLP DMA Writing Software Descriptor Counter

TSP_PTIX_LL_P_READ

Address: Operational Base + offset (0x0110)

LLP DMA Reading Hardware Descriptor Counter

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	counter LLP DMA Reading Hardware Descriptor Counter Counter LLP DMA Reading Hardware Descriptor Counter

TSP_PTIX_PID_STS0

Address: Operational Base + offset (0x0114)

PTI PID Channel Status 0 Register

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_done PID31 Channel Status 1 means done
30	W1C	0x0	pid30_done PID30 Channel Status 1 means done
29	W1C	0x0	pid29_done PID29 Channel Status 1 means done

Bit	Attr	Reset Value	Description
28	W1C	0x0	pid28_done PID28 Channel Status 1 means done
27	W1C	0x0	pid27_done PID27 Channel Status 1 means done
26	W1C	0x0	pid26_done PID26 Channel Status 1 means done
25	W1C	0x0	pid25_done PID25 Channel Status 1 means done
24	W1C	0x0	pid24_done PID24 Channel Status 1 means done
23	W1C	0x0	pid23_done PID23 Channel Status 1 means done
22	W1C	0x0	pid22_done PID22 Channel Status 1 means done
21	W1C	0x0	pid21_done PID21 Channel Status 1 means done
20	W1C	0x0	pid20_done PID20 Channel Status 1 means done
19	W1C	0x0	pid19_done PID19 Channel Status 1 means done
18	W1C	0x0	pid18_done PID18 Channel Status 1 means done
17	W1C	0x0	pid17_done PID17 Channel Status 1 means done
16	W1C	0x0	pid16_done PID16 Channel Status 1 means done
15	W1C	0x0	pid15_done PID15 Channel Status 1 means done
14	W1C	0x0	pid14_done PID14 Channel Status 1 means done
13	W1C	0x0	pid13_done PID13 Channel Status 1 means done
12	W1C	0x0	pid12_done PID12 Channel Status 1 means done

Bit	Attr	Reset Value	Description
11	W1C	0x0	pid11_done PID11 Channel Status 1 means done
10	W1C	0x0	pid10_done PID10 Channel Status 1 means done
9	W1C	0x0	pid9_done PID9 Channel Status 1 means done
8	W1C	0x0	pid8_done PID8 Channel Status 1 means done
7	W1C	0x0	pid7_done PID7 Channel Status 1 means done
6	W1C	0x0	pid6_done PID6 Channel Status 1 means done
5	W1C	0x0	pid5_done PID5 Channel Status 1 means done
4	W1C	0x0	pid4_done PID4 Channel Status 1 means done
3	W1C	0x0	pid3_done PID3 Channel Status 1 means done
2	RW	0x0	pid2_done PID2 Channel Status 1 means done
1	W1C	0x0	pid1_done PID1 Channel Status 1 means done
0	W1C	0x0	pid0_done PID0 Channel Status 1 means done

TSP_PTIx_PID_STS1

Address: Operational Base + offset (0x0118)

PTI PID Channel Status 1 Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	pid63_done PID63 Channel Status 1 means done
30	W1C	0x0	pid62_done PID62 Channel Status 1 means done
29	W1C	0x0	pid61_done PID61 Channel Status 1 means done

Bit	Attr	Reset Value	Description
28	W1C	0x0	pid60_done PID60 Channel Status 1 means done
27	W1C	0x0	pid59_done PID59 Channel Status 1 means done
26	W1C	0x0	pid58_done PID58 Channel Status 1 means done
25	W1C	0x0	pid57_done PID57 Channel Status 1 means done
24	W1C	0x0	pid56_done PID56 Channel Status 1 means done
23	W1C	0x0	pid55_done PID55 Channel Status 1 means done
22	W1C	0x0	pid54_done PID54 Channel Status 1 means done
21	W1C	0x0	pid53_done PID53 Channel Status 1 means done
20	W1C	0x0	pid52_done PID52 Channel Status 1 means done
19	W1C	0x0	pid51_done PID51 Channel Status 1 means done
18	W1C	0x0	pid50_done PID50 Channel Status 1 means done
17	W1C	0x0	pid49_done PID49 Channel Status 1 means done
16	W1C	0x0	pid48_done PID48 Channel Status 1 means done
15	W1C	0x0	pid47_done PID47 Channel Status 1 means done
14	W1C	0x0	pid46_done PID46 Channel Status 1 means done
13	W1C	0x0	pid45_done PID45 Channel Status 1 means done
12	W1C	0x0	pid44_done PID44 Channel Status 1 means done

Bit	Attr	Reset Value	Description
11	W1C	0x0	pid43_done PID43 Channel Status 1 means done
10	W1C	0x0	pid42_done PID42 Channel Status 1 means done
9	W1C	0x0	pid41_done PID41 Channel Status 1 means done
8	W1C	0x0	pid40_done PID40 Channel Status 1 means done
7	W1C	0x0	pid39_done PID39 Channel Status 1 means done
6	W1C	0x0	pid38_done PID38 Channel Status 1 means done
5	W1C	0x0	pid37_done PID37 Channel Status 1 means done
4	W1C	0x0	pid36_done PID36 Channel Status 1 means done
3	RW	0x0	pid35_done PID35 Channel Status 1 means done
2	W1C	0x0	pid34_done PID34 Channel Status 1 means done
1	W1C	0x0	pid33_done PID33 Channel Status 1 means done
0	RW	0x0	pid32_done PID32 Channel Status 1 means done

TSP_PTIx_PID_STS2

Address: Operational Base + offset (0x011c)

PTI PID Channel Status 2 Register

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_error PID31 Error Interrupt Status 1 means error detected
30	W1C	0x0	pid30_error PID30 Error Interrupt Status 1 means error detected
29	W1C	0x0	pid29_error PID29 Error Interrupt Status 1 means error detected

Bit	Attr	Reset Value	Description
28	W1C	0x0	pid28_error PID28 Error Interrupt Status 1 means error detected
27	W1C	0x0	pid27_error PID27 Error Interrupt Status 1 means error detected
26	W1C	0x0	pid26_error PID26 Error Interrupt Status 1 means error detected
25	W1C	0x0	pid25_error PID25 Error Interrupt Status 1 means error detected
24	W1C	0x0	pid24_error PID24 Error Interrupt Status 1 means error detected
23	W1C	0x0	pid23_error PID23 Error Interrupt Status 1 means error detected
22	W1C	0x0	pid22_error PID22 Error Interrupt Status 1 means error detected
21	W1C	0x0	pid21_error PID21 Error Interrupt Status 1 means error detected
20	W1C	0x0	pid20_error PID20 Error Interrupt Status 1 means error detected
19	W1C	0x0	pid19_error PID19 Error Interrupt Status 1 means error detected
18	W1C	0x0	pid18_error PID18 Error Interrupt Status 1 means error detected
17	W1C	0x0	pid17_error PID17 Error Interrupt Status 1 means error detected
16	W1C	0x0	pid16_error PID16 Error Interrupt Status 1 means error detected
15	W1C	0x0	pid15_error PID15 Error Interrupt Status 1 means error detected
14	W1C	0x0	pid14_error PID14 Error Interrupt Status 1 means error detected
13	W1C	0x0	pid13_error PID13 Error Interrupt Status 1 means error detected
12	W1C	0x0	pid12_error PID12 Error Interrupt Status 1 means error detected

Bit	Attr	Reset Value	Description
11	W1C	0x0	pid11_error PID11 Error Interrupt Status 1 means error detected
10	W1C	0x0	pid10_error PID10 Error Interrupt Status 1 means error detected
9	W1C	0x0	pid9_error PID9 Error Interrupt Status 1 means error detected
8	W1C	0x0	pid8_error PID8 Error Interrupt Status 1 means error detected
7	W1C	0x0	pid7_error PID7 Error Interrupt Status 1 means error detected
6	W1C	0x0	pid6_error PID6 Error Interrupt Status 1 means error detected
5	W1C	0x0	pid5_error PID5 Error Interrupt Status 1 means error detected
4	W1C	0x0	pid4_error PID4 Error Interrupt Status 1 means error detected
3	W1C	0x0	pid3_error PID3 Error Interrupt Status 1 means error detected
2	W1C	0x0	pid2_error PID2 Error Interrupt Status 1 means error detected
1	W1C	0x0	pid1_error PID1 Error Interrupt Status 1 means error detected
0	W1C	0x0	pid0_error PID0 Error Interrupt Status 1 means error detected

TSP_PTIX_PID_STS3

Address: Operational Base + offset (0x0120)

PTI PID Channel Status 3 Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	pid63_error PID63 Error Interrupt Status
30	W1C	0x0	pid62_error PID62 Error Interrupt Status
29	W1C	0x0	pid61_error PID61 Error Interrupt Status

Bit	Attr	Reset Value	Description
28	W1C	0x0	pid60_error PID60 Error Interrupt Status
27	W1C	0x0	pid59_error PID59 Error Interrupt Status
26	W1C	0x0	pid58_error PID58 Error Interrupt Status
25	W1C	0x0	pid57_error PID57 Error Interrupt Status
24	W1C	0x0	pid56_error PID56 Error Interrupt Status
23	W1C	0x0	pid55_error PID55 Error Interrupt Status
22	W1C	0x0	pid54_error PID54 Error Interrupt Status
21	W1C	0x0	pid53_error PID53 Error Interrupt Status
20	W1C	0x0	pid52_error PID52 Error Interrupt Status
19	W1C	0x0	pid51_error PID51 Error Interrupt Status
18	W1C	0x0	pid50_error PID50 Error Interrupt Status
17	W1C	0x0	pid49_error PID49 Error Interrupt Status
16	W1C	0x0	pid48_error PID48 Error Interrupt Status
15	W1C	0x0	pid47_error PID47 Error Interrupt Status
14	W1C	0x0	pid46_error PID46 Error Interrupt Status
13	W1C	0x0	pid45_error PID45 Error Interrupt Status
12	W1C	0x0	pid44_error PID44 Error Interrupt Status

Bit	Attr	Reset Value	Description
11	W1C	0x0	pid43_error PID43 Error Interrupt Status
10	W1C	0x0	pid42_error PID42 Error Interrupt Status
9	W1C	0x0	pid41_error PID41 Error Interrupt Status
8	W1C	0x0	pid40_error PID40 Error Interrupt Status
7	W1C	0x0	pid39_error PID39 Error Interrupt Status
6	W1C	0x0	pid38_error PID38 Error Interrupt Status
5	W1C	0x0	pid37_error PID37 Error Interrupt Status
4	W1C	0x0	pid36_error PID36 Error Interrupt Status
3	W1C	0x0	pid35_error PID35 Error Interrupt Status
2	W1C	0x0	pid34_error PID34 Error Interrupt Status
1	W1C	0x0	pid33_error PID33 Error Interrupt Status
0	W1C	0x0	pid32_error PID32 Error Interrupt Status

TSP_PTIX_PID_INT_ENA0

Address: Operational Base + offset (0x0124)

PID Interrupt Enable Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_done_ena PID31 Done Enable 1:enabled 0:disabled
30	RW	0x0	pid30_done_ena PID30 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
29	RW	0x0	pid29_done_ena PID29 Done Enable 1:enabled 0:disabled
28	RW	0x0	pid28_done_ena PID28 Done Enable 1:enabled 0:disabled
27	RW	0x0	pid27_done_ena PID27 Done Enable 1:enabled 0:disabled
26	RW	0x0	pid26_done_ena PID26 Done Enable 1:enabled 0:disabled
25	RW	0x0	pid25_done_ena PID25 Done Enable 1:enabled 0:disabled
24	RW	0x0	pid24_done_ena PID24 Done Enable 1:enabled 0:disabled
23	RW	0x0	pid23_done_ena PID23 Done Enable 1:enabled 0:disabled
22	RW	0x0	pid22_done_ena PID22 Done Enable 1:enabled 0:disabled
21	RW	0x0	pid21_done_ena PID21 Done Enable 1:enabled 0:disabled
20	RW	0x0	pid20_done_ena PID20 Done Enable 1:enabled 0:disabled
19	RW	0x0	pid19_done_ena PID19 Done Enable 1:enabled 0:disabled
18	RW	0x0	pid18_done_ena PID18 Done Enable 1:enabled 0:disabled
17	RW	0x0	pid17_done_ena PID17 Done Enable

Bit	Attr	Reset Value	Description
16	RW	0x0	pid16_done_ena PID16 Done Enable 1:enabled 0:disabled
15	RW	0x0	pid15_done_ena PID15 Done Enable 1:enabled 0:disabled
14	RW	0x0	pid14_done_ena PID14 Done Enable 1:enabled 0:disabled
13	RW	0x0	pid13_done_ena PID13 Done Enable 1:enabled 0:disabled
12	RW	0x0	pid12_done_ena PID12 Done Enable 1:enabled 0:disabled
11	RW	0x0	pid11_done_ena PID11 Done Enable 1:enabled 0:disabled
10	RW	0x0	pid10_done_ena PID10 Done Enable 1:enabled 0:disabled
9	RW	0x0	pid9_done_ena PID9 Done Enable 1:enabled 0:disabled
8	RW	0x0	pid8_done_ena PID8 Done Enable 1:enabled 0:disabled
7	RW	0x0	pid7_done_ena PID7 Done Enable 1:enabled 0:disabled
6	RW	0x0	pid6_done_ena PID6 Done Enable 1:enabled 0:disabled
5	RW	0x0	pid5_done_ena PID5 Done Enable 1:enabled 0:disabled
4	RW	0x0	pid4_done_ena PID4 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
3	RW	0x0	pid3_done_ena PID3 Done Enable 1:enabled 0:disabled
2	RW	0x0	pid2_done_ena PID2 Done Enable 1:enabled 0:disabled
1	RW	0x0	pid1_done_ena PID1 Done Enable 1:enabled 0:disabled
0	RW	0x0	pid0_done_ena PID0 Done Enable 1:enabled 0:disabled

TSP_PTIX_PID_INT_ENA1

Address: Operational Base + offset (0x0128)

PID Interrupt Enable Register 1

Bit	Attr	Reset Value	Description
31	RW	0x0	pid63_done PID63 Done Enable 1:enabled 0:disabled
30	RW	0x0	pid62_done PID62 Done Enable 1:enabled 0:disabled
29	RW	0x0	pid61_done PID61 Done Enable 1:enabled 0:disabled
28	RW	0x0	pid60_done PID60 Done Enable 1:enabled 0:disabled
27	RW	0x0	pid59_done PID59 Done Enable 1:enabled 0:disabled
26	RW	0x0	pid58_done PID58 Done Enable 1:enabled 0:disabled
25	RW	0x0	pid57_done PID57 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
24	RW	0x0	pid56_done PID56 Done Enable 1:enabled 0:disabled
23	RW	0x0	pid55_done PID55 Done Enable 1:enabled 0:disabled
22	RW	0x0	pid54_done PID54 Done Enable 1:enabled 0:disabled
21	RW	0x0	pid53_done PID53 Done Enable 1:enabled 0:disabled
20	RW	0x0	pid52_done PID52 Done Enable 1:enabled 0:disabled
19	RW	0x0	pid51_done PID51 Done Enable 1:enabled 0:disabled
18	RW	0x0	pid50_done PID50 Done Enable 1:enabled 0:disabled
17	RW	0x0	pid49_done PID49 Done Enable 1:enabled 0:disabled
16	RW	0x0	pid48_done PID48 Done Enable 1:enabled 0:disabled
15	RW	0x0	pid47_done PID47 Done Enable 1:enabled 0:disabled
14	RW	0x0	pid46_done PID46 Done Enable 1:enabled 0:disabled
13	RW	0x0	pid45_done PID45 Done Enable 1:enabled 0:disabled
12	RW	0x0	pid44_done PID44 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
11	RW	0x0	pid43_done PID43 Done Enable 1:enabled 0:disabled
10	RW	0x0	pid42_done PID42 Done Enable 1:enabled 0:disabled
9	RW	0x0	pid41_done PID41 Done Enable 1:enabled 0:disabled
8	RW	0x0	pid40_done PID40 Done Enable 1:enabled 0:disabled
7	RW	0x0	pid39_done PID39 Done Enable 1:enabled 0:disabled
6	RW	0x0	pid38_done PID38 Done Enable 1:enabled 0:disabled
5	RW	0x0	pid37_done PID37 Done Enable 1:enabled 0:disabled
4	RW	0x0	pid36_done PID36 Done Enable 1:enabled 0:disabled
3	RW	0x0	pid35_done PID35 Done Enable 1:enabled 0:disabled
2	RW	0x0	pid34_done PID34 Done Enable 1:enabled 0:disabled
1	RW	0x0	pid33_done PID33 Done Enable 1:enabled 0:disabled
0	RW	0x0	pid32_done PID32 Done Enable 1:enabled 0:disabled

TSP_PTIX_PID_INT_ENA2

Address: Operational Base + offset (0x012c)

PID Interrupt Enable Register 2

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_error PID31 Error Interrupt Enable 1:enabled 0:disabled
30	RW	0x0	pid30_error PID30 Error Interrupt Enable 1:enabled 0:disabled
29	RW	0x0	pid29_error PID29 Error Interrupt Enable 1:enabled 0:disabled
28	RW	0x0	pid28_error PID28 Error Interrupt Enable 1:enabled 0:disabled
27	RW	0x0	pid27_error PID27 Error Interrupt Enable 1:enabled 0:disabled
26	RW	0x0	pid26_error PID26 Error Interrupt Enable 1:enabled 0:disabled
25	RW	0x0	pid25_error PID25 Error Interrupt Enable 1:enabled 0:disabled
24	RW	0x0	pid24_error PID24 Error Interrupt Enable 1:enabled 0:disabled
23	RW	0x0	pid23_error PID23 Error Interrupt Enable 1:enabled 0:disabled
22	RW	0x0	pid22_error PID22 Error Interrupt Enable 1:enabled 0:disabled
21	RW	0x0	pid21_error PID21 Error Interrupt Enable 1:enabled 0:disabled
20	RW	0x0	pid20_error PID20 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
19	RW	0x0	pid19_error PID19 Error Interrupt Enable 1:enabled 0:disabled
18	RW	0x0	pid18_error PID18 Error Interrupt Enable 1:enabled 0:disabled
17	RW	0x0	pid17_error PID17 Error Interrupt Enable 1:enabled 0:disabled
16	RW	0x0	pid16_error PID16 Error Interrupt Enable 1:enabled 0:disabled
15	RW	0x0	pid15_error PID15 Error Interrupt Enable 1:enabled 0:disabled
14	RW	0x0	pid14_error PID14 Error Interrupt Enable 1:enabled 0:disabled
13	RW	0x0	pid13_error PID13 Error Interrupt Enable 1:enabled 0:disabled
12	RW	0x0	pid12_error PID12 Error Interrupt Enable 1:enabled 0:disabled
11	RW	0x0	pid11_error PID11 Error Interrupt Enable 1:enabled 0:disabled
10	RW	0x0	pid10_error PID10 Error Interrupt Enable 1:enabled 0:disabled
9	RW	0x0	pid9_error PID9 Error Interrupt Enable 1:enabled 0:disabled
8	RW	0x0	pid8_error PID8 Error Interrupt Enable 1:enabled 0:disabled
7	RW	0x0	pid7_error PID7 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
6	RW	0x0	pid6_error PID6 Error Interrupt Enable 1:enabled 0:disabled
5	RW	0x0	pid5_error PID5 Error Interrupt Enable 1:enabled 0:disabled
4	RW	0x0	pid4_error PID4 Error Interrupt Enable 1:enabled 0:disabled
3	RW	0x0	pid3_error PID3 Error Interrupt Enable 1:enabled 0:disabled
2	RW	0x0	pid2_error PID2 Error Interrupt Enable 1:enabled 0:disabled
1	RW	0x0	pid1_error PID1 Error Interrupt Enable 1:enabled 0:disabled
0	RW	0x0	pid0_error PID0 Error Interrupt Enable 1:enabled 0:disabled

TSP_PTIX_PID_INT_ENA3

Address: Operational Base + offset (0x0130)

PID Interrupt Enable Register 3

Bit	Attr	Reset Value	Description
31	RW	0x0	pid63_error PID63 Error Interrupt Enable 1:enabled 0:disabled
30	RW	0x0	pid62_error PID62 Error Interrupt Enable 1:enabled 0:disabled
29	RW	0x0	pid61_error PID61 Error Interrupt Enable 1:enabled 0:disabled
28	RW	0x0	pid60_error PID60 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
27	RW	0x0	pid59_error PID59 Error Interrupt Enable 1:enabled 0:disabled
26	RW	0x0	pid58_error PID58 Error Interrupt Enable 1:enabled 0:disabled
25	RW	0x0	pid57_error PID57 Error Interrupt Enable 1:enabled 0:disabled
24	RW	0x0	pid56_error PID56 Error Interrupt Enable 1:enabled 0:disabled
23	RW	0x0	pid55_error PID55 Error Interrupt Enable 1:enabled 0:disabled
22	RW	0x0	pid54_error PID54 Error Interrupt Enable 1:enabled 0:disabled
21	RW	0x0	pid53_error PID53 Error Interrupt Enable 1:enabled 0:disabled
20	RW	0x0	pid52_error PID52 Error Interrupt Enable 1:enabled 0:disabled
19	RW	0x0	pid51_error PID51 Error Interrupt Enable 1:enabled 0:disabled
18	RW	0x0	pid50_error PID50 Error Interrupt Enable 1:enabled 0:disabled
17	RW	0x0	pid49_error PID49 Error Interrupt Enable 1:enabled 0:disabled
16	RW	0x0	pid48_error PID48 Error Interrupt Enable 1:enabled 0:disabled
15	RW	0x0	pid47_error PID47 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
14	RW	0x0	pid46_error PID46 Error Interrupt Enable 1:enabled 0:disabled
13	RW	0x0	pid45_error PID45 Error Interrupt Enable 1:enabled 0:disabled
12	RW	0x0	pid44_error PID44 Error Interrupt Enable 1:enabled 0:disabled
11	RW	0x0	pid43_error PID43 Error Interrupt Enable 1:enabled 0:disabled
10	RW	0x0	pid42_error PID42 Error Interrupt Enable 1:enabled 0:disabled
9	RW	0x0	pid41_error PID41 Error Interrupt Enable 1:enabled 0:disabled
8	RW	0x0	pid40_error PID40 Error Interrupt Enable 1:enabled 0:disabled
7	RW	0x0	pid39_error PID39 Error Interrupt Enable 1:enabled 0:disabled
6	RW	0x0	pid38_error PID38 Error Interrupt Enable 1:enabled 0:disabled
5	RW	0x0	pid37_error PID37 Error Interrupt Enable 1:enabled 0:disabled
4	RW	0x0	pid36_error PID36 Error Interrupt Enable 1:enabled 0:disabled
3	RW	0x0	pid35_error PID35 Error Interrupt Enable 1:enabled 0:disabled
2	RW	0x0	pid34_error PID34 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
1	RW	0x0	pid33_error PID33 Error Interrupt Enable 1:enabled 0:disabled
0	RW	0x0	pid32_error PID32 Error Interrupt Enable 1:enabled 0:disabled

TSP_PTIX_PCR_INT_STS

Address: Operational Base + offset (0x0134)

PTI PCR Interrupt Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	W1C	0x0	pcr7_done PCR7 Status 1: done; 0: not done;
6	W1C	0x0	pcr6_done PCR6 Status 1: done; 0: not done;
5	W1C	0x0	pcr5_done PCR5 Status 1: done; 0: not done;
4	W1C	0x0	pcr4_done PCR4 Status 1: done; 0: not done;
3	W1C	0x0	pcr3_done PCR3 Status 1: done; 0: not done;
2	W1C	0x0	pcr2_done PCR2 Status 1: done; 0: not done;
1	W1C	0x0	pcr1_done PCR1 Status 1: done; 0: not done;
0	W1C	0x0	pcr0_done PCR0 Status 1: done; 0: not done;

TSP_PTIX_PCR_INT_ENA

Address: Operational Base + offset (0x0138)

PTI PCR Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	pcr7_done_ena pcr7 done interrupt enable 1: enabled; 0: disabled;
6	RW	0x0	pcr6_done_ena pcr6 done interrupt enable 1: enabled; 0: disabled;
5	RW	0x0	pcr5_done_ena pcr5 done interrupt enable 1: enabled; 0: disabled;
4	RW	0x0	pcr4_done_ena pcr4 done interrupt enable 1: enabled; 0: disabled;
3	RW	0x0	pcr3_done_ena pcr3 done interrupt enable 1: enabled; 0: disabled;
2	RW	0x0	pcr2_done_ena pcr2 done interrupt enable 1: enabled; 0: disabled;
1	RW	0x0	pcr1_done_ena pcr1 done interrupt enable 1: enabled; 0: disabled;
0	RW	0x0	pcr0_done_ena pcr0 done interrupt enable 1: enabled; 0: disabled;

TSP_PTIX_PCRn_CTRL

Address: Operational Base + offset (0x013c)

PID PCR Control Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:1	RW	0x0000	pid PCR Extraction PID number This 13-bit field sets the PID number that needs PCR extraction.
0	RW	0x0	on PCR Extraction Switch 1'b1: PCR extraction switched on ; 1'b0: PCR extraction switched off ;

TSP_PTIX_PCRn_H

Address: Operational Base + offset (0x015c)

High Order PCR value

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	pcr PCR[32] pcr[32]

TSP_PTIX_PCRn_L

Address: Operational Base + offset (0x0160)

Low Order PCR value

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pcr pcr[31:0] pcr[31:0]

TSP_PTIX_DMA_STS

Address: Operational Base + offset (0x019c)

LLP DMA Interrupt Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	W1C	0x0	llp_error LLP DMA Error Status 1: error response during DMA transaction; 0: no error response during DMA transaction;
0	W1C	0x0	llp_done LLP DMA Done Status 1: DMA transaction completed; 0: DMA transaction not completed;

TSP_PTIX_DMA_ENA

Address: Operational Base + offset (0x01a0)

DMA Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	llp_error_ena LLP DMA Error Interrupt Enable 1: enabled 0: disabled
0	RW	0x0	llp_done_ena LLP DMA Done Interrupt Enable 1: enabled 0: disabled

TSP_PTIX_DATA_FLAG0

Address: Operational Base + offset (0x01a4)

PTI_PID_WRITE Flag 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data_write_flag_0 From PID0 TO PID31

TSP_PTIX_DATA_FLAG1

Address: Operational Base + offset (0x01a8)

PTI_PID_WRITE Flag 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data_write_flag_1 From PID32 TO PID63

TSP_PTIX_LIST_FLAG

Address: Operational Base + offset (0x01ac)

PTIx_LIST_WRITE Flag

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	list_write_flag From PID0 TO PID15

TSP_PTIX_DST_STS0

Address: Operational Base + offset (0x01b0)

PTI Destination Status Register

Bit	Attr	Reset Value	Description
31:0	W1C	0x00000000	demux_dma_status_0 From 0 to 31 channel

TSP_PTIX_DST_STS1

Address: Operational Base + offset (0x01b4)

PTI Destination Status Register

Bit	Attr	Reset Value	Description
31:0	W1C	0x00000000	demux_dma_status_0 From 32 to 63 channel

TSP_PTIX_DST_ENA0

Address: Operational Base + offset (0x01b8)

PTI Destination Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	demux_dma_enable_0 From 0 to 31 channel

TSP_PTIX_DST_ENA1

Address: Operational Base + offset (0x01bc)

PTI Destination Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	demux_dma_enable_1 From 32 to 63 channel

TSP_PTIX_ECWN_H

Address: Operational Base + offset (0x0200)

The Even Control Word High Order

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ecw_h The Even Control Word High Order ECW[63:32]

TSP_PTIX_ECWN_L

Address: Operational Base + offset (0x0204)

The Even Control Word Low Order

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ecw_l The Even Control Word Low Order ECW[31:0]

TSP_PTIX_OCWN_H

Address: Operational Base + offset (0x0208)

The Odd Control Word High Order

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ocw_h The Odd Control Word High order OCW[63:32]

TSP_PTIX_OCWN_L

Address: Operational Base + offset (0x020c)

The Odd Control Word Low Order

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ocw_l The Odd Control Word Low Order OCW[31:0]

TSP_PTIX_PIDn_CTRL

Address: Operational Base + offset (0x0300)

PID Channel Control Register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	cw_num Control Word Order Number This fields indicates the corresponding order number of control word to be used to descramble TS packets.
15:3	RW	0x0000	pid PID number This 13-bit sets the desired PID number to be processed by PTI channel.
2	RW	0x0	csa_on Descrambling Switch 1'b1: Descrambling function turned on; 1'b0: Descrambling function turned off;

Bit	Attr	Reset Value	Description
1	RWSC	0x0	clear PID Channel Clear Write 1 to clear PID channel. This bit will be set to 0 if the channel is clear.
0	RWSC	0x0	en PID Channel Enable Write 1 to enable channel. Write 0 to this bit will not take any effect. This bit will be 0 when channel is cleared.

TSP_PTIX_PIDn_BASE

Address: Operational Base + offset (0x0400)

PTI Data Memory Buffer Base Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI Data Memory Buffer Base Address PTI Data Memory Buffer Base Address

TSP_PTIX_PIDn_TOP

Address: Operational Base + offset (0x0404)

PTI Data Memory Buffer Top Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI Data Memory Buffer Top Address PTI Data Memory Buffer Top Address

TSP_PTIX_PIDn_WRITE

Address: Operational Base + offset (0x0408)

PTI Data Memory Buffer Hardware Writing Address

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	address PTI Data Memory Buffer Hardware Writing Address PTI Data Memory Buffer Hardware Writing Address

TSP_PTIX_PIDn_READ

Address: Operational Base + offset (0x040c)

PTI Data Memory Buffer Software Reading Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI Data Memory Buffer Software Reading Address PTI Data Memory Buffer Software Reading Address

TSP_PTIX_LISTn_BASE

Address: Operational Base + offset (0x0800)

PTI List Memory Buffer Base Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI Data Memory Buffer Software Reading Address PTI Data Memory Buffer Software Reading Address

TSP_PTIX_LISTn_TOP

Address: Operational Base + offset (0x0804)

PTI List Memory Buffer Top Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI List Memory Buffer Top Address PTI List Memory Buffer Top Address

TSP_PTIX_LISTn_WRITE

Address: Operational Base + offset (0x0808)

PTI List Memory Buffer Hardware Writing Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI List Memory Buffer Hardware Writing Address PTI List Memory Buffer Hardware Writing Address

TSP_PTIX_LISTn_READ

Address: Operational Base + offset (0x080c)

PTI List Memory Buffer Software Reading Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI List Memory Buffer Software Reading Address PTI List Memory Buffer Software Reading Address

TSP_PTIX_PIDn_CFG

Address: Operational Base + offset (0x0900)

PID Demux Configure Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	filter_en Filter Byte Enable The proper position of filter byte Enable. For Section filter, the 1st,4th,5th,..18th byte of section header are used to be filtered; For PES filter, the 4th,7th,8th...21th byte of pes header are used to be filtered.
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	scd_en Start Code Detection Switch Start code detection 1: enabled; 0: disabled; This bit is only valid when n < 16.
10	RW	0x0	cni_on Current Next Indicator Abort when current_next_indicator == 1'b1, 1'b1: abort ; 1'b0: do nothing ;
9:8	RW	0x0	filt_mode Section Filter Mode Filter Mode when the filter mode is configured as section filter. 2'b00: stop per unit; 2'b01: full stop; 2'b10: recycle, update when version number change 2'b11: reserved
7:6	RW	0x0	video_type Video filtering Type 2'b00: MPEG2 2'b01: H264 2'b10: VC-1 2'b11: Reserved
5:4	RW	0x0	filt_type Filter Type 2'b00: section filtering; 2'b01: pes filtering; 2'b10: es filtering; 2'b11: ts filtering; if n>=16, it is reserved as only section filtering, other values are invalid.
3	RW	0x1	cc_abort Continue Counter Error Abort when continuity counter error happens: 1: abort; 0: do nothing;
2	RW	0x0	tei_abort Ts_error_indicator Abort when ts_error_indicator == 1: 1'b1: abort ; 1'b0: do nothing;
1	RW	0x0	crc_abort CRC Error Abort This bit is valid only when crc_on == 1'b1. When crc error happens, 1'b1: abort ; 1'b0: do nothing.

Bit	Attr	Reset Value	Description
0	RW	0x0	crc_on CRC Check 1'b1: CRC check function turned on 1'b0: CRC check function turned off

TSP_PTIX_PIDn_FILT_0

Address: Operational Base + offset (0x0904)

Fliter Word 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	filt_byte_3 Fliter Byte 2 This byte refers to 6th byte of section header or 9th byte of pes header
23:16	RW	0x00	filt_byte_2 Fliter Byte 2 This byte refers to 5th byte of section header or 8th byte of pes header
15:8	RW	0x00	filt_byte_1 Fliter Byte 1 This byte refers to 4th byte of section header or 7th byte of pes header
7:0	RW	0x00	filt_byte_0 Fliter Byte 0 This byte refers to 1st byte of section header or 4th byte of pes header

TSP_PTIX_PIDn_FILT_1

Address: Operational Base + offset (0x0908)

Fliter Word 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	filt_byte_3 Fliter Byte 2 This byte refers to 10th byte of section header or 13rd byte of pes header
23:16	RW	0x00	filt_byte_2 Fliter Byte 2 This byte refers to 9th byte of section header or 12nd byte of pes header
15:8	RW	0x00	filt_byte_1 Fliter Byte 1 This byte refers to 8th byte of section header or 11st byte of pes header
7:0	RW	0x00	filt_byte_0 Fliter Byte 0 This byte refers to 7th byte of section header or 10th byte of pes header

TSP_PTIX_PIDn_FILT_2

Address: Operational Base + offset (0x090c)

Fliter Word 2

Bit	Attr	Reset Value	Description
31:24	RW	0x00	filt_byte_3 Fliter Byte 2 This byte refers to 14th byte of section header or 17th byte of pes header
23:16	RW	0x00	filt_byte_2 Fliter Byte 2 This byte refers to 13rd byte of section header or 16th byte of pes header
15:8	RW	0x00	filt_byte_1 Fliter Byte 1 This byte refers to 12nd byte of section header or 15th byte of pes header
7:0	RW	0x00	filt_byte_0 Fliter Byte 0 This byte refers to 11st byte of section header or 14th byte of pes header

TSP_PTIX_PIDn_FILT_3

Address: Operational Base + offset (0x0910)

Fliter Word 3

Bit	Attr	Reset Value	Description
31:24	RW	0x00	filt_byte_3 Fliter Byte 2 This byte refers to 18th byte of section header or 21st byte of pes header
23:16	RW	0x00	filt_byte_2 Fliter Byte 2 This byte refers to 17th byte of section header or 20th byte of pes header
15:8	RW	0x00	filt_byte_1 Fliter Byte 1 This byte refers to 16th byte of section header or 19th byte of pes header
7:0	RW	0x00	filt_byte_0 Fliter Byte 0 This byte refers to 15th byte of section header or 18th byte of pes header

42.5 Interface Description

Table 42-1 TSP Interface Description

Module Pin	IO	Pad Name	IOMUX Setting
ts_data0	I/O	IO_UART1BBsin_TS0data0_BBgpio5b0	GPIO5B_IOMUX[1:0]= 2'b10
ts_data1	I/O	IO_UART1BBsout_TS0data1_BBgpio5b1	GPIO5B_IOMUX[3:2]= 2'b10
ts_data2	I/O	IO_UART1BBCtsn_TS0data2_BBgpio5b2	GPIO5B_IOMUX[5:4]= 2'b10
ts_data3	I/O	IO_UART1BBrtsn_TS0data3_BBgpio5b3	GPIO5B_IOMUX[7:6]= 2'b10
ts_data4	I/O	IO_SPI0clk_TS0data4_UART4EXPcts_BBgpio5b4	GPIO5B_IOMUX[9:8]= 2'b10
ts_data5	I/O	IO_SPI0csn0_TS0data5_UART4EXPrtsn_BBgpio5b5	GPIO5C_IOMUX[11:10]= 2'b10
ts_data6	I/O	IO_SPI0txd_TS0data6_UART4EXPouts_BBgpio5b6	GPIO5B_IOMUX[13:12]= 2'b10
ts_data7	I/O	IO_SPI0rxr_TS0data7_UART4EXPsin_BBgpio5b7	GPIO5B_IOMUX[15:14]= 2'b10

ts_valid	I/O	IO_TS0valid_BBgpio5c1	GPIO5C_IOMUX[2]= 1'b1
ts_sync	I/O	IO_SPI0csn1_TS0sync_BBgpio5c0	GPIO5C_IOMUX[1:0]= 2'b10
ts_err	I/O	IO_TS0err_BBgpio5c3	GPIO5C_IOMUX[6]= 1'b1
ts_clk	I/O	IO_TS0clk_BBgpio5c2	GPIO5C_IOMUX[4]= 1'b1
hsadc_data0	I	IO_CIFdata2_HOSTdin0_HSADCdata0_DVPgpio2a0	GPIO2A_IOMUX[1:0]= 2'b11
hsadc_data1	I	IO_CIFdata3_HOSTdin1_HSADCdata1_DVPgpio2a1	GPIO2A_IOMUX[3:2]= 2'b11
hsadc_data2	I	IO_CIFdata4_HOSTdin2_HSADCdata2_DVPgpio2a2	GPIO2A_IOMUX[5:4]= 2'b11
hsadc_data3	I	IO_CIFdata5_HOSTdin3_HSADCdata3_DVPgpio2a3	GPIO2A_IOMUX[7:6]= 2'b11
hsadc_data4	I	IO_CIFdata6_HOSTckinp_HSADCdata4_DVPgpio2a4	GPIO2A_IOMUX[9:8]= 2'b11
hsadc_data5	I	IO_CIFdata7_HOSTckinn_HSADCdata5_DVPgpio2a5	GPIO2A_IOMUX[11:10]= 2'b11
hsadc_data6	I	IO_CIFdata8_HOSTdin4_HSADCdata6_DVPgpio2a6	GPIO2A_IOMUX[13:12]= 2'b11
hsadc_data7	I	IO_CIFdata9_HOSTdin5_HSADCdata7_DVPgpio2a7	GPIO2A_IOMUX[15:14]= 2'b11
hsadc_valid	I	IO_CIFhref_HOSTdin7_HSADCTSvalid_DVPgpio2b1	GPIO2B_IOMUX[3:2]= 2'b11
hsadc_sync	I	IO_CIFVsync_HOSTdin6_HSADCTSsync_DVPgpio2b0	GPIO2B_IOMUX[1:0]= 2'b11
hsadc_err	I	IO_CIFclkout_HOSTTwkreq_HSADCTSfail_DVPgpio2b3	GPIO2B_IOMUX[7:6]= 2'b01
gps_clk	I	IO_CIFclkin_HOSTTwkack_GPSclk_HSADCclkout_DVPgpio2b2	GPIO2B_IOMUX[5:4]= 2'b11
gpst1_clk	I	IO_UART3GPSctsn_GPSrfclk_GPST1clk_GPIO30gpio7b1	GPIO7B_IOMUX[3:2]= 2'b11

Notes: I=input, O=output, I/O=input/output, bidirectional

42.6 Application Notes

42.6.1 Overall Operation Sequence

- Enable desired modules to work by writing correspond bit with '1' in TSP_GCFG. Note: it is important to do this step at first, otherwise writing the corresponding registers will not take effect.
- Set up TS configuration by writing corresponding registers.
- Wait for the interrupts to pick up the desired TS packets following the rules detailed in the following section.

Note: PTI1 addr = PTI0 addr + 0x1000;

42.6.2 TS Source

TS source can be chosen by writing the bit 9 of TSP_PTIx_CTRL(x=0,1), '1' for demodulator, '0' for local memory.

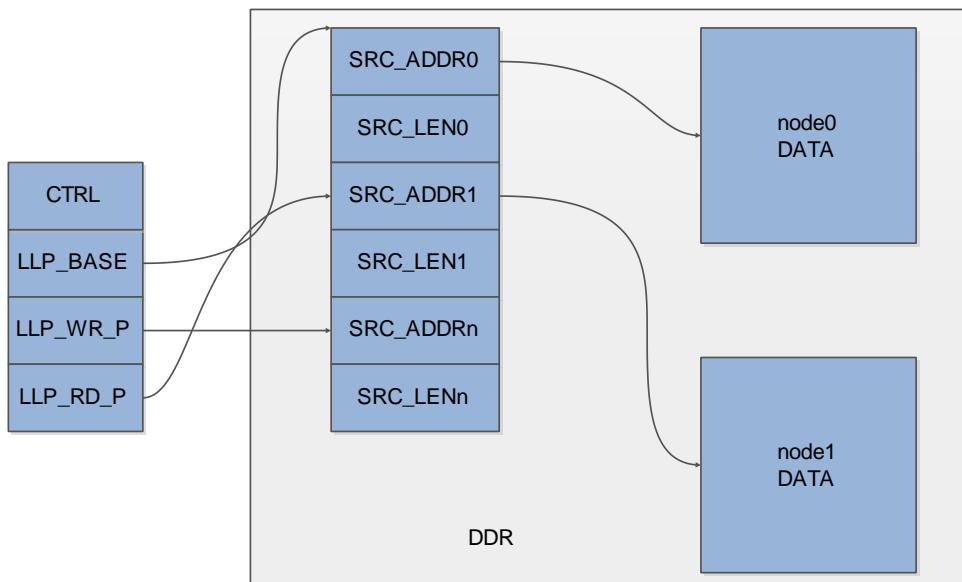
1.TS_IN Interface

Writing bit 10 of TSP_PTIx_CTRL to choose bit ordering, and writing bit [12:11] to choose input TS mode.

TS_IN interface supports 4 input TS stream mode: sync/valid serial mode, sync/valid parallel mode, sync/burst parallel mode, nosync/valid parallel mode.

2.Local Memory

PTI also can process the TS data read from local memory by using LLP DMA mode.



Chapter 1 Write PTIx_LL_P_BASE with the list base address;

Chapter 2 Starting from the list base address, write the list nodes. One list node comprised of two words. The first word describes the TS data base address, the second one describes the length of TS data in unit of word.

Chapter 3 Write the PTIx_LL_WRITE with the number of words that you have written in list memory. Note it is not the number of LLP nodes, so that the number you are writing should be an even one.

Chapter 4 Write PTIx_LL_CFG with the configuration you want. Write the bit 0 with 1 to start LLP DMA. If all the list nodes are written, don't forget to write 1 to bit 3 to tell DMAC that the configuration is finished.

Note:

- The MSB(bit7) of the 8-bit pointer in the PTIx_LL_Write and PTIx_LL_Read is used as the flag bit, and remaining 7 bits are used for addressing. Therefore the the pointer is referred to 7-bit space, not 8-bit space, and remember write the pointer with the correct flag bit. For example, if you have configured 63 LLP nodes and then you have to write the 64th LLP node starting from the list base address,
- PTIx_LL_READ informs that how many words has been processed by LLP DMA. An interrupt may be generated when number of the processed words has reach to the threshold set in the PTIx_LL_CFG.
- If you write the PTIx_LL_Write several times in a complete DMA transaction, it is important to notice the flag bit of PTIx_LL_Write, and never make the writing pointer catch up with the reading pointer.

42.6.3 TS Synchronous Operation

Synchronous mode and Bypass mode can be switched by writing bit 15 of TSP_PTIX_CTRL.

In the synchronous mode, 188/192/204 byte TS packets are supported and self-adjusted. Set up locked times in TSP_PTIX_CTRL to inform the successive

times of TS packet header detection needs to lock the header of TS packets when in the unlocked mode, and set up unlocked times to informs the successive times of TS packet header error needs to re-lock header of TS packets in the locked mode. It is recommended to use 2-3 as the locked times to quickly and correctly locked the header, and 2-3 as unlocked times to avoid unnecessarily entering into unlocked searching mode.

In the bypass mode, the input TS data will not be re-synchronized and directly fed into the PTI channel.

42.6.4 Descrambling Operation

Descrambler can achieve PES or TS level descrambling which conforms to the CSA v2.0.

- Enable the channel you want by writing 1 to bit 0 of TSP_PTIX_PIDn_CTRL ($x=0\sim1$, $n=0\sim64$);
- Set the desired PID number
- Turn on descrambling function by setting 1 to bit 2. If the corresponding CW is available or TS is required to be left undescrambled, CSA_ON bit is set to 0;
- Choose corresponding Control Word by setting bit[19:16], and 16 set Control Word are available to be chosen. Don't forget Control Word should be prepared before the descrambling function is enabled.

Note: If the enabled channel is needed to be disabled, write the CLEAR bit to disabled the channel rather than write '0' to EN bit.

42.6.5 Demux Operation

Refer to TSP_PTIX_PIDn_CFG for Demux operation. The software users should be familiar with the demux knowledge.

Users should create a separate memory buffer to receive the processed data for each desired PID channel, and write the base and top address information of the memory buffer into TSP_PTIX_PIDn_BASE and TSP_PTIX_PIDn respectively. Also initial writing address and reading address, normally the same as base address, are also needed to be written into TSP_PTIX_PIDn_WRITE and TSP_PTIX_PIDn_READ respectively. For ES/PES filter, another separate memory needs to be created to store list data, which is used to assist obtaining PES/ES data. List base address, top address, initial writing address and reading address are also needed to write into corresponding registers.

Note:

For channel whose PID channel number larger than 15, the channels can only be used section filter. For others, there is no such limit. They can be configured as section filter, pes filter, es filter or ts filter.

Data memory address boundary should be aligned with word-size, and list memory address boundary should be aligned with word size. If the memory buffer is not larger to store processed data so that writing address reaches the top address, TSP will return to the base address to write data. So fetch the data in time, don't make the writing address catches up with reading address. The list memory buffer has the same issue.

1.Demux data obtain

● TS filter

To obtain TS data and section data, when an desired PID done interrupt is generated, read TSP_PTIX_PIDn_READ firstly to know the address that last reading stops, and then read TSP_PTIX_PIDn_WRITE to know the address that hardware has reached. For ts data, start from the TSP_PTIX_PIDn_READ address to get the TS packet data, and stop at the address you want. However, the ending address should not catch up with writing address. It is recommended to obtain the TS data in the unit of TS packet which is 47-word size. At last, don't forget to write the ending address into TSP_PTIX_PIDn_READ to leave a hint where current reading stops.

B. Section filter

Section filter can run three mode to meet different needs: stop-per-unit; full stop; recycle , update when version number change. The PID done interrupt will be generated after each part of a complete section is processed in the first mode, and the PID done will be generated only after the whole section is completed in the last two modes. In the frist two mode, the PID channel will be disabled after the whole section is completed. In the recycle mode, the channel will remain active and start a new section processing when the version number changes. Section filter also supports 16-byte filtering function, which can assign 1st , 4th to 18th byte to be filtered.

The process to obtain section data is similar to the process for TS data. After a PID done interrupt done is generated, refer to the corresponding PID error status register to check if the section data is correct. Read the frist word of the section start address to know the total length of the section according to the format of section data.

$$\begin{aligned}\text{Section Length} &= \{\text{First Word}[11:8], \text{First Word}[23:16]\}; \\ \text{Total Length} &= \text{Section Length};\end{aligned}$$

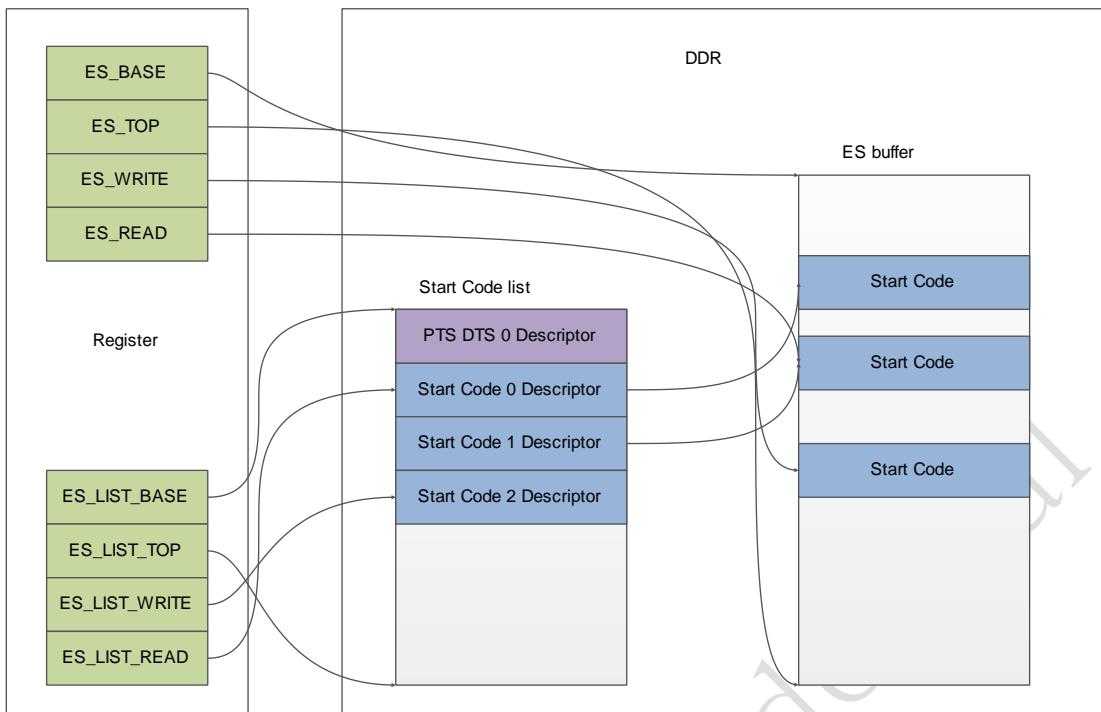
Then start to fetch section data according to the total length. Again don't forget to write the stopped address.

C. PES/ES filter

PES filter supports 16-byte filtering function, which can assign 4th , 7th to 21st byte to be filtered.

ES filter supports start code detection, including MPEG2 start code 0x000001b3, 0x000000100, VC-1 start code 0x0000010d, 0x000010f, H264 start code 0x000001.

To obtain the pes/es data, the assistant of list descriptor is needed.



List memory buffer contains descriptors which contains information to obtain es/pes data which are stored in data memory buffer.

The descriptor stored in list memory buffer can be separated into two groups: PTS_DTS Descriptor and Start Code Descriptor. The descriptor is composed by 4 word content, word_0, word_1, word_2 and word_3. The word_x (x means the sequence number in a descriptor, and they are stored in the memory in sequence order). The format of the 4 words are listed as follows:

(1) start code descriptor

Word_0:

Word_0[29:28] indicates the attributes of the bytes of the pointed word. 2'b00 means the whole word belongs to the new ES/PES packet; 2'b01 means that word[7:0] belongs to the previous packet, and the remaining bytes belong to the new packet; 2'b10 means that word[15:0] belongs to the previous packet, and the remaining bytes belong to the new packet; 2'b11 means that word[23:0] belongs to the previous packet, and the remaining bytes belong to the new packet. This pointed word is the word where start code starts, word_2 describes the location of start code.

Word_0[27:24] is equal to 0x0 in the start code descriptor. Users can used to tell two kinds of descriptor.

If the video type is H.264, word_0[23:8] means first_mb_in slice, and word_0 means nal_nuit_type.

Word_1:

the start code of stream.

Word_2:

DDR offset address in the DDR of the word where the start code is located.

Word_3:

0x0

(2) PTS_DTS Descriptor

Word_0:

Word_0[29:28]: the same as start code descriptor
Word_0[27:24]: 0x1 in PTS_DTS descriptor.
Word_0[3] : PTS[32];
Word_0[2] : DTS[32];
Word_0[1:0] : pts_dts_flag;

Word_1:

DDR offset address of the word that valid data starts.

Word_2:

PTS[31:0]

Word_3

DTS[31:0]

To obtain PES data or ES data when start code detection is disabled, use PTS_DTS descriptor. To obtain ES data when start code detection is enabled, use start code descriptor.

When a PID done interrupt is generated, make sure there is no corresponding PID error generated. Read the TSP_PTIx_LISTn_READ to know the list reading address in the last time. Start from here, read the 4-word descriptor one by one to know the offset of the packets. Refer to the offset in the DDR where in the data memory buffer to obtain data. Finally write TSP_PTIx_LISTn_READ and TSP_PTIx_PIDn_READ with corresponding reading address.

42.6.6 TS Out Interface

All the configuration is done by writing TSP_TSOUT_CTRL. Before programming this register, make sure that you have enabled the TS OUT interface. If you want to disable TS out interface, write '0' to the START bit(bit 0) of TSP_TSOUT_CTRL, and then disable it in the TSP_GFCG.

Each PTI channel can provide TS out interface with PID-filtering TS Packets or non-PID-filtering TS packets, and therefore there are totally 4 sources can be chosen for TS out interface.

42.6.7 PVR

PVR module provide you with the function to record the programs you want. The 4 sources can be assigned with PVR, and they are the same as TS out interface.

Assign the PVR length and PVR address, and then configure TSP_PVR_CTRL to start PVR module. If you want to stop PVR function during recording, write '1' to STOP bit (bit 0) to TSP_PVR_CTRL to stop it. Remember to take care of the status of PVR_ON bit of TSP_GFCG when programming the PVR-related registers.

42.6.8 PCR extraction

PCR extraction can be enabled by configure PTIx_PCRn_CTRL. Then if the PID-matched TS data contain PCR field, the 33-bit PCR_base field will be written

corresponding PTIx_PCRn_H and PTIx_PCRn_L registers. An interrupt will be asserted if PCR interrupt is enabled.

Rockchip Confidential

Chapter 43 Smart Card Controller

43.1 Overview

The Smart Card Reader (SCR) is a communication controller that transmits data between the superior system and the Smart Card. The controller can perform a complete smart card session, including card activation, card deactivation, cold/warm reset, Answer to Reset (ATR) response reception, data transfers, etc.

SCR supports the following features:

- Supports the ISO/IEC 7816-3:1997(E) and EMV2000 (4.0) specifications
- Performs functions needed for complete smart card sessions, including:
 - Card activation and deactivation
 - Cold/warm reset
 - Answer to Reset (ATR) response reception
 - Data transfers to and from the card
- Extensive interrupt support system
- Adjustable clock rate and bit (baud) rate
- Configurable automatic byte repetition
- Handles commonly used communication protocols:
 - T=0 for asynchronous half-duplex character transmission
 - T=1 for asynchronous half-duplex block transmission
- Automatic convention detection
- Configurable timing functions:
 - Smart card activation time
 - Smart card reset time
 - Guard time
 - Timeout timers
- Automatic operating voltage class selection
- Supports synchronous and any other non-ISO 7816 and non-EMV cards
- Advanced Peripheral Bus (APB) slave interface for easy integration with AMBA-based host systems

43.2 Block Diagram

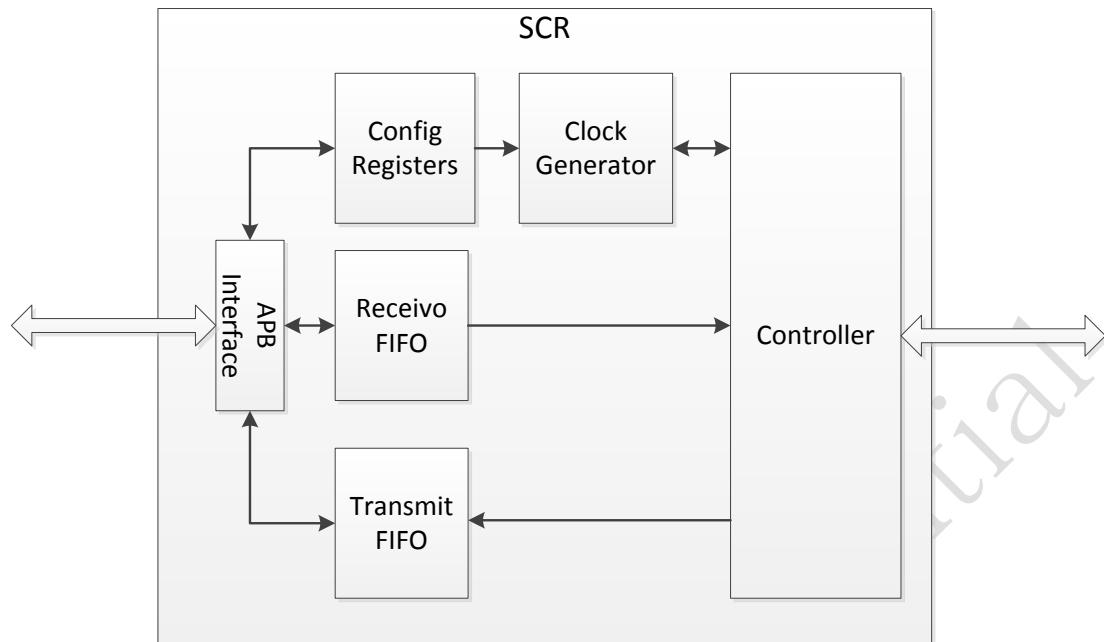


Fig.43-1 SCR Block Diagram

The host processor gets access to PWM Register Block through the APB slave interface.

43.2.1 APB Interface

The host processor gets access to PWM Register Block through the APB slave interface.

43.2.2 Configuration Registers

The Configuration Registers block provides control over all functions of the Smart Card Reader

43.2.3 Controller

The Controller is the main block in the SCR core. This block controls receiving characters transmitted by the Smart Card, storing them in the RX FIFO, and transmitting them to the Smart Card. This block also performs card activation, deactivation, and cold and warm reset. After the card is reset, the Answer To Reset (ATR) sequence is received by the controller and stored in RX FIFO.

The parallel to serial conversion needed to transmit data from a Smart Card Reader to a Smart Card and the serial to parallel conversion needed to transmit data in the opposite direction is performed by the UART. The UART also performs the guardtime, parity checking and character repeating functions.

43.2.4 Receive FIFO

The Receive FIFO is used to store the data received from the Smart Card until the data is read out by the superior system.

43.2.5 Transmit FIFO

The Transmit FIFO is used to store the data to be transmitted to the Smart Card.

43.2.6 Clock Generator

The Clock Generator generates the Smart Card Clock signal and the Baud Clock Impulse signal, used in timing the Smart Card Reader.)

43.3 Function Description

A Smart Card session consists of following stages:

1. Smart Card insertion
2. Activation of contacts and cold reset sequence
3. Answer To Reset sequence (ATR)
4. Execution of transaction
5. Deactivation of contacts
6. Smart Card removal

43.3.1 Smart Card Insertion

A Smart Card session starts with the insertion of the Smart Card. This event is signaled to the SCR using the SCDETECT input. The SCPRESENT bit is set and also the SCINS interrupt is asserted (if enabled).

When the external card detect switch is not used, the input pin SCDETECT must be tied to inactive state.

43.3.2 Automatic operating voltage class selection

There are three operating classes (1.8V - class C, 3V - class B and 5V - class A) defined in ISO/IEC 7816-3(2006) specification. Only 1.8V and 3.3V are supported by the SCR.

Before the activation of contacts, operating classes have to be enabled via bits VCC18, VCC33 in CTRL2 register. In case that no operating class is enabled, the controller performs activation for all two voltage classes (1.8V, 3V) in sequence.

When Smart Card Reader performs activation of contacts the lowest enabled voltage class is automatically applied first. When the first character start bit of ATR sequence is received, the selected voltage class is correct (even if the ATR is then received with errors). When the ATR sequence reception does not start, ATRFAIL interrupt is not activated, deactivation is performed and next higher enabled voltage class is applied. If the ATR sequence reception does not start and no other higher class is enabled was already applied the ATRFAIL interrupt is activated and the last applied voltage class remains active.

After the automatic voltage class selection is finished the selected class can be read from bits VCC18, VCC33 in CTRL2 register. If the automatic voltage class selection fails, these bits remain untouched.

There is a delay applied between deactivation of contacts with lower voltage

class and activation of contacts with higher voltage class. This delay should be at least 10 ms according to the ISO/IEC 7816-3 specification.

43.3.3 Activation of Contacts and Cold Reset Sequence

When the Smart Card is properly inserted and the ACT bit in CTRL2 register is asserted, the activation of contacts can be started. The duration of each part of the activation is the time T_a , which is equal to the ADEATIME register value. If no Vpp is necessary, the activation and deactivation part of Vpp can be omitted by clearing the AUTOADEAVPP bit in SCPADS register.

The Cold Reset sequence follows immediately after the activation. Time (T_c) is the duration of the Reset. The EMV specification recommends that this value should be between 40000 and 45000. The activation of contacts and cold reset sequence is shown in Fig.43-2.

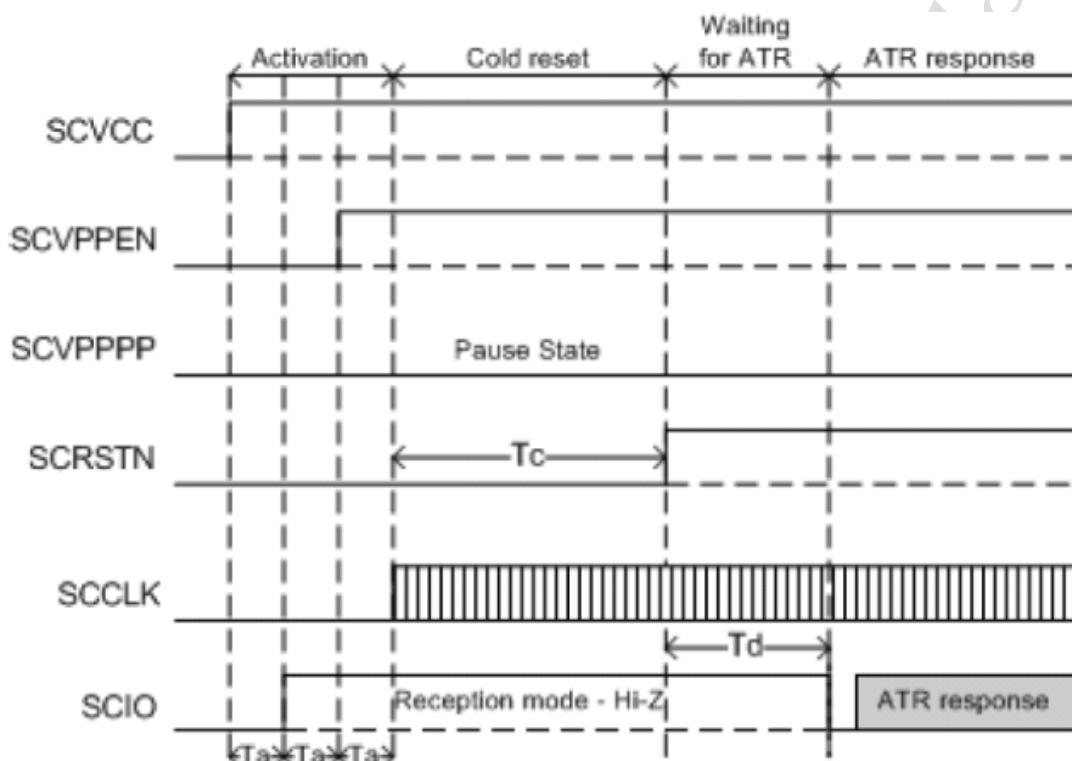


Fig.43-2 Activation, Cold Reset and ATR

43.3.4 Execution of Transaction

All transfers between the Smart Card Reader and a Smart Card are under the control of the superior system. It controls the number of characters sent to the Smart Card and it knows the number of characters expected to be returned from the Smart Card.

43.3.5 Warm Reset

The Warm Reset sequence is initialized by setting the WRST bit in the CTRL2 register to '1'. Smart Card Reader drives the SCRSTN signal to '0' to perform the Warm Reset as shown in Fig.43-3. After the SCRSTN assertion, the Warm Reset sequence then continues the same way as the Cold Reset sequence.

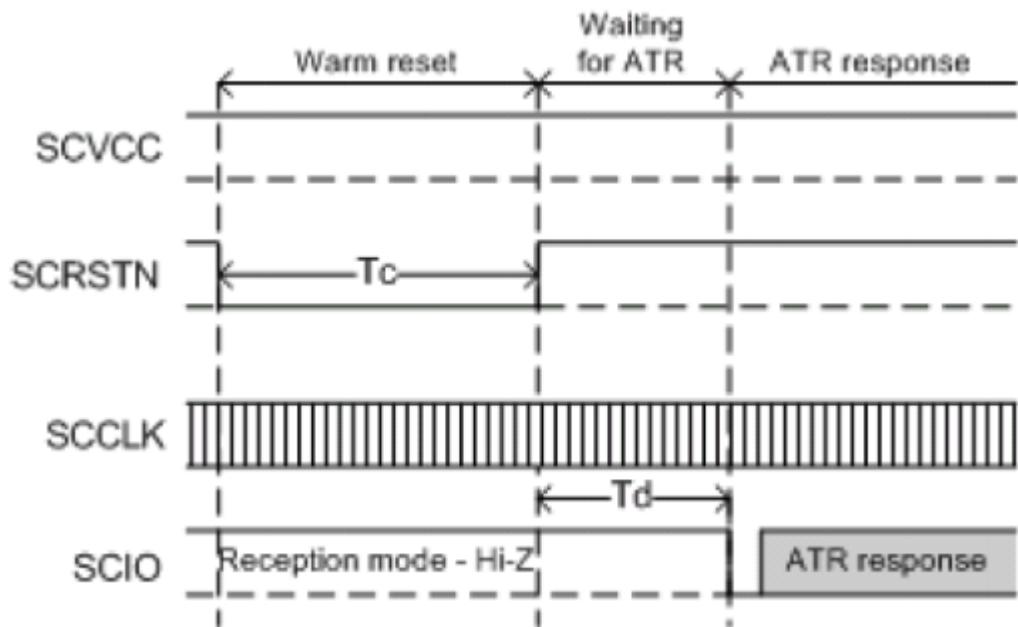


Fig.43-3 Warm Reset and ATR

43.3.6 Deactivation of Contacts

After the smart card reader detects the removal of the smart card (SCREM interrupt) or the superior system initiates deactivation by setting the DEACT bit in the CTRL2 register to '1', the deactivation is performed immediately as shown in . The duration time (T_a), of each part of the deactivation sequence time is defined in the ADEATIME register.

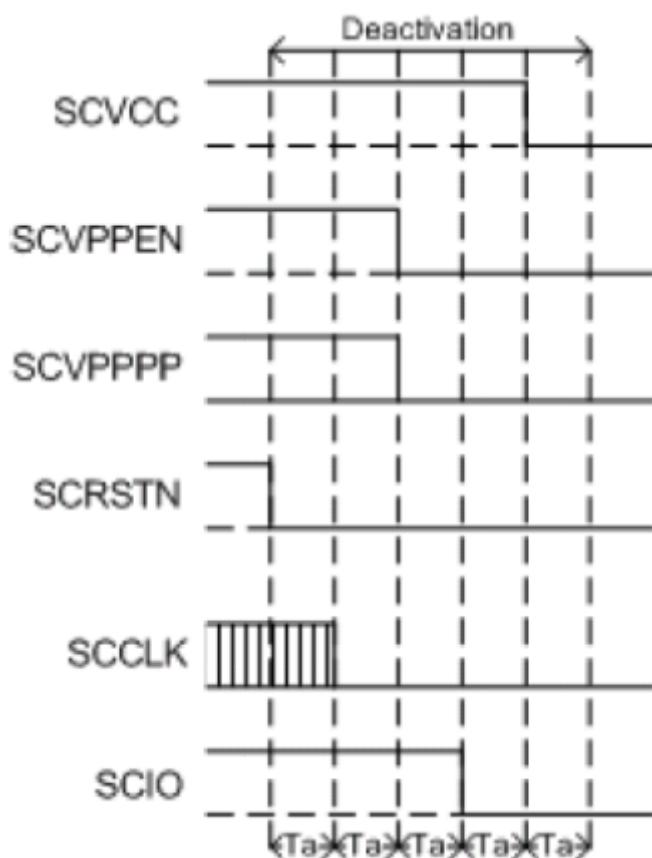


Fig.43-4 Deactivation Sequence

43.4 Register description

43.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
SCR_CTRL1	0x0000	HW	0x0000	Control Register 1
SCR_CTRL2	0x0004	HW	0x0000	Control Register 2
SCR_SCPADS	0x0008	HW	0x0000	Smart Card Pads Register
SCR_INTEN1	0x000c	HW	0x0000	Interrupt Enable Register 1
SCR_INTSTAT1	0x0010	HW	0x0000	Interrupt Status Register 1
SCR_FIFOCTRL	0x0014	HW	0x0000	FIFO Control Register
SCR_LEGTXFICNT	0x0018	B	0x00	Legacy TX FIFO Counter
SCR_LEGRXFICNT	0x0019	B	0x00	Legacy RX FIFO Counter
SCR_RXFITH	0x001c	HW	0x0000	RX FIFO Threshold
SCR_REP	0x0020	B	0x00	Repeat
SCR_SCCDDIV	0x0024	HW	0x0000	Smart Card Clock Divisor
SCR_BAUDDIV	0x0028	HW	0x0000	Baud Clock Divisor
SCR_SCGUTIME	0x002c	B	0x00	Smart Card Guardtime
SCR_ADEATIME	0x0030	HW	0x0000	Activation / Deactivation Time
SCR_LOWRSTTIME	0x0034	HW	0x0000	Reset Duration
SCR_ATRSTARTLIMIT	0x0038	HW	0x0000	ATR Start Limit
SCR_C2CLIM	0x003c	HW	0x0000	Two Characters Delay Limit
SCR_INTEN2	0x0040	HW	0x0000	Interrupt Enable Register 2
SCR_INTSTAT2	0x0044	HW	0x0000	Interrupt Status Register 2

Name	Offset	Size	Reset Value	Description
SCR_TXFITH	0x0048	HW	0x0000	TX FIFO Threshold
SCR_TXFIFOCNT	0x004c	HW	0x0000	TX FIFO Counter
SCR_RXFIFOCNT	0x0050	HW	0x0000	RX FIFO Counter
SCR_BAUTUNE	0x0054	B	0x00	Baud Tune Register
SCR_FIFO	0x0200	B	0x00	FIFO

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

43.4.2 Detail Register Description

SCR_CTRL1

Address: Operational Base + offset (0x0000)

Control Register 1

Bit	Attr	Reset Value	Description
15	RW	0x0	GINTEN Global Interrupt Enable When high, INTERRUPT output assertion is enabled.
14	RO	0x0	reserved
13	RW	0x0	TCKEN TCK enable When enabled all ATR bytes beginning from T0 are being XOR-ed. The result must be equal to TCK byte (when present). If the TCK byte does not match the computed value the ATR is considered to be malformed.
12	RW	0x0	ATRSTFLUSH ATR Start Flush FIFO When enabled, both FIFOs are flushed before the ATR is started.

Bit	Attr	Reset Value	Description
11	RW	0x0	<p>T0T1 T0/T1 Protocol Controls the using of T=0 or T=1 protocol. No character repeating is used when T=1 protocol is selected.</p> <p>The Character Guardtime (minimum delay between the leading edges of two consecutive characters) is reduced to 11 ETU when T=1 protocol is used and Guardtime value N = 255.</p> <p>The delay between the leading edge of the last received character and the leading edge of the first character transmitted is 16 ETU when T=0 protocol is used and 22 ETU when T=1 protocol is used.</p>
10	RW	0x0	<p>TS2FIFO TS to FIFO Enables to store the first ATR character TS in RX FIFO. During ideal card session there is no necessity to store TS character, so it can be disabled</p>
9	RW	0x0	<p>RXEN Receiving enable When enabled the characters sent by the Smart Card are received by the UART and stored in RX FIFO. Receiving is internally disabled while a transmission is in progress.</p>
8	RW	0x0	<p>TXEN Transmission enable When enabled the characters are read from TX FIFO and transmitted through UART to the Smart Card</p>
7	RW	0x0	<p>CLKSTOPVAL Clock Stop Value The value of the scclk output during the clock stop state.</p>

Bit	Attr	Reset Value	Description
6	RW	0x0	<p>CLKSTOP Clock Stop</p> <p>Clock Stop. When this bit is asserted and the smart card I/O line is in 'Z' state, the SCR core stops driving of the smart card clock signal after the CLKSTOPDELAY time expires. The smart card clock is restarted immediately after the CLKSTOP signal is deasserted. New character transmission can be started by superior system after the CLKSTARTDELAY time expires. The expiration of both times is signaled by the CLKSTOPRUN bit in the Interrupt registers. Reading '1' from this bit signals that the clock is stopped or CLKSTARTDELAY time not expired yet. Reading '0' from this bit signals that the clock is not stopped.</p>
5:3	RO	0x0	reserved
2	RW	0x0	<p>PECH2FIFO Character With Wrong Parity to FIFO</p> <p>Enables storage of the characters received with wrong parity in RX FIFO.</p>
1	RW	0x0	<p>INVORD Inverse Bit Ordering</p> <p>When High, inverse bit ordering convention(MSB-LSB) is used.</p>
0	RW	0x0	<p>INVLEV Inverse Bit Level</p> <p>When high, inverse level convention is used(A= '1', Z='0');</p>

SCR_CTRL2

Address: Operational Base + offset (0x0004)

Control Register 2

Bit	Attr	Reset Value	Description
15:8	RO	0x00	Reserved3 Reserved Reserved bits are hard-wired to zero
7	RW	0x0	VCC50 Control 5V Smart Card Vcc Control 5V Smart Card Vcc. Setting of this bit allows selection of 5V Vcc for Smart Card session (Class A). After the selection of operating class is completed, this bit is in '1'.if this class was selected. Default value after reset is '0'..
6	RW	0x0	VCC33 Control 3V Smart Card Vcc Setting of this bit allows selection of 3V Vcc for Smart Card session (Class B). After the selection of operating class is completed, this bit is in '1'.if this class was selected. Default value after reset is '0'.
5	RW	0x0	VCC18 Control 1.8V Smart Card Vcc Control 1.8V Smart Card Vcc. Setting of this bit allows selection of 1.8V Vcc for Smart Card session (Class C). After the selection of operating class is completed, this bit is in '1'.if this class was selected. Default value after reset is '0'..
4	RW	0x0	DEACT Deactivation Setting of this bit initializes the deactivation sequence. When the deactivation is finished, the DEACT bit is automatically cleared.
3	RW	0x0	ACT Activation Setting of this bit initializes the activation sequence. When the activation is finished, the ACT bit is automatically cleared.

Bit	Attr	Reset Value	Description
2	WO	0x0	<p>WARMRST Warm Reset Command Writing '1'.to this bit initializes Warm Reset of the Smart Card. This bit is always read as '0'..</p>
1:0	RO	0x0	reserved

SCR SCPADS

Address: Operational Base + offset (0x0008)

Smart Card Pads Register

Bit	Attr	Reset Value	Description
15:10	RO	0x0	reserved
9	RO	0x0	<p>SCPPRESENT Smart Card presented This bit is set to '1'.when the SCDETECT input is active at least for SCDETECTTIME</p>
8	RW	0x0	<p>DSCFCB Direct Smart Card Function Code Bit It provides direct access to SCFCB output</p>
7	RW	0x0	<p>DSCVPPPP Direct Smart Card Vpp Pause/Prog It provides direct access to SCVPPPPP output</p>
6	RW	0x0	<p>DSCVPSEN Direct Smart Card Vpp Enable It provides direct access to SCVPSEN output</p>

Bit	Attr	Reset Value	Description
5	RW	0x0	AUTOADEAVPP Automatic Vpp Handling. When high, it enables automatic handling of DSCVPSEN and DSCVPPPP signals during activation and deactivation sequence.
4	RW	0x0	DSCVCC Direct Smart Card Vcc Direct Smart Card Vcc. When DIRACCPADS = '1'., the DSCVCC bit provides direct access to SCVCCx outputs. The appropriate SCVCC18, SCVCC33 and SCVCC50 outputs are driven according to state of bits VCC18, VCC33 and VCC50 in CTRL2 register.
3	RW	0x0	DSCRST Direct Smart Card Reset When DIRACCPADS = '1'., the DSCRST bit provides direct access to SCRST output
2	RW	0x0	DSCCLK Direct Smart Card Clock When DIRACCPADS = '1'., the DSCCLK bit provides direct access to SCCLK output
1	RW	0x0	DSCIO Direct Smart Card Input/Output When DIRACCPADS = '1'., the DSCIO bit provides direct access to SCIO pad.
0	RW	0x0	DIRACCPADS Direct Access To Smart Card Pads When high, it disables a serial interface functionality and enables direct control of the smart card pads using following 4 bits.

SCR_INTEN1

Address: Operational Base + offset (0x000c)

Interrupt Enable Register 1

Bit	Attr	Reset Value	Description
15	RW	0x0	SCDEACT Smart Card Deactivation Interrupt When enabled, this interrupt is asserted after the Smart Card deactivation sequence is complete.
14	RW	0x0	SCACT Smart Card Activation Interrupt. When enabled, this interrupt is asserted after the Smart Card activation sequence is complete.
13	RW	0x0	SCINS Smart Card Inserted Interrupt When enabled, this interrupt is asserted after the smart card insertion
12	RW	0x0	SCREM Smart Card Removed Interrupt. When enabled, this interrupt is asserted after the smart card removal.
11	RW	0x0	ATRDONE ATR Done Interrupt When enabled, this interrupt is asserted after the ATR sequence is successfully completed.
10	RW	0x0	ATRFAIL ATR Fail Interrupt When enabled, this interrupt is asserted if the ATR sequence fails.
9	RW	0x0	RXTHRESHOLD RX FIFO Threshold Interrupt When enabled, this interrupt is asserted if the number of bytes in RX FIFO is equal or exceeds the RX FIFO threshold.

Bit	Attr	Reset Value	Description
8	RW	0x0	C2CFULL Two Consecutive Characters Limit Interrupt When enabled, this interrupt is asserted if the time between two consecutive characters, transmitted between the Smart Card and the Reader in both directions, is equal the Two Characters Delay Limit described below. The C2CFULL interrupt is internally enabled from the ATR start to the deactivation or ATR restart initialization. It is recommended to use this counter to detect unresponsive Smart Cards.
7	RW	0x0	RXPERR Reception Parity Error Interrupt When enabled, this interrupt is asserted after the character with wrong parity was received when the number of repeated receptions exceeds RXREPEAT value or T=1 protocol is used
6	RW	0x0	TXPERR Transmission Parity Error Interrupt. When enabled, this interrupt is asserted if the Smart Card signals wrong character parity during the guardtime after the character transmission was repeated TXREPEAT-times
5	RW	0x0	RXDONE Reception Done Interrupt When enabled, this interrupt is asserted after a character was received from the Smart Card.
4	RW	0x0	TXDONE Transmission Done Interrupt When enabled, this interrupt is asserted after one character was transmitted to the Smart Card.

Bit	Attr	Reset Value	Description
3	RW	0x0	<p>CLKSTOPRUN Smart Card Clock Stop Interrupt When enabled, this interrupt is asserted in two cases:</p> <ol style="list-style-type: none"> 1. When the smart card clock is stopped (after CLOCKSTOP assertion). 2. When the new character transfer can be started (the smart card clock is fully running after CLOCKSTOP de-assertion).
2	RW	0x0	<p>RXFIFULL RX FIFO Full Interrupt When enabled, this interrupt is asserted if the RX FIFO is filled up.</p>
1	RW	0x0	<p>TXFIEMPTY TX FIFO Empty Interrupt. When enabled, this interrupt is asserted if the TX FIFO is emptied out.</p>
0	RW	0x0	<p>TXFIDONE TX FIFO Done Interrupt When enabled, this interrupt is asserted after all bytes from TX FIFO were transferred to the Smart Card</p>

SCR_INTSTAT1

Address: Operational Base + offset (0x0010)

Interrupt Status Register 1

Bit	Attr	Reset Value	Description
15	RW	0x0	<p>SCDEACT Smart Card Deactivation Interrupt When enabled, this interrupt is asserted after the Smart Card deactivation sequence is complete.</p>

Bit	Attr	Reset Value	Description
14	RW	0x0	SCACT Smart Card Activation Interrupt. When enabled, this interrupt is asserted after the Smart Card activation sequence is complete.
13	RW	0x0	SCINS Smart Card Inserted Interrupt When enabled, this interrupt is asserted after the smart card insertion
12	RW	0x0	SCREM Smart Card Removed Interrupt. When enabled, this interrupt is asserted after the smart card removal.
11	RW	0x0	ATRDONE ATR Done Interrupt When enabled, this interrupt is asserted after the ATR sequence is successfully completed.
10	RW	0x0	ATRFAIL ATR Fail Interrupt When enabled, this interrupt is asserted if the ATR sequence fails.
9	RW	0x0	RXTHRESHOLD RX FIFO Threshold Interrupt When enabled, this interrupt is asserted if the number of bytes in RX FIFO is equal or exceeds the RX FIFO threshold.

Bit	Attr	Reset Value	Description
8	RW	0x0	C2CFULL Two Consecutive Characters Limit Interrupt When enabled, this interrupt is asserted if the time between two consecutive characters, transmitted between the Smart Card and the Reader in both directions, is equal the Two Characters Delay Limit described below. The C2CFULL interrupt is internally enabled from the ATR start to the deactivation or ATR restart initialization. It is recommended to use this counter to detect unresponsive Smart Cards.
7	RW	0x0	RXPERR Reception Parity Error Interrupt When enabled, this interrupt is asserted after the character with wrong parity was received when the number of repeated receptions exceeds RXREPEAT value or T=1 protocol is used
6	RW	0x0	TXPERR Transmission Parity Error Interrupt. When enabled, this interrupt is asserted if the Smart Card signals wrong character parity during the guardtime after the character transmission was repeated TXREPEAT-times
5	RW	0x0	RXDONE Reception Done Interrupt When enabled, this interrupt is asserted after a character was received from the Smart Card.
4	RW	0x0	TXDONE Transmission Done Interrupt When enabled, this interrupt is asserted after one character was transmitted to the Smart Card.

Bit	Attr	Reset Value	Description
3	RW	0x0	<p>CLKSTOPRUN Smart Card Clock Stop Interrupt When enabled, this interrupt is asserted in two cases:</p> <ol style="list-style-type: none"> 1. When the smart card clock is stopped (after CLOCKSTOP assertion). 2. When the new character transfer can be started (the smart card clock is fully running after CLOCKSTOP de-assertion).
2	RW	0x0	<p>RXFIFULL RX FIFO Full Interrupt When enabled, this interrupt is asserted if the RX FIFO is filled up.</p>
1	RW	0x0	<p>TXFIEMPTY TX FIFO Empty Interrupt. When enabled, this interrupt is asserted if the TX FIFO is emptied out.</p>
0	RW	0x0	<p>TXFIDONE TX FIFO Done Interrupt When enabled, this interrupt is asserted after all bytes from TX FIFO were transferred to the Smart Card</p>

SCR_FIFOCTRL

Address: Operational Base + offset (0x0014)

FIFO Control Register

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	WO	0x0	RXFIFLUSH Flush RX FIFO RX FIFO is flushed, when '1'.is written to this bit.
9	RO	0x0	RXFIFULL RX FIFO Full RX FIFO Full
8	RO	0x0	RXFIEMPTY RX FIFO Empty Field0000 Description
7:3	RO	0x0	reserved
2	WO	0x0	TXFIFLUSH Flush TX FIFO. TX FIFO is flushed, when '1'.is written to this bit.
1	RO	0x0	TXFIFULL TX FIFO Full TX FIFO Full
0	RO	0x0	TXFIEMPTY TX FIFO Empty. TX FIFO Empty.

SCR_LEGTXFICNT

Address: Operational Base + offset (0x0018)

Legacy TX FIFO Counter

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
7:0	RO	0x00	<p>LEGTXFICNT Legacy TX FIFO Counter</p> <p>It is equal to TX FIFO Counter up to value 255. All values above 255 are read as 255. It is recommended to use the 16-bit TX FIFO Counter instead of this register.</p>

SCR_LEG_RXFICNT

Address: Operational Base + offset (0x0019)

Legacy RX FIFO Counter

Bit	Attr	Reset Value	Description
7:0	RO	0x00	<p>LEGRXFICNT Legacy RX FIFO Counter</p> <p>It is equal to RX FIFO Counter up to value 255. All values above 255 are read as 255. It is recommended to use the 16-bit RX FIFO Counter instead of this register.</p>

SCR_RXFITH

Address: Operational Base + offset (0x001c)

RX FIFO Threshold

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>RXFITH RX FIFO Threshold</p> <p>The interrupt is asserted when the number of bytes it receives is equal to, or exceeds the threshold</p>

SCR_REP

Address: Operational Base + offset (0x0020)

Repeat

Bit	Attr	Reset Value	Description
7:4	RW	0x0	<p>RXREP RX Repeat</p> <p>This is a 4-bit, read/write register that specifies the number of attempts to request character re-transmission after wrong parity was detected. The re-transmission of the character is requested using the 1 ETU long error signal during the guardtime</p>
3:0	RW	0x0	<p>TXREP TX Repeat</p> <p>This is a 4-bit, read/write register that specifies the number of attempts to re-transmit the character after the Smart Card signals the wrong parity during the guardtime.</p>

SCR_SCCDDIV

Address: Operational Base + offset (0x0024)

Smart Card Clock Divisor

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>SCCDDIV Smart Card Clock Divisor</p> <p>This is a 16-bit, read/write register that defines the divisor value used to generate the Smart Card Clock from the system clock.</p>

SCR_BAUDDIV

Address: Operational Base + offset (0x0028)

Baud Clock Divisor

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>BAUDDIV Baud Clock Divisor</p> <p>This is a 16-bit, read/write register that defines a divisor value used to generate the Baud Clock impulses from the system clock</p>

SCR_SCGUTIME

Address: Operational Base + offset (0x002c)

Smart Card Guardtime

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>SCGUTI Smart Card Guardtime</p> <p>This is an 8-bit, read/write register that sets a delay at the end of each character transmitted from the Smart Card Reader to the Smart Card. The value is in Elementary Time Units (ETU). The parity error is besides signaled during the guardtime</p>

SCR_ADEATIME

Address: Operational Base + offset (0x0030)

Activation / Deactivation Time

Bit	Attr	Reset Value	Description
15:8	RW	0x00	<p>ADEATIME Activation / Deactivation Time</p> <p>Sets the duration of each part of the activation and deactivation sequence. The value is in Smart Card Clock Cycles.</p>

Bit	Attr	Reset Value	Description
7:0	RW	0x00	Reserved Reserved Reserved bits are hard-wired to zero.

SCR_LOWRSTTIME

Address: Operational Base + offset (0x0034)

Reset Duration

Bit	Attr	Reset Value	Description
15:8	RW	0x00	LOWRSTTIME Reset Duration Sets the duration of the smart card reset sequence. This value is same for the cold and warm reset. The value is in terms of smart card clock cycles.
7:0	RW	0x00	Reserved Reserved Bits (7:0) of this register are hard-wired to zero.

SCR_ATRSTARTLIMIT

Address: Operational Base + offset (0x0038)

ATR Start Limit

Bit	Attr	Reset Value	Description
15:8	RW	0x00	ATRSTARTLIMIT ATR Start Limit Defines the maximum time between the rising edge of the SCRSTN signal and the start of ATR response. The value is in terms of smart card clock cycles

Bit	Attr	Reset Value	Description
7:0	RW	0x00	Reserved Reserved Bits (7:0) of this register are hard-wired to zero

SCR_C2CLIM

Address: Operational Base + offset (0x003c)

Two Characters Delay Limit

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	C2CLIM Two Characters Delay Limit This is a 16-bit, read/write register that sets the maximum time between the leading edges of two, consecutive characters. The value is in ETUs.

SCR_INTEN2

Address: Operational Base + offset (0x0040)

Interrupt Enable Register 2

Bit	Attr	Reset Value	Description
15:2	RO	0x0	reserved
1	RW	0x0	TCKERR TCK Error Interrupt. When enabled, this interrupt is asserted if the TCK byte does not match computed value.

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>TXTHRESHOLD TX FIFO Threshold Interrupt</p> <p>When enabled, this interrupt is asserted if the number of bytes in TX FIFO is equal or less than the TX FIFO threshold.</p>

SCR_INTSTAT2

Address: Operational Base + offset (0x0044)

Interrupt Status Register 2

Bit	Attr	Reset Value	Description
15:2	RO	0x0	reserved
1	RW	0x0	<p>TCKERR TCK Error Interrupt</p> <p>When enabled, this interrupt is asserted if the TCK byte does not match computed value.</p>
0	RW	0x0	<p>TXTHRESHOLD TX FIFO Threshold Interrupt</p> <p>When enabled, this interrupt is asserted if the number of bytes in TX FIFO is equal or less than the TX FIFO threshold.</p>

SCR_TXFITH

Address: Operational Base + offset (0x0048)

TX FIFO Threshold

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>TXFITH TX FIFO Threshold</p> <p>The interrupt is asserted when the number of bytes in TX FIFO is equal or less than the threshold</p>

SCR_TXFIFOCNT

Address: Operational Base + offset (0x004c)

TX FIFO Counter

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	<p>TXFIFOCNT TX FIFO Counter</p> <p>This is a 16-bit, read-only register that provides the number of bytes stored in the RX FIFO</p>

SCR_RXFIFOCNT

Address: Operational Base + offset (0x0050)

RX FIFO Counter

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	<p>RXFIFOCNT RX FIFO Counter</p> <p>This is a 16-bit, read-only register that provides the number of bytes stored in the RX FIFO.</p>

SCR_BAUTUNE

Address: Operational Base + offset (0x0054)

Baud Tune Register

Bit	Attr	Reset Value	Description
7:4	RO	0x0	reserved
3:0	RW	0x0	BAUDTUNE Baud Tune Register This is a 3-bit, read/write register that defines an additional value used to increase the accuracy of the Baud Clock impulses

SCR_FIFO

Address: Operational Base + offset (0x0200)

FIFO

Bit	Attr	Reset Value	Description
7:0	RW	0x00	FIFO FIFO This is an 8-bit, read/write register that provides access to the receive and transmit FIFO buffers. The TX FIFO is accessed during the APB write transfer. The RX FIFO is accessed during the APB read transfer. All read/write accesses at address range 200h-3ffh are redirected to the FIFO.

43.5 Interface Description

Table 43-1IOMUX Setting

Module Pin	IO	Pad Name	IOMUX Setting
SIM0			
sim_clk	O	SCclk_URT0sin_GPIO2d3	GPIO2D_IOMUX[7:6]= 2'b01
sim_rstn	O	SCrst_URT0sout_GPIO2d2	GPIO2D_IOMUX[5:4]= 2'b01
sim_data	I/O	SCio_URT0rtsn_GPIO0c1	GPIO0C_IOMUX[2]= 1'b1
sim_detect	O	SCdetect_URT0ctsn_GPIO2d5	GPIO2D_IOMUX[11:10]= 2'b01

Notes: 1. I=Input, O=Output, I/O=input/output, bidirectional

2. Pull up sim0/1 data pin when data transaction

43.6 Application Notes

43.6.1 SCR Clock

The Smart Card Clock signal is used as the main clock for the smart card. Its frequency can be adjusted using the Smart Card Clock Divisor (SCCDIV). This value is used to divide the system clock.

The SCCLK frequency is given by the following equation:

$$\text{SCCLK}_{\text{freq}} = \frac{\text{CLK}_{\text{freq}}}{2 * (\text{SCCDIV} + 1)}, \quad \text{SCCDIV} \cong \frac{\text{CLK}_{\text{freq}}}{2 * \text{SCCLK}_{\text{freq}}} - 1$$

SCCLK_freq- Smart Card Clock Frequency

CLK_freq- System Clock Frequency

The Baud Clock Impulse signal is used to transmit and receive serial data between the Smart Card Reader and the Smart Card. The baud rate can be modified using the Baud Clock Divisor (BAUDDIV) which is used to divide the system clock. The BAUDDIV value must be ≥ 4 . The BAUD rate is given by the following equation:

$$\text{BAUD}_{\text{rate}} = \frac{\text{CLK}_{\text{freq}}}{2 * (\text{BAUDDIV} + 1)}$$

The duration of one bit, Elementary Time Unit (ETU) and parameters F and D are defined in the ISO/IEC7816-3 specification.

$$\frac{1}{\text{BAUD}_{\text{rate}}} \cong \text{ETU} = \frac{F}{D} * \frac{1}{\text{SCCLK}_{\text{freq}}}, \frac{F}{D} \cong \frac{\text{BAUDDIV} + 1}{\text{SCCDIV} + 1}$$

BAUDDIV equation based on SCCDIV value and Smart Card parameters F and D is following:

$$\text{BAUDDIV} \cong (\text{SCCDIV} + 1) * \frac{F}{D} - 1$$

During the first answer to reset response after the cold reset, the initial ETU must be equal to 372 SmartCard Clock Cycles (given by parameters F=372 and D=1). In this case, the BAUDDIV should be:

$$\text{BAUDDIV} \cong (\text{SCCDIV} + 1) * \frac{372}{1} - 1$$

After the ATR is completed, the BAUDDIV register value can be changed according to Smart Card parameters F and D.

Baud Tune Register (BAUDTUNE) 3-bit value that can be used to increase the accuracy of the BaudClock impulses timing by using the BAUDTUNE Increment from Table listed below in combination with BAUDDIVregister value.

Table 43-2BAUDTUNE register

BAUDTUNE	000	001	010	011	100	101	110	111
----------	-----	-----	-----	-----	-----	-----	-----	-----

BAUDTUNE _{INCR}	+0	+0.125	0.25	+0.375	+0.5	+0.625	+0.75	+0.875
--------------------------	----	--------	------	--------	------	--------	-------	--------

$$\text{BAUDDIV} + \text{BAUDTUNE}_{\text{INCR}} \cong (\text{SCCDIV} + 1) * \frac{F}{D} - 1$$

The BAUDDIV register value (nearest integer) can be computed using following equation:

$$\text{BAUDDIV} \cong (\text{SCCDIV} + 1) * \frac{F}{D} - 1 - \text{BAUDTUNE}_{\text{INCR}}$$

Rockchip Confidential

Chapter 44 GMAC Ethernet Interface

44.1 Overview

The GMAC Ethernet Controller provides a complete Ethernet interface from processor to a Reduced Media Independent Interface (RMII) and Reduced Gigabit Media Independent Interface (RGMII) compliant Ethernet PHY.

The GMAC includes a DMA controller. The DMA controller efficiently moves packet data from microprocessor's RAM, formats the data for an IEEE 802.3-2002 compliant packet and transmits the data to an Ethernet Physical Interface (PHY). It also efficiently moves packet data from RXFIFO to microprocessor's RAM.

44.1.1 Features

- Supports 10/100/1000-Mbps data transfer rates with the RGMII interfaces
- Supports 10/100-Mbps data transfer rates with the RMII interfaces
- Supports both full-duplex and half-duplex operation
 - Supports CSMA/CD Protocol for half-duplex operation
 - Supports packet bursting and frame extension in 1000 Mbps half-duplex operation
 - Supports IEEE 802.3x flow control for full-duplex operation
 - Optional forwarding of received pause control frames to the user application in full-duplex operation
 - Back-pressure support for half-duplex operation
 - Automatic transmission of zero-quanta pause frame on deassertion of flow control input in full-duplex operation
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard Ethernet frames
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes:
 - 64-bit Hash filter (optional) for multicast and uni-cast (DA) addresses
 - Option to pass all multicast addressed frames
 - Promiscuous mode support to pass all frames without any filtering for network monitoring
 - Passes all incoming packets (as per filter) with a status report
- Separate 32-bit status returned for transmission and reception packets
- Supports IEEE 802.1Q VLAN tag detection for reception frames
- MDIO Master interface for PHY device configuration and

- management
- Support detection of LAN wake-up frames and AMD Magic Packet frames
- Support checksum off-load for received IPv4 and TCP packets encapsulated by the Ethernet frame
- Support checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams
- Comprehensive status reporting for normal operation and transfers with errors
- Support per-frame Transmit/Receive complete interrupt control
- Supports 4-KB receive FIFO depths on reception.
- Supports 2-KB FIFO depth on transmission
- Automatic generation of PAUSE frame control or backpressure signal to the GMAC core based on Receive FIFO-fill (threshold configurable) level
- Handles automatic retransmission of Collision frames for transmission
- Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- AXI interface to any CPU or memory
- Software can select the type of AXI burst (fixed and variable length burst) in the AXI Master interface
- Supports internal loopback on the RGMII/RMII for debugging
- Debug status register that gives status of FSMs in Transmit and Receive data-paths and FIFO fill-levels.

44.2 Block Diagram

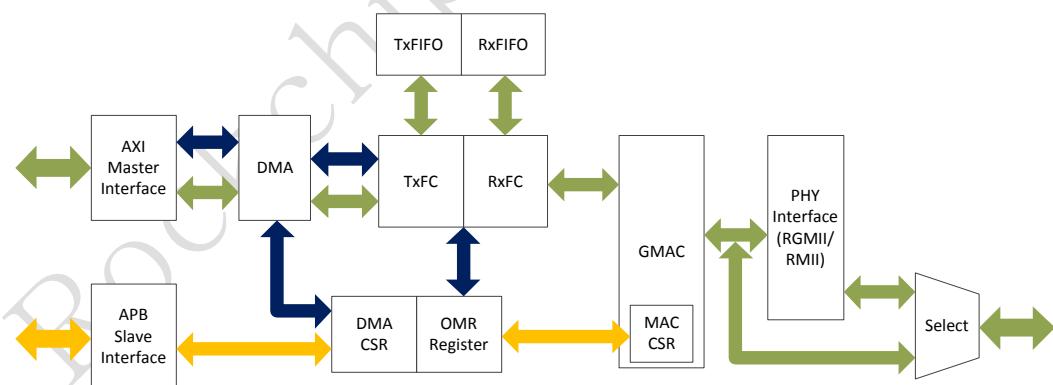


Fig. 44-1 GMAC architecture

The GMAC is broken up into multiple separate functional units. These blocks are interconnected in the MAC module. The block diagram shows the general flow of data and control signals between these blocks.

The GMAC transfers data to system memory through the AXI master interface. The host CPU uses the APB Slave interface to access the GMAC subsystem's control and status registers (CSRs).

The GMAC supports the PHY interfaces of reduced GMII (RGMII) and reduced MII (RMII).

The Transmit FIFO (Tx FIFO) buffers data read from system memory by the DMA before transmission by the GMAC Core. Similarly, the Receive FIFO (Rx FIFO) stores the Ethernet frames received from the line until they are transferred to system memory by the DMA. These are asynchronous FIFOs, as they also transfer the data between the application clock and the GMAC line clocks.

44.3 Function Description

44.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig 32-2.

<inter-frame><preamble><sfd><data><efd>

Fig. 44-2 MAC Frame structure

The preamble <preamble> begins a frame transmission. The bit value of the preamble field consists of 7 octets with the following bit values:

10101010 10101010 10101010 10101010 10101010 10101010 10101010

The SFD (start frame delimiter) <sfd> indicates the start of a frame and follows the preamble. The bit value is 10101011.

The data in a well formed frame shall consist of N octets data.

44.3.2 RMII Interface timing diagram

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorporating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port - a 62.5% decrease in pin count.

The RMII module is instantiated between the GMAC and the PHY. This helps translation of the MAC's MII into the RMII. The RMII block has the following characteristics:

- Supports 10-Mbps and 100-Mbps operating rates. It does not support 1000-Mbps operation.
 - Two clock references are sourced externally or CRU, providing independent, 2-bit wide transmit and receive paths.

Transmit Bit Ordering

Each nibble from the MII must be transmitted on the RMII a di-bit at a time with the order of di-bit transmission shown in Fig.1-3. The lower order bits (D1 and D0) are transmitted first followed by higher order bits (D2 and D3).

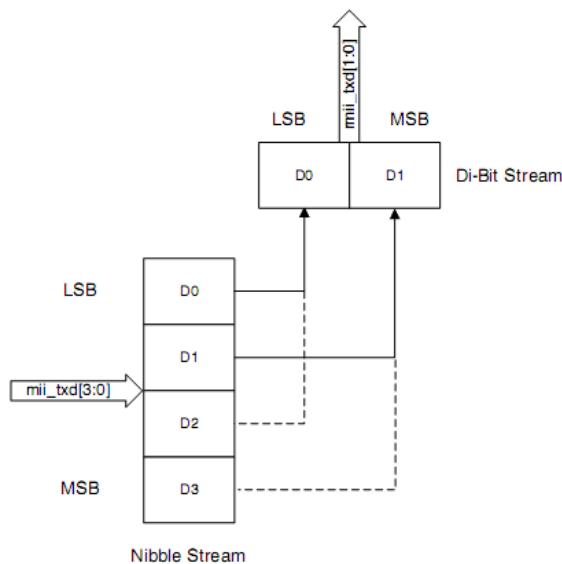


Fig. 44-3 RMII transmission bit ordering

RMII Transmit Timing Diagrams

Fig.1-4 through 1-7 show MII-to-RMII transaction timing. The clk_rmii_i (REF_CLK) frequency is 50MHz in RMII interface. In 10Mb/s mode, as the REF_CLK frequency is 10 times as the data rate, the value on rmii_txd_o[1:0] (TXD[1:0]) shall be valid such that TXD[1:0] may be sampled every 10th cycle, regard-less of the starting cycle within the gRup and yield the correct frame data.

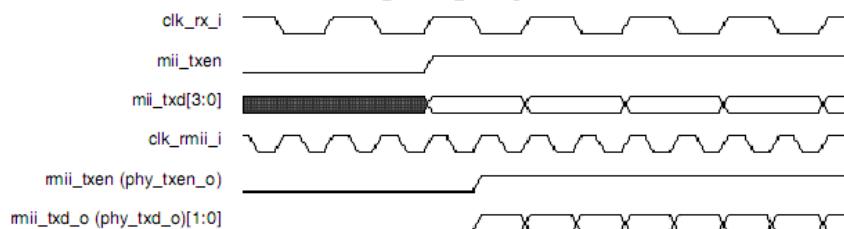


Fig. 44-4 Start of MII and RMII transmission in 100-Mbps mode

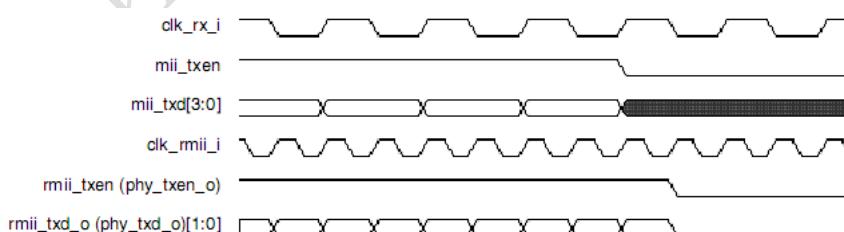


Fig. 44-5 End of MII and RMII Transmission in 100-Mbps Mode

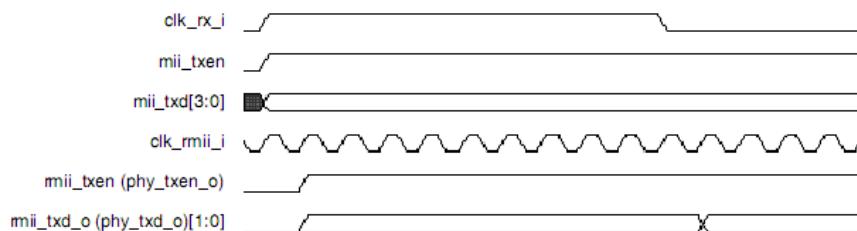


Fig. 44-6 Start of MII and RMII Transmission in 10-Mbps Mode

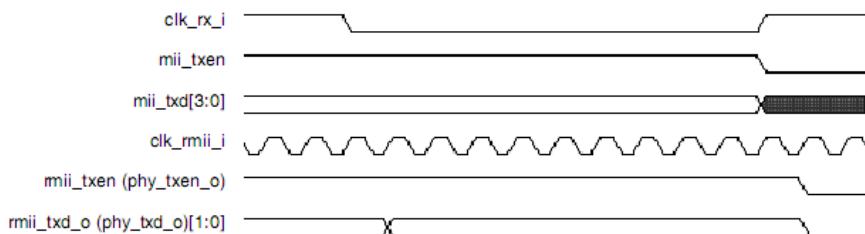


Fig. 44-7 End of MII and RMII Transmission in 10-Mbps Mode

Receive Bit Ordering

Each nibble is transmitted to the MII from the di-bit received from the RMII in the nibble transmission order shown in Fig. 1-8. The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

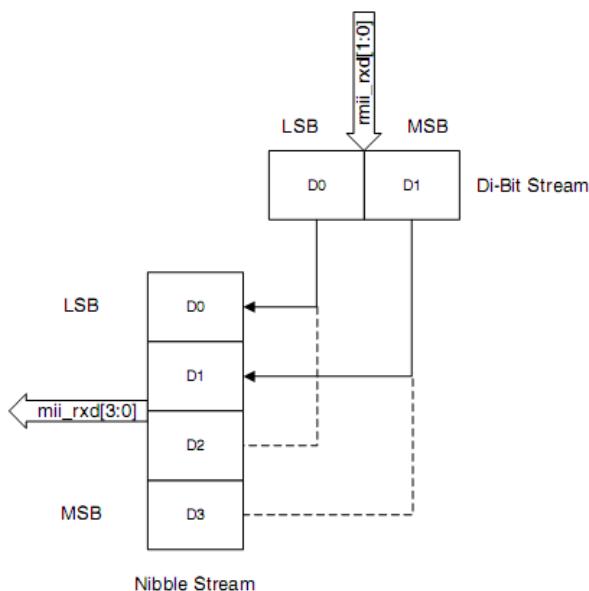


Fig. 44-8 RMII receive bit ordering

44.3.3 RGMII interface

The Reduced Gigabit Media Independent Interface (RGMII) specification reduces the pin count of the interconnection between the GMAC 10/100/1000 controller and the PHY for GMII and MII interfaces. To achieve this, the data path and control signals are reduced and multiplexed together with both the edges of the transmit and receive clocks. For gigabit operation the clocks operate at 125 MHz; for 10/100 operation, the clock rates are 2.5 MHz/25 MHz.

In the GMAC 10/100/1000 controller, the RGMII module is instantiated between

the GMAC core's GMII and the PHY to translate the control and data signals between the GMII and RGMII protocols.

The RGMII block has the following characteristics:

- Supports 10-Mbps, 100-Mbps, and 1000-Mbps operation rates.
- For the RGMII block, no extra clock is required because both the edges of the incoming clocks are used.
- The RGMII block extracts the in-band (link speed, duplex mode and link status) status signals from the PHY and provides them to the GMAC core logic for link detection.

44.3.4 Management Interface

The MAC management interface provides a simple, two-wire, serial interface to connect the GMAC and a managed PHY, for the purposes of controlling the PHY and gathering status from the PHY. The management interface consists of a pair of signals that transport the management information across the MII bus: MDIO and MDC.

The GMAC initiates the management write/read operation. The clock gmii_mdc_o(MDC) is a divided clock from the application clock pclk_gmac. The divide factor depends on the clock range setting in the GMII address register. Clock range is set as follows:

Selection	pclk_gmac	MDC Clock
0000	60-100 MHz	pclk_gmac/42
0001	100-150 MHz	pclk_gmac/62
0010	20-35 MHz	pclk_gmac/16
0011	35-60 MHz	pclk_gmac/26
0100	150-250 MHz	pclk_gmac/102
0101	250-300 MHz	pclk_gmac/124
0110, 0111	Reserved	

The MDC is the derivative of the application clock pclk_gmac. The management operation is performed through the gmii_mdi_i, gmii_mdo_o and gmii_mdo_o_e signals. A three-state buffer is implemented in the PAD.

The frame structure on the MDIO line is shown below.

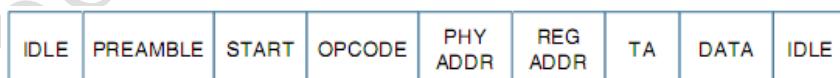


Fig. 44-9 MDIO frame structure

- IDLE: The mdio line is three-state; there is no clock on gmii_mdc_o
- PREAMBLE: 32 continuous bits of value 1
- START: Start-of-frame is 2'01
- OPCODE: 2'b10 for read and 2'b01 for write
- PHY ADDR: 5-bit address select for one of 32 PHYs
- REG ADDR: Register address in the selected PHY
- TA: Turnaround is 2'bZ0 for read and 2'b10 for Write
- DATA: Any 16-bit value. In a write operation, the GMAC drives mdio; in a read operation, PHY drives it.

44.3.5 Power Management Block

Power management(PMT) supports the reception of network (remote) wake-up frames and Magic Packet frames. PMT does not perform the clock gate function, but generates interrupts for wake-up frames and Magic Packets received by the GMAC. The PMT block sits on the receiver path of the GMAC and is enabled with remote wake-up frame enable and Magic Packet enable. These enables are in the PMT control and status register and are programmed by the application.

When the power down mode is enabled in the PMT, then all received frames are dropped by the core and they are not forwarded to the application. The core comes out of the power down mode only when either a Magic Packet or a Remote Wake-up frame is received and the corresponding detection is enabled.

Remote Wake-Up Frame Detection

When the GMAC is in sleep mode and the remote wake-up bit is enabled in register GMAC_PMT_CTRL_STA (0x002C), normal operation is resumed after receiving a remote wake-up frame. The application writes all eight wake-up filter registers, by performing a sequential write to address (0028). The application enables remote wake-up by writing a 1 to bit 2 of the register GMAC_PMT_CTRL_STA.

PMT supports four programmable filters that allow support of different receive frame patterns. If the incoming frame passes the address filtering of Filter Command, and if Filter CRC-16 matches the incoming examined pattern, then the wake-up frame is received.

Filter_offset (minimum value 12, which refers to the 13th byte of the frame) determines the offset from which the frame is to be examined. Filter Byte Mask determines which bytes of the frame must be examined. The thirty-first bit of Byte Mask must be set to zero.

The remote wake-up CRC block determines the CRC value that is compared with Filter CRC-16. The wake-up frame is checked only for length error, FCS error, dribble bit error, GMII error, collision, and to ensure that it is not a runt frame. Even if the wake-up frame is more than 512 bytes long, if the frame has a valid CRC value, it is considered valid. Wake-up frame detection is updated in the register GMAC_PMT_CTRL_STA for every remote Wake-up frame received. A PMT interrupt to the application triggers a read to the GMAC_PMT_CTRL_STA register to determine reception of a wake-up frame.

Magic Packet Detection

The Magic Packet frame is based on a method that uses Advanced Micro Device's Magic Packet technology to power up the sleeping device on the network. The GMAC receives a specific packet of information, called a Magic Packet, addressed to the node on the network.

Only Magic Packets that are addressed to the device or a broadcast address will be checked to determine whether they meet the wake-up requirements. Magic Packets that pass the address filtering (unicast or broadcast) will be checked to determine whether they meet the remote Wake-on-LAN data format of 6 bytes of all ones followed by a GMAC Address appearing 16 times.

The application enables Magic Packet wake-up by writing a 1 to Bit 1 of the register GMAC_PMT_CTRL_STA. The PMT block constantly monitors each frame addressed to the node for a specific Magic Packet pattern. Each frame received is checked for a 48'hFF_FF_FF_FF_FF_FF pattern following the destination and

source address field. The PMT block then checks the frame for 16 repetitions of the GMAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the 48'hFF_FF_FF_FF_FF_FF pattern is scanned for again in the incoming frame. The 16 repetitions can be anywhere in the frame, but must be preceded by the synchronization stream (48'hFF_FF_FF_FF_FF_FF). The device will also accept a multicast frame, as long as the 16 duplications of the GMAC address are detected.

If the MAC address of a node is 48'h00_11_22_33_44_55, then the GMAC scans for the data sequence:

```
Destination Address Source Address ..... FF FF FF FF FF FF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
...CRC
```

Magic Packet detection is updated in the PMT Control and Status register for Magic Packet received. A PMT interrupt to the Application triggers a read to the PMT CSR to determine whether a Magic Packet frame has been received.

44.3.6 MAC Management Counters

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. These include a control register for controlling the behavior of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Non-32-bit accesses are allowed as long as the address is word-aligned.

The organization of these registers is shown in Register Description. The MMCs are accessed using transactions, in the same way the CSR address space is accessed. The Register Description in this chapter describe the various counters and list the address for each of the statistics counters. This address will be used for Read/Write accesses to the desired transmit/receive counter.

The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet frames.

44.4 Register description

44.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
GMAC_MAC_CONF	0x0000	W	0x00000000	MAC Configuration Register
GMAC_MAC_FRM_FILT	0x0004	W	0x00000000	MAC Frame Filter
GMAC_HASH_TAB_HI	0x0008	W	0x00000000	Hash Table High Register
GMAC_HASH_TAB_LO	0x000c	W	0x00000000	Hash Table Low Register
GMAC_GMII_ADDR	0x0010	W	0x00000000	GMII Address Register

Name	Offset	Size	Reset Value	Description
GMAC_GMII_DATA	0x0014	W	0x00000000	GMII Data Register
GMAC_FLOW_CTRL	0x0018	W	0x00000000	Flow Control Register
GMAC_VLAN_TAG	0x001c	W	0x00000000	VLAN Tag Register
GMAC_DEBUG	0x0024	W	0x00000000	Debug register
GMAC_PMT_CTRL_STA	0x002c	W	0x00000000	PMT Control and Status Register
GMAC_INT_STATUS	0x0038	W	0x00000000	Interrupt Status Register
GMAC_INT_MASK	0x003c	W	0x00000000	Interrupt Mask Register
GMAC_MAC_ADDR0_HI	0x0040	W	0x0000ffff	MAC Address0 High Register
GMAC_MAC_ADDR0_LO	0x0044	W	0xffffffff	MAC Address0 Low Register
GMAC_AN_CTRL	0x00c0	W	0x00000000	AN Control Register
GMAC_AN_STATUS	0x00c4	W	0x00000008	AN Status Register
GMAC_AN_ADV	0x00c8	W	0x000001e0	Auto_Negotiation Advertisement Register
GMAC_AN_LINK_PART_AB	0x00cc	W	0x00000000	Auto_Negotiation Link Partner Ability Register
GMAC_AN_EXP	0x00d0	W	0x00000000	Auto_Negotiation Expansion Register
GMAC_INTF_MODE_STA	0x00d8	W	0x00000000	RGMII Status Register
GMAC_MMC_CTRL	0x0100	W	0x00000000	MMC Control Register
GMAC_MMC_RX_INT_R	0x0104	W	0x00000000	MMC Receive Interrupt Register
GMAC_MMC_TX_INT_R	0x0108	W	0x00000000	MMC Transmit Interrupt Register
GMAC_MMC_RX_INT_MSK	0x010c	W	0x00000000	MMC Receive Interrupt Mask Register
GMAC_MMC_TX_INT_MSK	0x0110	W	0x00000000	MMC Transmit Interrupt Mask Register
GMAC_MMC_TXOCTET_TCNT_GB	0x0114	W	0x00000000	MMC TX OCTET Good and Bad Counter
GMAC_MMC_TXFRMCNT_GB	0x0118	W	0x00000000	MMC TX Frame Good and Bad Counter
GMAC_MMC_TXUNDFLWERR	0x0148	W	0x00000000	MMC TX Underflow Error
GMAC_MMC_TXCARERR	0x0160	W	0x00000000	MMC TX Carrier Error
GMAC_MMC_TXOCTET_TCNT_G	0x0164	W	0x00000000	MMC TX OCTET Good Counter
GMAC_MMC_TXFRMCNT_G	0x0168	W	0x00000000	MMC TX Frame Good Counter
GMAC_MMC_RXFRMCNT_GB	0x0180	W	0x00000000	MMC RX Frame Good and Bad Counter
GMAC_MMC_RXOCTET_TCNT_GB	0x0184	W	0x00000000	MMC RX OCTET Good and Bad Counter
GMAC_MMC_RXOCTET_TCNT_G	0x0188	W	0x00000000	MMC RX OCTET Good Counter

Name	Offset	Size	Reset Value	Description
GMAC_MMCRXMCFRMCNT_G	0x0190	W	0x00000000	MMC RX Multicast Frame Good Counter
GMAC_MMCRXCRCERRR	0x0194	W	0x00000000	MMC RX Carrier
GMAC_MMCRXLENERRR	0x01c8	W	0x00000000	MMC RX Length Error
GMAC_MMCRXFIFO_OVRFLW	0x01d4	W	0x00000000	MMC RX FIFO Overflow
GMAC_MMCIIPCINT_MSK	0x0200	W	0x00000000	MMC Receive Checksum Offload Interrupt Mask Register
GMAC_MMCIIPCINTR	0x0208	W	0x00000000	MMC Receive Checksum Offload Interrupt Register
GMAC_MMCRXIPV4_GFRM	0x0210	W	0x00000000	MMC RX IPV4 Good Frame
GMAC_MMCRXIPV4_HDERRFRM	0x0214	W	0x00000000	MMC RX IPV4 Head Error Frame
GMAC_MMCRXIPV6_GFRM	0x0224	W	0x00000000	MMC RX IPV6 Good Frame
GMAC_MMCRXIPV6_HDERRFRM	0x0228	W	0x00000000	MMC RX IPV6 Head Error Frame
GMAC_MMCRXUDPERRFRM	0x0234	W	0x00000000	MMC RX UDP Error Frame
GMAC_MMCRXTCPERRFRM	0x023c	W	0x00000000	MMC RX TCP Error Frame
GMAC_MMCRXICMPERRFRM	0x0244	W	0x00000000	MMC RX ICMP Error Frame
GMAC_MMCRXIPV4_HDERROCT	0x0254	W	0x00000000	MMC RX OCTET IPV4 Head Error
GMAC_MMCRXIPV6_HDERROCT	0x0268	W	0x00000000	MMC RX OCTET IPV6 Head Error
GMAC_MMCRXUDPERROCT	0x0274	W	0x00000000	MMC RX OCTET UDP Error
GMAC_MMCRXTCPERRROCT	0x027c	W	0x00000000	MMC RX OCTET TCP Error
GMAC_MMCRXICMPERRROCT	0x0284	W	0x00000000	MMC RX OCTET ICMP Error
GMAC_BUS_MODE	0x1000	W	0x00020101	Bus Mode Register
GMAC_TX_POLL_DEMAND	0x1004	W	0x00000000	Transmit Poll Demand Register
GMAC_RX_POLL_DEMAND	0x1008	W	0x00000000	Receive Poll Demand Register
GMAC_RX_DESC_LIST_ADDR	0x100c	W	0x00000000	Receive Descriptor List Address Register
GMAC_TX_DESC_LIST_ADDR	0x1010	W	0x00000000	Transmit Descriptor List Address Register
GMAC_STATUS	0x1014	W	0x00000000	Status Register
GMAC_OP_MODE	0x1018	W	0x00000000	Operation Mode Register
GMAC_INT_ENA	0x101c	W	0x00000000	Interrupt Enable Register

Name	Offset	Size	Reset Value	Description
GMAC_OVERFLOW_CNT	0x1020	W	0x00000000	Missed Frame and Buffer Overflow Counter Register
GMAC_REC_INT_WDT_TIMER	0x1024	W	0x00000000	Receive Interrupt Watchdog Timer Register
GMAC_AXI_BUS_MODE	0x1028	W	0x00110001	AXI Bus Mode Register
GMAC_AXI_STATUS	0x102c	W	0x00000000	AXI Status Register
GMAC_CUR_HOST_TX_DESC	0x1048	W	0x00000000	Current Host Transmit Descriptor Register
GMAC_CUR_HOST_RX_DESC	0x104c	W	0x00000000	Current Host Receive Descriptor Register
GMAC_CUR_HOST_TX_Buf_ADDR	0x1050	W	0x00000000	Current Host Transmit Buffer Address Register
GMAC_CUR_HOST_RX_BUF_ADDR	0x1054	W	0x00000000	Current Host Receive Buffer Adderss Register
GMAC_HW_FEA_REG	0x1058	W	0x000d0f17	The presence of the optional features/functions of the core

Notes: Size : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** -WORD (32 bits) access

44.4.2 Detail Register Description

GMAC_MAC_CONF

Address: Operational Base + offset (0x0000)

MAC Configuration Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	TC Transmit Configuration in RGMII/SGMII/SMII When set, this bit enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RGMII ports. When this bit is reset, no such information is driven to the PHY.
23	RW	0x0	WD Watchdog Disable When this bit is set, the GMAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the GMAC allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that.

Bit	Attr	Reset Value	Description
22	RW	0x0	JD Jabber Disable When this bit is set, the GMAC disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. When this bit is reset, the GMAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.
21	RW	0x0	BE Frame Burst Enable When this bit is set, the GMAC allows frame bursting during transmission in GMII Half-Duplex mode.
20	RO	0x0	reserved
19:17	RW	0x0	IFG Inter-Frame Gap These bits control the minimum IFG between frames during transmission. 3'b000: 96 bit times 3'b001: 88 bit times 3'b010: 80 bit times ... 3'b111: 40 bit times
16	RW	0x0	DCRS Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions.
15	RW	0x0	PS Port Select Selects between GMII and MII: 1'b0: GMII (1000 Mbps) 1'b1: MII (10/100 Mbps)
14	RW	0x0	FES Speed Indicates the speed in Fast Ethernet (MII) mode: 1'b0: 10 Mbps 1'b1: 100 Mbps

Bit	Attr	Reset Value	Description
13	RW	0x0	<p>DO Disable Receive Own When this bit is set, the GMAC disables the reception of frames when the gmii_txen_o is asserted in Half-Duplex mode. When this bit is reset, the GMAC receives all packets that are given by the PHY while transmitting.</p>
12	RW	0x0	<p>LM Loopback Mode When this bit is set, the GMAC operates in loopback mode at GMII/MII. The (G)MII Receive clock input (clk_rx_i) is required for the loopback to work properly, as the Transmit clock is not looped-back internally.</p>
11	RW	0x0	<p>DM Duplex Mode When this bit is set, the GMAC operates in a Full-Duplex mode where it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in Full-Duplex-only configuration.</p>
10	RW	0x0	<p>IPC Checksum Offload When this bit is set, the GMAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25-26 or 29-30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The GMAC core also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables IPv4 checksum checking for received frame payloads TCP/UDP/ICMP headers. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared.</p>

Bit	Attr	Reset Value	Description
9	RW	0x0	<p>DR Disable Retry When this bit is set, the GMAC will attempt only 1 transmission. When a collision occurs on the GMII/MII, the GMAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the GMAC will attempt retries based on the settings of BL.</p>
8	RW	0x0	<p>LUD Link Up/Down Indicates whether the link is up or down during the transmission of configuration in RGMII interface: 1'b0: Link Down 1'b1: Link Up</p>
7	RW	0x0	<p>ACS Automatic Pad/CRC Stripping When this bit is set, the GMAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the GMAC will pass all incoming frames to the Host unmodified.</p>
6:5	RW	0x0	<p>BL Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the GMAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode and is reserved (RO) in Full-Duplex-only configuration. 2'b00: $k = \min(n, 10)$ 2'b01: $k = \min(n, 8)$ 2'b10: $k = \min(n, 4)$ 2'b11: $k = \min(n, 1)$, where n = retransmission attempt. The random integer r takes the value in the range $0 = r < 2^k$</p>

Bit	Attr	Reset Value	Description
4	RW	0x0	<p>DC Deferral Check When this bit is set, the deferral check function is enabled in the GMAC. The GMAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmit state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Core is configured for 1000 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the GMII/MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts.</p> <p>When this bit is reset, the deferral check function is disabled and the GMAC defers until the CRS signal goes inactive.</p>
3	RW	0x0	<p>TE Transmitter Enable When this bit is set, the transmit state machine of the GMAC is enabled for transmission on the GMII/MII. When this bit is reset, the GMAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames.</p>
2	RW	0x0	<p>RE Receiver Enable When this bit is set, the receiver state machine of the GMAC is enabled for receiving frames from the GMII/MII. When this bit is reset, the GMAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the GMII/MII.</p>
1:0	RO	0x0	reserved

GMAC_MAC_FRM_FILT

Address: Operational Base + offset (0x0004)

MAC Frame Filter

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31	RW	0x0	RA Receive All When this bit is set, the GMAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter.
30:11	RO	0x0	reserved
10	RW	0x0	HPF Hash or Perfect Filter When set, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When low and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter.
9	RW	0x0	SAF Source Address Filter Enable The GMAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the GMAC drops the frame. When this bit is reset, then the GMAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address comparison.
8	RW	0x0	SAIF SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter.

Bit	Attr	Reset Value	Description
7:6	RW	0x0	<p>PCF Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of Register GMAC_FLOW_CTRL[2].</p> <p>2'b00: GMAC filters all control frames from reaching the application.</p> <p>2'b01: GMAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter.</p> <p>2'b10: GMAC forwards all control frames to application even if they fail the Address Filter.</p> <p>2'b11: GMAC forwards control frames that pass the Address Filter.</p>
5	RW	0x0	<p>DBF Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames.</p>
4	RW	0x0	<p>PM Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit.</p>
3	RW	0x0	<p>DAIF DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.</p>
2	RW	0x0	<p>HMC Hash Multicast When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.</p>

Bit	Attr	Reset Value	Description
1	RW	0x0	HUC Hash Unicast When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.
0	RW	0x0	PR Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set.

GMAC_HASH_TAB_HI

Address: Operational Base + offset (0x0008)

Hash Table High Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTH Hash Table High This field contains the upper 32 bits of Hash table

GMAC_HASH_TAB_LO

Address: Operational Base + offset (0x000c)

Hash Table Low Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTL Hash Table Low This field contains the lower 32 bits of Hash table

GMAC_GMII_ADDR

Address: Operational Base + offset (0x0010)

GMII Address Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:11	RW	0x00	PA Physical Layer Address This field tells which of the 32 possible PHY devices are being accessed
10:6	RW	0x00	GR GMII Register These bits select the desired GMII register in the selected PHY device

Bit	Attr	Reset Value	Description																																										
5:2	RW	0x0	<p>CR APB Clock Range The APB Clock Range selection determines the frequency of the MDC clock as per the pclk_gmac frequency used in your design. The suggested range of pclk_gmac frequency applicable for each value below (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz - 2.5 MHz.</p> <table> <thead> <tr> <th>Selection</th> <th>pclk_gmac</th> <th>MDC Clock</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>pclk_gmac/42</td> <td>60-100 MHz</td> </tr> <tr> <td>0001</td> <td>pclk_gmac/62</td> <td>100-150 MHz</td> </tr> <tr> <td>0010</td> <td>pclk_gmac/16</td> <td>20-35 MHz</td> </tr> <tr> <td>0011</td> <td>pclk_gmac/26</td> <td>35-60 MHz</td> </tr> <tr> <td>0100</td> <td>pclk_gmac/102</td> <td>150-250 MHz</td> </tr> <tr> <td>0101</td> <td>pclk_gmac/124</td> <td>250-300 MHz</td> </tr> <tr> <td>0110, 0111</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <p>When bit 5 is set, you can achieve MDC clock of frequency higher than the IEEE 802.3 specified frequency limit of 2.5 MHz and program a clock divider of lower value. For example, when pclk_gmac is of frequency 100 Mhz and you program these bits as "1010", then the resultant MDC clock will be of 12.5 Mhz which is outside the limit of IEEE 802.3 specified range. Please program the values given below only if the interfacing chips supports faster MDC clocks.</p> <table> <thead> <tr> <th>Selection</th> <th>MDC Clock</th> </tr> </thead> <tbody> <tr> <td>1000</td> <td>pclk_gmac/4</td> </tr> <tr> <td>1001</td> <td>pclk_gmac/6</td> </tr> <tr> <td>1010</td> <td>pclk_gmac/8</td> </tr> <tr> <td>1011</td> <td>pclk_gmac/10</td> </tr> <tr> <td>1100</td> <td>pclk_gmac/12</td> </tr> <tr> <td>1101</td> <td>pclk_gmac/14</td> </tr> <tr> <td>1110</td> <td>pclk_gmac/16</td> </tr> <tr> <td>1111</td> <td>pclk_gmac/18</td> </tr> </tbody> </table>	Selection	pclk_gmac	MDC Clock	0000	pclk_gmac/42	60-100 MHz	0001	pclk_gmac/62	100-150 MHz	0010	pclk_gmac/16	20-35 MHz	0011	pclk_gmac/26	35-60 MHz	0100	pclk_gmac/102	150-250 MHz	0101	pclk_gmac/124	250-300 MHz	0110, 0111	Reserved		Selection	MDC Clock	1000	pclk_gmac/4	1001	pclk_gmac/6	1010	pclk_gmac/8	1011	pclk_gmac/10	1100	pclk_gmac/12	1101	pclk_gmac/14	1110	pclk_gmac/16	1111	pclk_gmac/18
Selection	pclk_gmac	MDC Clock																																											
0000	pclk_gmac/42	60-100 MHz																																											
0001	pclk_gmac/62	100-150 MHz																																											
0010	pclk_gmac/16	20-35 MHz																																											
0011	pclk_gmac/26	35-60 MHz																																											
0100	pclk_gmac/102	150-250 MHz																																											
0101	pclk_gmac/124	250-300 MHz																																											
0110, 0111	Reserved																																												
Selection	MDC Clock																																												
1000	pclk_gmac/4																																												
1001	pclk_gmac/6																																												
1010	pclk_gmac/8																																												
1011	pclk_gmac/10																																												
1100	pclk_gmac/12																																												
1101	pclk_gmac/14																																												
1110	pclk_gmac/16																																												
1111	pclk_gmac/18																																												
1	RW	0x0	<p>GW GMII Write When set, this bit tells the PHY that this will be a Write operation using register GMAC_GMII_DATA. If this bit is not set, this will be a Read operation, placing the data in register GMAC_GMII_DATA.</p>																																										

Bit	Attr	Reset Value	Description
0	W1C	0x0	<p>GB GMII Busy This bit should read a logic 0 before writing to Register GMII_ADDR and Register GMII_DATA. This bit must also be set to 0 during a Write to Register GMII_ADDR. During a PHY register access, this bit will be set to 1'b1 by the Application to indicate that a Read or Write access is in progress. Register GMII_DATA (GMII Data) should be kept valid until this bit is cleared by the GMAC during a PHY Write operation. The Register GMII_DATA is invalid until this bit is cleared by the GMAC during a PHY Read operation. The Register GMII_ADDR (GMII Address) should not be written to until this bit is cleared.</p>

GMAC_GMII_DATA

Address: Operational Base + offset (0x0014)

GMII Data Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	<p>GD GMII Data This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation.</p>

GMAC_FLOW_CTRL

Address: Operational Base + offset (0x0018)

Flow Control Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>PT Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain.</p>
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description										
7	RW	0x0	<p>DZPQ Disable Zero-Quanta Pause When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the deassertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i/mti_flowctrl_i). When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled.</p>										
6	RO	0x0	reserved										
5:4	RW	0x0	<p>PLT Pause Low Threshold This field configures the threshold of the PAUSE timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot-times after the first PAUSE frame is transmitted.</p> <table> <tr> <td>Selection</td> <td>Threshold</td> </tr> <tr> <td>00</td> <td>Pause time minus 4 slot times</td> </tr> <tr> <td>01</td> <td>Pause time minus 28 slot times</td> </tr> <tr> <td>10</td> <td>Pause time minus 144 slot times</td> </tr> <tr> <td>11</td> <td>Pause time minus 256 slot times</td> </tr> </table> <p>Slot time is defined as time taken to transmit 512 bits (64 bytes) on the GMII/MII interface.</p>	Selection	Threshold	00	Pause time minus 4 slot times	01	Pause time minus 28 slot times	10	Pause time minus 144 slot times	11	Pause time minus 256 slot times
Selection	Threshold												
00	Pause time minus 4 slot times												
01	Pause time minus 28 slot times												
10	Pause time minus 144 slot times												
11	Pause time minus 256 slot times												
3	RW	0x0	<p>UP Unicast Pause Frame Detect When this bit is set, the GMAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the GMAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard.</p>										
2	RW	0x0	<p>RFE Receive Flow Control Enable When this bit is set, the GMAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled.</p>										

Bit	Attr	Reset Value	Description
1	RW	0x0	<p>TFE Transmit Flow Control Enable In Full-Duplex mode, when this bit is set, the GMAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the GMAC is disabled, and the GMAC will not transmit any Pause frames.</p> <p>In Half-Duplex mode, when this bit is set, the GMAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled.</p>
0	RW	0x0	<p>FCB_BPA Flow Control Busy/Backpressure Activate This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set.</p> <p>In Full-Duplex mode, this bit should be read as 1'b0 before writing to the register GMAC_FLOW_CTRL. To initiate a pause control frame, the application must set this bit to 1'b1. During a transfer of the control frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the GMAC will reset this bit to 1'b0. The register GMAC_FLOW_CTRL should not be written to until this bit is cleared.</p> <p>In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the GMAC Core. During backpressure, when the GMAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically OR'ed with the mti_flowctrl_i input signal for the backpressure function.</p>

GMAC_VLAN_TAG

Address: Operational Base + offset (0x001c)

VLAN Tag Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16	RW	0x0	<p>ETV Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. Bits[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison.</p>
15:0	RW	0x0000	<p>VL VLAN Tag Identifier for Receive Frames This contains the 802.1Q VLAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. Bits[15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field. When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the GMAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames.</p>

GMAC_DEBUG

Address: Operational Base + offset (0x0024)

Debug register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	<p>TFIFO3 When high, it indicates that the MTL TxStatus FIFO is full and hence the MTL will not be accepting any more frames for transmission.</p>
24	RW	0x0	<p>TFIFO2 When high, it indicates that the MTL TxFIFO is not empty and has some data left for transmission.</p>
23	RO	0x0	reserved
22	RW	0x0	<p>TFIFO1 When high, it indicates that the MTL TxFIFO Write Controller is active and transferring data to the TxFIFO.</p>

Bit	Attr	Reset Value	Description
21:20	RW	0x0	TFIFOSTA This indicates the state of the TxFIFO read Controller: 2'b00: IDLE state 2'b01: READ state (transferring data to MAC transmitter) 2'b10: Waiting for TxStatus from MAC transmitter 2'b11: Writing the received TxStatus or flushing the TxFIFO
19	RW	0x0	PAUSE When high, it indicates that the MAC transmitter is in PAUSE condition (in full-duplex only) and hence will not schedule any frame for transmission
18:17	RW	0x0	TSAT This indicates the state of the MAC Transmit Frame Controller module: 2'b00: IDLE 2'b01: Waiting for Status of previous frame or IFG/backoff period to be over 2'b10: Generating and transmitting a PAUSE control frame (in full duplex mode) 2'b11: Transferring input frame for transmission
16	RW	0x0	TACT When high, it indicates that the MAC GMII/MII transmit protocol engine is actively transmitting data and not in IDLE state.
15:10	RO	0x0	reserved
9:8	RW	0x0	RFIFO This gives the status of the RxFIFO Fill-level: 2'b00: RxFIFO Empty 2'b01: RxFIFO fill-level below flow-control de-activate threshold 2'b10: RxFIFO fill-level above flow-control activate threshold 2'b11: RxFIFO Full
7	RO	0x0	reserved
6:5	RW	0x0	RFIFORD It gives the state of the RxFIFO read Controller: 2'b00: IDLE state 2'b01: Reading frame data 2'b10: Reading frame status (or time-stamp) 2'b11: Flushing the frame data and Status
4	RW	0x0	RFIFOWR When high, it indicates that the MTL RxFIFO Write Controller is active and transferring a received frame to the FIFO.
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:1	RW	0x0	ACT When high, it indicates the active state of the small FIFO Read and Write controllers respectively of the MAC receive Frame Controller module
0	RW	0x0	RDB When high, it indicates that the MAC GMII/MII receive protocol engine is actively receiving data and not in IDLE state.

GMAC_PMT_CTRL_STA

Address: Operational Base + offset (0x002c)

PMT Control and Status Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	WFFRPR Wake-Up Frame Filter Register Pointer Reset When set, resets the Remote Wake-up Frame Filter register pointer to 3'b000. It is automatically cleared after 1 clock cycle.
30:10	RO	0x0	reserved
9	RW	0x0	GU Global Unicast When set, enables any unicast packet filtered by the GMAC (DAF) address recognition to be a wake-up frame.
8:7	RO	0x0	reserved
6	RC	0x0	WFR Wake-Up Frame Received When set, this bit indicates the power management event was generated due to reception of a wake-up frame. This bit is cleared by a read into this register.
5	RC	0x0	MPR Magic Packet Received When set, this bit indicates the power management event was generated by the reception of a Magic Packet. This bit is cleared by a read into this register.
4:3	RO	0x0	reserved
2	RW	0x0	WFE Wake-Up Frame Enable When set, enables generation of a power management event due to wake-up frame reception.
1	RW	0x0	MPE Magic Packet Enable When set, enables generation of a power management event due to Magic Packet reception.

Bit	Attr	Reset Value	Description
0	R/WSC	0x0	<p>PD Power Down When set, all received frames will be dropped. This bit is cleared automatically when a magic packet or Wake-Up frame is received, and Power-Down mode is disabled. Frames received after this bit is cleared are forwarded to the application. This bit must only be set when either the Magic Packet Enable or Wake-Up Frame Enable bit is set high.</p>

GMAC_INT_STATUS

Address: Operational Base + offset (0x0038)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	<p>MRCOIS MMC Receive Checksum Offload Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.</p>
6	RO	0x0	<p>MTIS MMC Transmit Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration.</p>
5	RO	0x0	<p>MRIS MMC Receive Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration.</p>
4	RO	0x0	<p>MIS MMC Interrupt Status This bit is set high whenever any of bits 7:5 is set high and cleared only when all of these bits are low. This bit is valid only when the optional MMC module is selected during configuration.</p>

Bit	Attr	Reset Value	Description
3	RO	0x0	PIS PMT Interrupt Status This bit is set whenever a Magic packet or Wake-on-LAN frame is received in Power-Down mode). This bit is cleared when both bits[6:5] are cleared due to a read operation to the register GMAC_PMT_CTRL_STA.
2:1	RO	0x0	reserved
0	RO	0x0	RIS RGMII Interrupt Status This bit is set due to any change in value of the Link Status of RGMII interface. This bit is cleared when the user makes a read operation to the RGMII Status register.

GMAC_INT_MASK

Address: Operational Base + offset (0x003c)

Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	PIM PMT Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of PMT Interrupt Status bit in Register GMAC_INT_STATUS.
2:1	RO	0x0	reserved
0	RW	0x0	RIM RGMII Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of RGMII Interrupt Status bit in Register GMAC_INT_STATUS.

GMAC_MAC_ADDR0_HI

Address: Operational Base + offset (0x0040)

MAC Address0 High Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0xffff	A47_A32 MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC_MAC_ADDR0_LO

Address: Operational Base + offset (0x0044)

MAC Address0 Low Register

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	A31_A0 MAC Address0 [31:0] This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC_AN_CTRL

Address: Operational Base + offset (0x00c0)

AN Control Register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	ANE Auto-Negotiation Enable When set, will enable the GMAC to perform auto-negotiation with the link partner. Clearing this bit will disable auto-negotiation.
11:10	RO	0x0	reserved
9	RWSC	0x0	RAN Restart Auto-Negotiation When set, will cause auto-negotiation to restart if the ANE is set. This bit is self-clearing after auto-negotiation starts. This bit should be cleared for normal operation.
8:0	RO	0x0	reserved

GMAC_AN_STATUS

Address: Operational Base + offset (0x00c4)

AN Status Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RO	0x0	ANC Auto-Negotiation Complete When set, this bit indicates that the auto-negotiation process is completed. This bit is cleared when auto-negotiation is reinitiated.
4	RO	0x0	reserved
3	RO	0x1	ANA Auto-Negotiation Ability This bit is always high, because the GMAC supports auto-negotiation.
2	RWSC	0x0	LS Link Status When set, this bit indicates that the link is up. When cleared, this bit indicates that the link is down.

Bit	Attr	Reset Value	Description
1:0	RO	0x0	reserved

GMAC_AN_ADV

Address: Operational Base + offset (0x00c8)

Auto_Negotiation Advertisement Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support This bit is tied to low, because the GMAC does not support the next page.
14	RO	0x0	reserved
13:12	RW	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating to a link partner that a fault or error condition has occurred.
11:9	RO	0x0	reserved
8:7	RW	0x3	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the GMAC is capable of configuring the PAUSE function as defined in IEEE 802.3x.
6	RW	0x1	HD Half-Duplex This bit, when set high, indicates that the GMAC supports Half-Duplex. This bit is tied to low (and RO) when the GMAC is configured for Full-Duplex-only operation.
5	RW	0x1	FD Full-Duplex This bit, when set high, indicates that the GMAC supports Full-Duplex.
4:0	RO	0x0	reserved

GMAC_AN_LINK_PART_AB

Address: Operational Base + offset (0x00cc)

Auto_Negotiation Link Partner Ability Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support When set, this bit indicates that more next page information is available. When cleared, this bit indicates that next page exchange is not desired.

Bit	Attr	Reset Value	Description
14	RO	0x0	ACK Acknowledge When set, this bit is used by the auto-negotiation function to indicate that the link partner has successfully received the GMAC's base page. When cleared, it indicates that a successful receipt of the base page has not been achieved.
13:12	RO	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating a fault or error condition of the link partner.
11:9	RO	0x0	reserved
8:7	RO	0x0	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the link partner's capability of configuring the PAUSE function as defined in IEEE 802.3x.
6	RO	0x0	HD Half-Duplex When set, this bit indicates that the link partner has the ability to operate in Half-Duplex mode. When cleared, the link partner does not have the ability to operate in Half-Duplex mode.
5	RO	0x0	FD Full-Duplex When set, this bit indicates that the link partner has the ability to operate in Full-Duplex mode. When cleared, the link partner does not have the ability to operate in Full-Duplex mode.
4:0	RO	0x0	reserved

GMAC_AN_EXP

Address: Operational Base + offset (0x00d0)

Auto_Negotiation Expansion Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	NPA Next Page Ability This bit is tied to low, because the GMAC does not support next page function.
1	RO	0x0	NPR New Page Received When set, this bit indicates that a new page has been received by the GMAC. This bit will be cleared when read.
0	RO	0x0	reserved

GMAC_INTF_MODE_STA

Address: Operational Base + offset (0x00d8)

RGMII Status Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	LST Link Status Indicates whether the link is up (1'b1) or down (1'b0)
2:1	RO	0x0	LSD Link Speed Indicates the current speed of the link: 2'b00: 2.5 MHz 2'b01: 25 MHz 2'b10: 125 MHz
0	RW	0x0	LM Link Mode Indicates the current mode of operation of the link: 1'b0: Half-Duplex mode 1'b1: Full-Duplex mode

GMAC_MMC_CTRL

Address: Operational Base + offset (0x0100)

MMC Control Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	FHP Full-Half preset When low and bit4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0xFFFF_F800 (half - 2KBytes) and all frame-counters gets preset to 0xFFFF_FFF0 (half - 16) When high and bit4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2KBytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16)
4	R/WSC	0x0	CP Counters Preset When set, all counters will be initialized or preset to almost full or almost half as per Bit5 above. This bit will be cleared automatically after 1 clock cycle. This bit along with bit5 is useful for debugging and testing the assertion of interrupts due to MMC counter becoming half-full or full.

Bit	Attr	Reset Value	Description
3	RW	0x0	MCF MMC Counter Freeze When set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.)
2	RW	0x0	ROR Reset on Read When set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read.
1	RW	0x0	CSR Counter Stop Rollover When set, counter after reaching maximum value will not roll over to zero
0	RWSC	0x0	CR Counters Reset When set, all counters will be reset. This bit will be cleared automatically after 1 clock cycle

GMAC_MMCR_RX_INTR

Address: Operational Base + offset (0x0104)

MMC Receive Interrupt Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 The bit is set when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value.
20:19	RO	0x0	reserved
18	RC	0x0	INT18 The bit is set when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value.
17:6	RO	0x0	reserved
5	RW	0x0	INT5 The bit is set when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value.
4	RC	0x0	INT4 The bit is set when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value.
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RC	0x0	INT2 The bit is set when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
1	RC	0x0	INT1 The bit is set when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TX_INTR

Address: Operational Base + offset (0x0108)

MMC Transmit Interrupt Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RC	0x0	INT21 The bit is set when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value.
20	RC	0x0	INT20 The bit is set when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
19	RC	0x0	INT19 The bit is set when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value.
18:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value.
12:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_RX_INT_MSK

Address: Operational Base + offset (0x010c)

MMC Receive Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21	RW	0x0	INT21 Setting this bit masks the interrupt when the rx fifo overflow counter reaches half the maximum value, and also when it reaches the maximum value.
20:19	RO	0x0	reserved
18	RW	0x0	INT18 Setting this bit masks the interrupt when the rx length error counter reaches half the maximum value, and also when it reaches the maximum value.
17:6	RO	0x0	reserved
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rx CRC error counter reaches half the maximum value, and also when it reaches the maximum value.
4	RW	0x0	INT4 Setting this bit masks the interrupt when the rx multicast frames_g counter reaches half the maximum value, and also when it reaches the maximum value.
3	RO	0x0	reserved
2	RW	0x0	INT2 Setting this bit masks the interrupt when the rx octet count_g counter reaches half the maximum value, and also when it reaches the maximum value.
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rx octet count_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rx frame count_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TX_INT_MSK

Address: Operational Base + offset (0x0110)

MMC Transmit Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the tx frame count_g counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
20	RW	0x0	INT20 Setting this bit masks the interrupt when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
19	RW	0x0	INT19 Setting this bit masks the interrupt when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value.
18:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value.
12:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TXOCTETCNT_GB

Address: Operational Base + offset (0x0114)

MMC TX OCTET Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_gb Number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad frames.

GMAC_MMC_TXFRMCNT_GB

Address: Operational Base + offset (0x0118)

MMC TX Frame Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_gb Number of good and bad frames transmitted, exclusive of retried frames.

GMAC_MMC_TXUNDFLWERR

Address: Operational Base + offset (0x0148)

MMC TX Underflow Error

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txunderflowerror Number of frames aborted due to frame underflow error.

GMAC_MMC_TXCARERR

Address: Operational Base + offset (0x0160)

MMC TX Carrier Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txcarriererror Number of frames aborted due to carrier sense error (no carrier or loss of carrier).

GMAC_MMC_TXOCTETCNT_G

Address: Operational Base + offset (0x0164)

MMC TX OCTET Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_g Number of bytes transmitted, exclusive of preamble, in good frames only.

GMAC_MMC_TXFRMCNT_G

Address: Operational Base + offset (0x0168)

MMC TX Frame Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_g Number of good frames transmitted.

GMAC_MMC_RXFRMCNT_GB

Address: Operational Base + offset (0x0180)

MMC RX Frame Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxframecount_gb Number of good and bad frames received.

GMAC_MMC_RXOCTETCNT_GB

Address: Operational Base + offset (0x0184)

MMC RX OCTET Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_gb Number of bytes received, exclusive of preamble, in good and bad frames.

GMAC_MMC_RXOCTETCNT_G

Address: Operational Base + offset (0x0188)

MMC RX OCTET Good Counter

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_g Number of bytes received, exclusive of preamble, only in good frames.

GMAC_MMC_RXMCFRMCNT_G

Address: Operational Base + offset (0x0190)

MMC RX Multicast Frame Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxmulticastframes_g Number of good multicast frames received.

GMAC_MMC_RXCRCERR

Address: Operational Base + offset (0x0194)

MMC RX Carrier

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxcrcerror Number of frames received with CRC error.

GMAC_MMC_RXLENERR

Address: Operational Base + offset (0x01c8)

MMC RX Length Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxlengtherror Number of frames received with length error (Length type field ≠ frame size), for all frames with valid length field.

GMAC_MMC_RXFIFOVRFLW

Address: Operational Base + offset (0x01d4)

MMC RX FIFO Overflow

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxfifooverflow Number of missed received frames due to FIFO overflow.

GMAC_MMC_IPC_INT_MSK

Address: Operational Base + offset (0x0200)

MMC Receive Checksum Offload Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	INT29 Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	RW	0x0	INT27 Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	RO	0x0	reserved
25	RW	0x0	INT25 Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24:23	RO	0x0	reserved
22	RW	0x0	INT22 Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
21:18	RO	0x0	reserved
17	RW	0x0	INT17 Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	RO	0x0	reserved
11	RW	0x0	INT11 Setting this bit masks the interrupt when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
10	RO	0x0	reserved
9	RW	0x0	INT9 Setting this bit masks the interrupt when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
8:7	RO	0x0	reserved
6	RW	0x0	INT6 Setting this bit masks the interrupt when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
4:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMIC_IPC_INTR

Address: Operational Base + offset (0x0208)

MMC Receive Checksum Offload Interrupt Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RC	0x0	INT29 The bit is set when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	RO	0x0	reserved
27	RC	0x0	INT27 The bit is set when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	RO	0x0	reserved
25	RC	0x0	INT25 The bit is set when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24:23	RO	0x0	reserved
22	RC	0x0	INT22 The bit is set when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
21:18	RO	0x0	reserved
17	RC	0x0	INT17 The bit is set when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	RO	0x0	reserved
11	RC	0x0	INT11 The bit is set when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
10	RO	0x0	reserved
9	RC	0x0	INT9 The bit is set when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
8:7	RO	0x0	reserved
6	RC	0x0	INT6 The bit is set when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	RC	0x0	INT5 The bit is set when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
4:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_RXIPV4GFRM

Address: Operational Base + offset (0x0210)

MMC RX IPV4 Good Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_gd_frms Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload

GMAC_MMC_RXIPV4HDERRFRM

Address: Operational Base + offset (0x0214)

MMC RX IPV4 Head Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_frms Number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors

GMAC_MMC_RXIPV6GFRM

Address: Operational Base + offset (0x0224)

MMC RX IPV6 Good Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_gd_frms Number of good IPv6 datagrams received with TCP, UDP, or ICMP payloads.

GMAC_MMC_RXIPV6HDERRFRM

Address: Operational Base + offset (0x0228)
 MMC RX IPV6 Head Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_frms Number of IPv6 datagrams received with header errors (length or version mismatch).

GMAC_MMC_RXUDPERRFRM

Address: Operational Base + offset (0x0234)
 MMC RX UDP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_frms Number of good IP datagrams whose UDP payload has a checksum error.

GMAC_MMC_RXTCPERRFRM

Address: Operational Base + offset (0x023c)
 MMC RX TCP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_frms Number of good IP datagrams whose TCP payload has a checksum error.

GMAC_MMC_RXICMPERRFRM

Address: Operational Base + offset (0x0244)
 MMC RX ICMP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_frms Number of good IP datagrams whose ICMP payload has a checksum error.

GMAC_MMC_RXIPV4HDERROCT

Address: Operational Base + offset (0x0254)
 MMC RX OCTET IPV4 Head Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_octets Number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter.

GMAC_MMC_RXIPV6HDERROCT

Address: Operational Base + offset (0x0268)
 MMC RX OCTET IPV6 Head Error

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_octets Number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the IPv6 header's Length field is used to update this counter.

GMAC_MMU_RXUDPERROCT

Address: Operational Base + offset (0x0274)

MMC RX OCTET UDP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_octets Number of bytes received in a UDP segment that had checksum errors.

GMAC_MMU_RXTCPPERROCT

Address: Operational Base + offset (0x027c)

MMC RX OCTET TCP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_octets Number of bytes received in a TCP segment with checksum errors.

GMAC_MMU_RXICMPERRROCT

Address: Operational Base + offset (0x0284)

MMC RX OCTET ICMP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_octets Number of bytes received in an ICMP segment with checksum errors.

GMAC_BUS_MODE

Address: Operational Base + offset (0x1000)

Bus Mode Register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	AAL Address-Aligned Beats When this bit is set high and the FB bit equals 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address.

Bit	Attr	Reset Value	Description
24	RW	0x0	PBL_Mode 8xPBL Mode When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) eight times. Thus the DMA will transfer data in to a maximum of 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.
23	RW	0x0	USP Use Separate PBL When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines.
22:17	RW	0x01	RPBL RxDMA PBL These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set high.
16	RW	0x0	FB Fixed Burst This bit controls whether the AXI Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AXI will use SINGLE and INCR burst transfer operations.
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x01	<p>PBL Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. When USP is set high, this PBL value is applicable for TxDMA transactions only.</p> <p>The PBL values have the following limitations. The maximum number of beats (PBL) possible is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO, except when specified (as given below). For different data bus widths and FIFO sizes, the valid PBL range (including x8 mode) is provided in the following table. If the PBL is common for both transmit and receive DMA, the minimum Rx FIFO and Tx FIFO depths must be considered. Do not program out-of-range PBL values, because the system may not behave properly.</p> <p>For TxFIFO, valid PBL range in full duplex mode and duplex mode is 128 or less.</p> <p>For RxFIFO, valid PBL range in full duplex mode is all.</p>
7	RO	0x0	reserved
6:2	RW	0x00	<p>DSL Descriptor Skip Length This bit specifies the number of Dword to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode.</p>
1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	R/WSC	0x1	<p>SWR Software Reset When this bit is set, the MAC DMA Controller resets all GMAC Subsystem internal registers and logic. It is cleared automatically after the reset operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core.</p> <p>Note: The reset operation is completed only when all the resets in all the active clock domains are de-asserted. Hence it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion.</p>

GMAC_TX_POLL_DEMAND

Address: Operational Base + offset (0x1004)

Transmit Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>TPD Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register GMAC_CUR_HOST_TX_DESC. If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA Register GMAC_STATUS[2] is asserted. If the descriptor is available, transmission resumes.</p>

GMAC_RX_POLL_DEMAND

Address: Operational Base + offset (0x1008)

Receive Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>RPD Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register GMAC_CUR_HOST_RX_DESC. If that descriptor is not available (owned by Host), reception returns to the Suspended state and Register GMAC_STATUS[7] is not asserted. If the descriptor is available, the Receive DMA returns to active state.</p>

GMAC_RX_DESC_LIST_ADDR

Address: Operational Base + offset (0x100c)

Receive Descriptor List Address Register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SRL Start of Receive List This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.

GMAC_TX_DESC_LIST_ADDR

Address: Operational Base + offset (0x1010)

Transmit Descriptor List Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STL Start of Transmit List This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.

GMAC_STATUS

Address: Operational Base + offset (0x1014)

Status Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RO	0x0	GPI GMAC PMT Interrupt This bit indicates an interrupt event in the GMAC core's PMT module. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.
27	RO	0x0	GMI GMAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the GMAC core. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.

Bit	Attr	Reset Value	Description
26	RO	0x0	<p>GLI GMAC Line interface Interrupt This bit reflects an interrupt event in the GMAC Core's PCS or RGMII interface block. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.</p>
25:23	RO	0x0	<p>EB Error Bits These bits indicate the type of error that caused a Bus Error (e.g., error response on the AXI interface). Valid only with Fatal Bus Error bit (Register GMAC_STATUS[13]) set. This field does not generate an interrupt.</p> <ul style="list-style-type: none"> Bit 23: 1'b1 Error during data transfer by TxDMA 1'b0 Error during data transfer by RxDMA Bit 24: 1'b1 Error during read transfer 1'b0 Error during write transfer Bit 25: 1'b1 Error during descriptor access 1'b0 Error during data buffer access
22:20	RO	0x0	<p>TS Transmit Process State These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt.</p> <ul style="list-style-type: none"> 3'b000: Stopped; Reset or Stop Transmit Command issued. 3'b001: Running; Fetching Transmit Transfer Descriptor. 3'b010: Running; Waiting for status. 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). 3'b100: TIME_STAMP write state. 3'b101: Reserved for future use. 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. 3'b111: Running; Closing Transmit Descriptor.

Bit	Attr	Reset Value	Description
19:17	RO	0x0	<p>RS Receive Process State These bits indicate the Receive DMA FSM state. This field does not generate an interrupt.</p> <p>3'b000: Stopped: Reset or Stop Receive Command issued. 3'b001: Running: Fetching Receive Transfer Descriptor. 3'b010: Reserved for future use. 3'b011: Running: Waiting for receive packet. 3'b100: Suspended: Receive Descriptor Unavailable. 3'b101: Running: Closing Receive Descriptor. 3'b110: TIME_STAMP write state. 3'b111: Running: Transferring the receive packet data from receive buffer to host memory.</p>
16	W1C	0x0	<p>NIS Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE:</p> <p>Register GMAC_STATUS[0]: Transmit Interrupt Register GMAC_STATUS[2]: Transmit Buffer Unavailable Register GMAC_STATUS[6]: Receive Interrupt Register GMAC_STATUS[14]: Early Receive Interrupt Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared.</p>

Bit	Attr	Reset Value	Description
15	W1C	0x0	<p>AIS Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE:</p> <ul style="list-style-type: none"> Register GMAC_STATUS[1]: Transmit Process Stopped Register GMAC_STATUS[3]: Transmit Jabber Timeout Register GMAC_STATUS[4]: Receive FIFO Overflow Register GMAC_STATUS[5]: Transmit Underflow Register GMAC_STATUS[7]: Receive Buffer Unavailable Register GMAC_STATUS[8]: Receive Process Stopped Register GMAC_STATUS[9]: Receive Watchdog Timeout Register GMAC_STATUS[10]: Early Transmit Interrupt Register GMAC_STATUS[13]: Fatal Bus Error <p>Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared.</p>
14	W1C	0x0	<p>ERI Early Receive Interrupt This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt Register GMAC_STATUS[6] automatically clears this bit.</p>
13	W1C	0x0	<p>FBI Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as detailed in [25:23]. When this bit is set, the corresponding DMA engine disables all its bus accesses.</p>
12:11	RO	0x0	reserved
10	W1C	0x0	<p>ETI Early Transmit Interrupt This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO.</p>
9	W1C	0x0	<p>RWT Receive Watchdog Timeout This bit is asserted when a frame with a length greater than 2,048 bytes is received.</p>

Bit	Attr	Reset Value	Description
8	W1C	0x0	RPS Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state.
7	W1C	0x0	RU Receive Buffer Unavailable This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. Register GMAC_STATUS[7] is set only when the previous Receive Descriptor was owned by the DMA.
6	W1C	0x0	RI Receive Interrupt This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state.
5	W1C	0x0	UNF Transmit Underflow This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set.
4	W1C	0x0	OVF Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11].
3	W1C	0x0	TJT Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.

Bit	Attr	Reset Value	Description
2	W1C	0x0	TU Transmit Buffer Unavailable This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command.
1	W1C	0x0	TPS Transmit Process Stopped This bit is set when the transmission is stopped.
0	W1C	0x0	TI Transmit Interrupt This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor.

GMAC_OP_MODE

Address: Operational Base + offset (0x1018)

Operation Mode Register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	DT Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the core does not drop frames that only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors in the encapsulated payload only. When this bit is reset, all error frames are dropped if the FEF bit is reset.
25	RW	0x0	RSF Receive Store and Forward When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits.
24	RW	0x0	DFF Disable Flushing of Received Frames When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset.

Bit	Attr	Reset Value	Description
23:22	RO	0x0	reserved
21	RW	0x0	<p>TSF Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Register GMAC_OP_MODE[16:14] are ignored. This bit should be changed only when transmission is stopped.</p>
20	W1C	0x0	<p>FTF Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter will not be flushed. It will be scheduled for transmission and will result in underflow and runt frame transmission. Note: The flush operation completes only after emptying the TxFIFO of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active.</p>
19:17	RO	0x0	reserved
16:14	RW	0x0	<p>TTC Transmit Threshold Control These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset. 3'b000: 64 3'b001: 128 3'b010: 192 3'b011: 256 3'b100: 40 3'b101: 32 3'b110: 24 3'b111: 16</p>

Bit	Attr	Reset Value	Description
13	RW	0x0	<p>ST Start/Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register GMAC_TX_DESC_LIST_ADDR, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (Register GMAC_STATUS[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting DMA Register TX_DESC_LIST_ADDR, then the DMA behavior is unpredictable.</p> <p>When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only the transmission of the current frame is complete or when the transmission is in the Suspended state.</p>
12:11	RW	0x0	<p>RFD Threshold for deactivating flow control (in both HD and FD) These bits control the threshold (Fill-level of Rx FIFO) at which the flow-control is deasserted after activation. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the deassertion is effective only after flow control is asserted.</p>
10:9	RW	0x0	<p>RFA Threshold for activating flow control (in both HD and FD) These bits control the threshold (Fill level of Rx FIFO) at which flow control is activated. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the above only applies to Rx FIFOs of 4 KB or more when the EFC bit is set high.</p>

Bit	Attr	Reset Value	Description
8	RW	0x0	EFC Enable HW flow control When this bit is set, the flow control signal operation based on fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled.
7	RW	0x0	FEF Forward Error Frames When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped. When FEF is set, all frames except runt error frames are forwarded to the DMA. But when RxFIFO overflows when a partial frame is written, then such frames are dropped even when FEF is set.
6	RW	0x0	FUF Forward Undersized Good Frames When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). When reset, the Rx FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold (e.g., RTC = 01).
5	RO	0x0	reserved
4:3	RW	0x0	RTC Receive Threshold Control These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. 2'b00: 64 2'b01: 32 2'b10: 96 2'b11: 128
2	RW	0x0	OSF Operate on Second Frame When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained.

Bit	Attr	Reset Value	Description
1	RW	0x0	<p>SR Start/Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by register GMAC_RX_DESC_LIST_ADDR or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (Register GMAC_STATUS[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting register GMAC_RX_DESC_LIST_ADDR, DMA behavior is unpredictable.</p> <p>When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.</p>
0	RO	0x0	reserved

GMAC_INT_ENA

Address: Operational Base + offset (0x101c)

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	<p>NIE Normal Interrupt Summary Enable When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits:</p> <p>Register GMAC_STATUS[0]: Transmit Interrupt Register GMAC_STATUS[2]: Transmit Buffer Unavailable Register GMAC_STATUS[6]: Receive Interrupt Register GMAC_STATUS[14]: Early Receive Interrupt</p>

Bit	Attr	Reset Value	Description
15	RW	0x0	<p>AIE Abnormal Interrupt Summary Enable When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits</p> <p>Register GMAC_STATUS[1]: Transmit Process Stopped Register GMAC_STATUS[3]: Transmit Jabber Timeout Register GMAC_STATUS[4]: Receive Overflow Register GMAC_STATUS[5]: Transmit Underflow Register GMAC_STATUS[7]: Receive Buffer Unavailable Register GMAC_STATUS[8]: Receive Process Stopped Register GMAC_STATUS[9]: Receive Watchdog Timeout Register GMAC_STATUS[10]: Early Transmit Interrupt Register GMAC_STATUS[13]: Fatal Bus Error</p>
14	RW	0x0	<p>ERE Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled.</p>
13	RW	0x0	<p>FBE Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled.</p>
12:11	RO	0x0	reserved
10	RW	0x0	<p>ETE Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (BIT 15), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled.</p>
9	RW	0x0	<p>RWE Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled.</p>

Bit	Attr	Reset Value	Description
8	RW	0x0	RSE Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled.
7	RW	0x0	RUE Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled
6	RW	0x0	RIE Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled.
5	RW	0x0	UNE Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled.
4	RW	0x0	OVE Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled
3	RW	0x0	TJE Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled.
2	RW	0x0	TUE Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled.
1	RW	0x0	TSE Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled.

Bit	Attr	Reset Value	Description
0	RW	0x0	TIE Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled.

GMAC_OVERFLOW_CNT

Address: Operational Base + offset (0x1020)

Missed Frame and Buffer Overflow Counter Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RC	0x0	FIFO_overflow_bit Overflow bit for FIFO Overflow Counter
27:17	RC	0x000	Frame_miss_number Indicates the number of frames missed by the application This counter is incremented each time the MTL asserts the sideband signal mtl_rxoverflow_o. The counter is cleared when this register is read with mci_be_i[2] at 1'b1.
16	RC	0x0	Miss_frame_overflow_bit Overflow bit for Missed Frame Counter
15:0	RC	0x0000	Frame_miss_number_2 Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.

GMAC_REC_INT_WDT_TIMER

Address: Operational Base + offset (0x1024)

Receive Interrupt Watchdog Timer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>RIWT RI Watchdog Timer count Indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the RxDMA completes the transfer of a frame for which the RI status bit is not set due to the setting in the corresponding descriptor RDES1[31]. When the watch-dog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when RI bit is set high due to automatic setting of RI as per RDES1[31] of any received frame.</p>

GMAC_AXI_BUS_MODE

Address: Operational Base + offset (0x1028)

AXI Bus Mode Register

Bit	Attr	Reset Value	Description
31	RW	0x0	<p>EN_LPI Enable LPI (Low Power Interface) When set to 1, enable the LPI (Low Power Interface) supported by the GMAC and accepts the LPI request from the AXI System Clock controller. When set to 0, disables the Low Power Mode and always denies the LPI request from the AXI System Clock controller.</p>
30	RW	0x0	<p>UNLCK_ON_MGK_RWK Unlock on Magic Packet or Remote Wake Up When set to 1, enables it to request coming out of Low Power mode only when Magic Packet or Remote Wake Up Packet is received. When set to 0, enables it requests to come out of Low Power mode when any frame is received.</p>
29:22	RO	0x0	reserved
21:20	RW	0x1	<p>WR_OSR_LMT AXI Maximum Write Out Standing Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1</p>
19:18	RO	0x0	reserved

Bit	Attr	Reset Value	Description
17:16	RW	0x1	RD_OSR_LMT AXI Maximum Read Out Standing Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1
15:13	RO	0x0	reserved
12	RO	0x0	AXI_AAL Address-Aligned Beats This bit is read-only bit and reflects the AAL bit Register0 (register GMAC_BUS_MODE[25]). When this bit set to 1, it performs address-aligned burst transfers on both read and write channels.
11:4	RO	0x0	reserved
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 16.
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 8.
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 4.
0	RO	0x1	UNDEF AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of FB bit in register GMAC_BUS_MODE[16]. When this bit is set to 1, it is allowed to perform any burst length equal to or below the maximum allowed burst length as programmed in bits[7:1]; When this bit is set to 0, the it is allowed to perform only fixed burst lengths as indicated by BLEN256/128/64/32/16/8/4, or a burst length of 1.

GMAC_AXI_STATUS

Address: Operational Base + offset (0x102c)

AXI Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	RD_CH_STA When high, it indicates that AXI Master's read channel is active and transferring data.

Bit	Attr	Reset Value	Description
0	RO	0x0	WR_CH_STA When high, it indicates that AXI Master's write channel is active and transferring data.

GMAC_CUR_HOST_TX_DESC

Address: Operational Base + offset (0x1048)

Current Host Transmit Descriptor Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTDAP Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_RX_DESC

Address: Operational Base + offset (0x104c)

Current Host Receive Descriptor Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRDAP Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_TX_Buf_ADDR

Address: Operational Base + offset (0x1050)

Current Host Transmit Buffer Address Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTBAP Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_RX_BUF_ADDR

Address: Operational Base + offset (0x1054)

Current Host Receive Buffer Adderss Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRBAP Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_HW_FEA_REG

Address: Operational Base + offset (0x1058)

The presence of the optional features/functions of the core Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	Reserved
24	RO	0x0	Alternate (Enhanced Descriptor)
23:20	RO	0x0	Reserved
19	RO	0x1	RxFIFO > 2048 Bytes

Bit	Attr	Reset Value	Description
18	RO	0x1	IP Checksum Offload (Type 2) in Rx
17	RO	0x0	IP Checksum Offload (Type 1) in Rx
16	RO	0x1	Checksum Offload in Tx
15:14	RO	0x0	Reserved
13	RO	0x0	IEEE 1588-2008 Advanced Time-Stamp
12	RO	0x0	IEEE 1588-2002 Time-Stamp
11	RO	0x1	RMON module
10	RO	0x1	PMT Magic Packet
9	RO	0x1	PMT Remote Wakeup
8	RO	0x1	SMA (MDIO) Interface
7	RO	0x0	Reserved
6	RO	0x0	PCS registers (TBI/SGMII/RTBI PHY interface)
5	RO	0x0	Multiple MAC Address Registers
4	RO	0x1	HASH Filter
3	RO	0x0	Reserved
2	RO	0x1	Half-Duplex support
1	RO	0x1	1000 Mbps support
0	RO	0x1	10/100 Mbps support

44.5 Interface Description

Table 44-1 RMII Interface Description

Module pin name	Direction	Pad name	IOMUX
RMII interface			
mac_clk	I/O	GPIO2_B[6]	GRF_GPIO2B_IOMUX[13:12]=2'b11
mac_txen	O	GPIO2_B[5]	GRF_GPIO2B_IOMUX[11:10]=2'b11
mac_txd1	O	GPIO2_C[2]	GRF_GPIO2C_IOMUX[5:4]=2'b11
mac_txd0	O	GPIO2_C[3]	GRF_GPIO2C_IOMUX[7:6]=2'b11
mac_rxdrv	I	GPIO2_B[0]	GRF_GPIO2B_IOMUX[1:0]=2'b11
mac_rxer	I	GPIO2_B[7]	GRF_GPIO2B_IOMUX[15:14]=2'b11
mac_rxd1	I	GPIO2_C[0]	GRF_GPIO2C_IOMUX[1:0]=2'b11
mac_rxd0	I	GPIO2_C[1]	GRF_GPIO2C_IOMUX[3:2]=2'b11
Management interface			
mac_mdio	I/O	GPIO2_B[4]	GRF_GPIO2B_IOMUX[9:8]=2'b11
mac_mdc	O	GPIO2_D[1]	GRF_GPIO2D_IOMUX[3:2]=2'b11

Table 44-2 RGMII Interface Description

Module pin name	Direction	Pad name	IOMUX
RGMII/RMII interface			
mac_clk	I/O	GPIO2_B[6]	GRF_GPIO2B_IOMUX[13:12]=2'b11
mac_txclk	O	GPIO2_B[1]	GRF_GPIO2B_IOMUX[3:2]=2'b11
mac_txen	O	GPIO2_B[5]	GRF_GPIO2B_IOMUX[11:10]=2'b11
mac_txd3	O	GPIO2_C[7]	GRF_GPIO2C_IOMUX2[14:12]=3'b100
mac_txd2	O	GPIO2_C[6]	GRF_GPIO2C_IOMUX2[10:8]=3'b100
mac_txd1	O	GPIO2_C[2]	GRF_GPIO2C_IOMUX[5:4]=2'b11
mac_txd0	O	GPIO2_C[3]	GRF_GPIO2C_IOMUX[7:6]=2'b11
mac_rxclk	I	GPIO2_B[3]	GRF_GPIO2B_IOMUX[7:6]=2'b11
mac_rxdrv	I	GPIO2_B[0]	GRF_GPIO2B_IOMUX[1:0]=2'b11
mac_rxd3	I	GPIO2_C[4]	GRF_GPIO2C_IOMUX2[2:0]=3'b100

mac_rxd2	I	GPIO2_C[5]	GRF_GPIO2C_IOMUX[6:4]=3'b100
mac_rxd1	I	GPIO2_C[0]	GRF_GPIO2C_IOMUX[1:0]=2'b11
mac_rxd0	I	GPIO2_C[1]	GRF_GPIO2C_IOMUX[3:2]=2'b11
mac_crs	I	GPIO2_B[2]	GRF_GPIO2B_IOMUX[5:4]=2'b11
mac_col	I	GPIO2_D[0]	GRF_GPIO2D_IOMUX[14:12]=3'b100
Management interface			
mac_mdio	I/O	GPIO2_B[4]	GRF_GPIO2B_IOMUX[9:8]=2'b11
mac_mdc	O	GPIO2_D[1]	GRF_GPIO2D_IOMUX[3:2]=2'b11

44.6 Application Notes

44.6.1 Descriptors

The DMA in GMAC can communicate with Host driver through descriptor lists and data buffers. The DMA transfers data frames received by the core to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.

There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into DMA Registers RX_DESC_LIST_ADDR and TX_DESC_LIST_ADDR, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1[24] and TDES1[24]). The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the Host physical memory space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled

The descriptor ring and chain structure is shown in following figure.

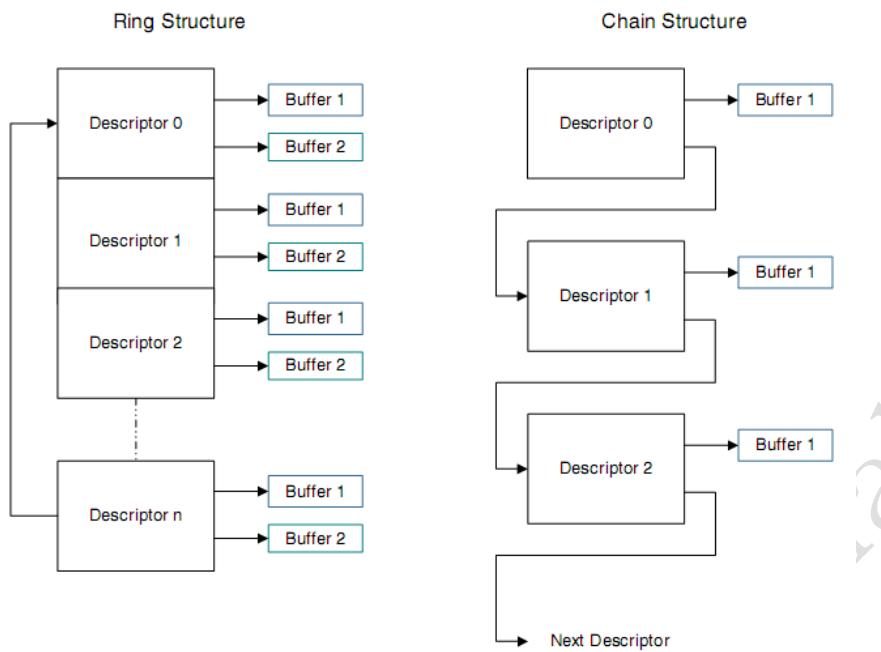


Fig. 44-10 Descriptor Ring and Chain Structure

Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes. The descriptor addresses must be aligned to the bus width used (Word/Dword/Lword for 32/64/128-bit buses).

DES1-DES0	63 Control Bits [9:0]	55 Byte Count Buffer2 [10:0]	47 Byte Count Buffer1[10:0]	31 O W N	23 Status [30:0]	15 7 0
DES3-DES2	Buffer2 Address [31:0] / Next Descriptor Address [31:0]			Buffer1 Address[31:0]		

Fig. 44-11 Rx/Tx Descriptors definition

44.6.2 Receive Descriptor

The GMAC Subsystem requires at least two descriptors when receiving a frame. The Receive state machine of the DMA always attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the RxDMA closes a descriptor, it will attempt to acquire the next descriptor even if no frames are received.

In a single descriptor (receive) system, the subsystem will generate a descriptor error if the receive buffer is unable to accommodate the incoming frame and the next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

Receive Descriptor 0 (RDES0)

RDES0 contains the received frame status, the frame length, and the descriptor ownership information.

Table 44-3 Receive Descriptor 0

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA of

	the GMAC Subsystem. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	AFM: Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the GMAC Core.
29:16	FL: Frame Length These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.
15	ES: Error Summary Indicates the logical OR of the following bits: <ul style="list-style-type: none">• RDES0[0]: Payload Checksum Error• RDES0[1]: CRC Error• RDES0[3]: Receive Error• RDES0[4]: Watchdog Timeout• RDES0[6]: Late Collision• RDES0[7]: IPC Checksum• RDES0[11]: Overflow Error• RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
14	DE: Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set
13	SAF: Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the GMAC Core.
12	LE: Length Error When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset. Length error status is not valid when CRC error is present.
11	OE: Overflow Error When set, this bit indicates that the received frame was damaged due to buffer overflow.
10	VLAN: VLAN Tag When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the GMACCore.
9	FS: First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the

	next Descriptor contains the beginning of the frame.
8	<p>LS: Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.</p>
7	<p>IPC: IP Checksum Error/Giant Frame When IP Checksum Engine is enabled, this bit, when set, indicates that the 16-bit IPv4 Header checksum calculated by the core did not match the received checksum bytes. The Error Summary bit[15] is NOT set when this bit is set in this mode.</p>
6	<p>LC: Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in Half-Duplex mode.</p>
5	<p>FT: Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 16'h0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes.</p>
4	<p>RWT: Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.</p>
3	<p>RE: Receive Error When set, this bit indicates that the gmii_rxer_i signal is asserted while gmii_rxdv_i is asserted during frame reception. This error also includes carrier extension error in GMII and Half-duplex mode. Error can be of less/no extension, or error ($rxd \neq 0f$) during extension.</p>
2	<p>DE: Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in MII Mode.</p>
1	<p>CE: CRC Error When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.</p>
0	<p>Rx MAC Address/Payload Checksum Error When set, this bit indicates that the Rx MAC Address registers value (1 to 15) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field. If Full Checksum Offload Engine is enabled, this bit, when set, indicates the TCP, UDP, or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP, or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame.</p>

Receive Descriptor 1 (RDES1)

RDES1 contains the buffer sizes and other bits that control the descriptor chain/ring.

Table 44-4 Receive Descriptor 1

Bit	Description
31	<p>Disable Interrupt on Completion When set, this bit will prevent the setting of the RI (CSR5[6]) bit of the</p>

	GMAC_STATUS Register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt to Host due to RI for that frame.
30:26	Reserved.
25	RER: Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
24	RCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When RDES1[24] is set, RBS2 (RDES1[21-11]) is a "don't care" value. RDES1[25] takes precedence over RDES1[24].
23:22	Reserved.
21:11	RBS2: Receive Buffer 2 Size These bits indicate the second data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. This field is not valid if RDES1[24] is set.
10:0	RBS1: Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES2 (buffer1 address pointer) is not aligned. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 24).

Receive Descriptor 2 (RDES2)

RDES2 contains the address pointer to the first data buffer in the descriptor.

Table 44-5 Receive Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. Note that the DMA performs a write operation with the RDES2[2:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[2:0] (corresponding to bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

Receive Descriptor 3 (RDES3)

RDES3 contains the address pointer either to the second data buffer in the descriptor or to the next descriptor.

Table 44-6 Receive Descriptor 3

Bit	Description
31:0	<p>Buffer 2 Address Pointer (Next Descriptor Address)</p> <p>These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present.</p> <p>If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[2:0] = 0, corresponding to a bus width of 64. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3[2:0] (corresponding to a bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.</p>

44.6.3 Transmit Descriptor

The descriptor addresses must be aligned to the bus width used (64). Each descriptor is provided with two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory-management schemes.

Transmit Descriptor 0 (TDES0)

TDES0 contains the transmitted frame status and the descriptor ownership information.

Table 44-7 Transmit Descriptor 0

Bit	Description
31	<p>OWN: Own Bit</p> <p>When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.</p>
30:17	Reserved.
16	<p>IHE: IP Header Error</p> <p>When set, this bit indicates that the Checksum Offload engine detected an IP header error and consequently did not modify the transmitted frame for any checksum insertion.</p>
15	<p>ES: Error Summary</p> <p>Indicates the logical OR of the following bits:</p> <ul style="list-style-type: none"> • TDES0[14]: Jabber Timeout • TDES0[13]: Frame Flush • TDES0[11]: Loss of Carrier • TDES0[10]: No Carrier • TDES0[9]: Late Collision

	<ul style="list-style-type: none"> • TDES0[8]: Excessive Collision • TDES0[2]: Excessive Deferral • TDES0[1]: Underflow Error
14	JT: Jabber Timeout When set, this bit indicates the GMAC transmitter has experienced a jabber time-out.
13	FF: Frame Flushed When set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.
12	PCE: Payload Checksum Error This bit, when set, indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either due to insufficient bytes, as indicated by the IP Header's Payload Length field, or the MTL starting to forward the frame to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet frame being transmitted: to avoid deadlock, the MTL starts forwarding the frame when the FIFO is full, even in Store-and-Forward mode.
11	LC: Loss of Carrier When set, this bit indicates that Loss of Carrier occurred during frame transmission. This is valid only for the frames transmitted without collision and when the GMAC operates in Half-Duplex Mode.
10	NC: No Carrier When set, this bit indicates that the carrier sense signal from the PHY was not asserted during transmission.
9	LC: Late Collision When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in RMII Mode and 512 byte times including Preamble and Carrier Extension in RGMII Mode). Not valid if Underflow Error is set.
8	EC: Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DR (Disable Retry) bit in the GMAC Configuration Register is set, this bit is set after the first collision and the transmission of the frame is aborted.
7	VF: VLAN Frame When set, this bit indicates that the transmitted frame was a VLAN-type frame.
6:3	CC: Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.
2	ED: Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000-Mbps mode) if the Deferral Check (DC) bit is set high in the GMAC Control Register.
1	UF: Underflow Error When set, this bit indicates that the GMAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that

	the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (Register GMAC_STATUS[5]) and Transmit Interrupt (Register GMAC_STATUS [0]).
0	<p>DB: Deferred Bit</p> <p>When set, this bit indicates that the GMAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.</p>

Transmit Descriptor 1 (TDES1)

TDES1 contains the buffer sizes and other bits which control the descriptor chain/ring and the frame being transferred.

Table 44-8 Transmit Descriptor 1

Bit	Description
31	<p>IC: Interrupt on Completion</p> <p>When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.</p>
30	<p>LS: Last Segment</p> <p>When set, this bit indicates that the buffer contains the last segment of the frame.</p>
29	<p>FS: First Segment</p> <p>When set, this bit indicates that the buffer contains the first segment of a frame.</p>
28:27	<p>CIC: Checksum Insertion Control</p> <p>These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below.</p> <ul style="list-style-type: none"> • 2'b00: Do nothing. Checksum Engine is bypassed • 2'b01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. • 2'b10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. • 2'b11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. <p>The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly.</p>
26	<p>DC: Disable CRC</p> <p>When set, the GMAC does not append the Cyclic Redundancy Check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES1[29]).</p>
25	<p>TER: Transmit End of Ring</p> <p>When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.</p>
24	TCH: Second Address Chained

	When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1[24] is set, TBS2 (TDES1[21–11]) are “don’t care” values. TDES1[25] takes precedence over TDES1[24].
23	DP: Disable Padding When set, the GMAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1[29]) is set.
22	Reserved.
21:11	TBS2: Transmit Buffer 2 Size These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1[24] is set.
10:0	TBS1: Transmit Buffer 1 Size These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).

Transmit Descriptor 2 (TDES2)

TDES2 contains the address pointer to the first buffer of the descriptor.

Table 44-9 Transmit Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Transmit Descriptor 3 (TDES3)

TDES3 contains the address pointer either to the second buffer of the descriptor or the next descriptor.

Table 44-10 Transmit Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

44.6.4 Programming Guide

DMA Initialization – Descriptors

The following operations must be performed to initialize the DMA.

1. Provide a software reset. This will reset all of the GMAC internal registers and logic. (GMAC_OP_MODE[0]).
2. Wait for the completion of the reset process (poll GMAC_OP_MODE[0], which

is only cleared after the reset operation is completed).

3. Program the following fields to initialize the Bus Mode Register by setting values in register GMAC_BUS_MODE

- a. Mixed Burst and AAL
- b. Fixed burst or undefined burst
- c. Burst length values and burst mode values.
- d. Descriptor Length (only valid if Ring Mode is used)
- e. Tx and Rx DMA Arbitration scheme

4. Program the AXI Interface options in the register GMAC_BUS_MODE

a. If fixed burst-length is enabled, then select the maximum burst-length possible on the AXI bus (Bits[7:1])

5. A proper descriptor chain for transmit and receive must be created. It should also ensure that the receive descriptors are owned by DMA (bit 31 of descriptor should be set). When OSF mode is used, at least two descriptors are required.

6. Software should create three or more different transmit or receive descriptors in the chain before reusing any of the descriptors.

7. Initialize receive and transmit descriptor list address with the base address of transmit and receive descriptor (register GMAC_RX_DESC_LIST_ADDR and GMAC_TX_DESC_LIST_ADDR).

8. Program the following fields to initialize the mode of operation by setting values in register GMAC_OP_MODE

- a. Receive and Transmit Store And Forward
- b. Receive and Transmit Threshold Control (RTC and TTC)
- c. Hardware Flow Control enable
- d. Flow Control Activation and De-activation thresholds for MTL Receive and Transmit FIFO (RFA and RFD)
- e. Error Frame and undersized good frame forwarding enable
- f. OSF Mode

9. Clear the interrupt requests, by writing to those bits of the status register (interrupt bits only) which are set. For example, by writing 1 into bit 16 - normal interrupt summary will clear this bit (register GMAC_STATUS).

10. Enable the interrupts by programming the interrupt enable register GMAC_INT_ENA.

11. Start the Receive and Transmit DMA by setting SR (bit 1) and ST (bit 13) of the control register GMAC_OP_MODE.

MAC Initialization

The following MAC Initialization operations can be performed after the DMA initialization sequence. If the MAC Initialization is done before the DMA is set-up, then enable the MAC receiver (last step below) only after the DMA is active. Otherwise, received frames will fill the RxFIFO and overflow. Steps (1) and (2) are to be followed if the TBI/SGMII/RTBI PHY interface is enabled, otherwise follow steps (3) and (4).

1. Program the AN Control register GMAC_AN_CTRL to enable Auto-negotiation ANE (bit-12). Setting ELE (bit-14) of this register will enable the PHY to loop back the transmit data.

2. Check the AN Status Register GMAC_AN_STATUS for completion of the Auto-negotiation process. ANC (bit-5) should be set, and link status (bit-2), when set, indicates that the link is up.
3. Program the register GMAC_GMII_ADDR for controlling the management cycles for external PHY, for example, Physical Layer Address PA (bits 15-11). Also set bit 0 (GMII Busy) for writing into PHY and reading from PHY.
4. Read the 16-bit data of (GMAC_GMII_DATA) from the PHY for link up, speed of operation, and mode of operation, by specifying the appropriate address value in register GMAC_GMII_ADDR (bits 15-11).
5. Provide the MAC address registers (GMAC_MAC_ADDR0_HI and GMAC_MAC_ADDR0_LO). If more than one MAC address is enabled in your configuration (during configuration in coreConsultant), program them appropriately.
6. If Hash filtering is enabled in your configuration, program the Hash filter register (GMAC_HASH_TAB_HI and GMAC_HASH_TAB_LO).
7. Program the following fields to set the appropriate filters for the incoming frames in register GMAC_MAC_FRM_FILT
 - a. Receive All
 - b. Promiscuous mode
 - c. Hash or Perfect Filter
 - d. Unicast, Multicast, broad cast and control frames filter settings etc.
8. Program the following fields for proper flow control in register GMAC_FLOW_CTRL.
 - a. Pause time and other pause frame control bits
 - b. Receive and Transmit Flow control bits
 - c. Flow Control Busy/Backpressure Activate
9. Program the Interrupt Mask register bits, as required, and if applicable, for your configuration.
10. Program the appropriate fields in register GMAC_MAC_CONF for example, Inter-frame gap while transmission, jabber disable, etc. Based on the Auto-negotiation you can set the Duplex mode (bit 11), port select (bit 15), etc.
11. Set the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register GMAC_MAC_CONF.

Normal Receive and Transmit Operation

For normal operation, the following steps can be followed.

- For normal transmit and receive interrupts, read the interrupt status. Then poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
- On completion of the above step, set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
- If the descriptors were not owned by the DMA (or no descriptor is available), the DMA will go into SUSPEND

state. The transmission or reception can be resumed by freeing the descriptors and issuing a poll demand by writing 0 into the Tx/Rx poll demand register (GMAC_TX_POLL_DEMAND and GMAC_RX_POLL_DEMAND).

- The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (GMAC_CUR_HOST_TX_DESC and GMAC_CUR_HOST_RX_DESC).
- The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (GMAC_CUR_HOST_TX_Buf_ADDR and GMAC_CUR_HOST_RX_Buf_ADDR).

Stop and Start Operation

When the transmission is required to be paused for some time then the following steps can be followed.

1. Disable the Transmit DMA (if applicable), by clearing ST (bit 13) of the control register GMAC_OP_MODE.
2. Wait for any previous frame transmissions to complete. This can be checked by reading the appropriate bits of MAC Debug register.
3. Disable the MAC transmitter and MAC receiver by clearing the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register GMAC_MAC_CONF.
4. Disable the Receive DMA (if applicable), after making sure the data in the RX FIFO is transferred to the system memory (by reading the register GMAC_DEBUG).
5. Make sure both the TX FIFO and RX FIFO are empty.
6. To re-start the operation, start the DMAs first, before enabling the MAC Transmitter and Receiver.

44.6.5 Clock Architecture

In RMII mode, reference clock and TX/RX clock can be from CRU or external OSC as following figure.

The mux select rmii_speed is GRF_MAC_CON1[11].

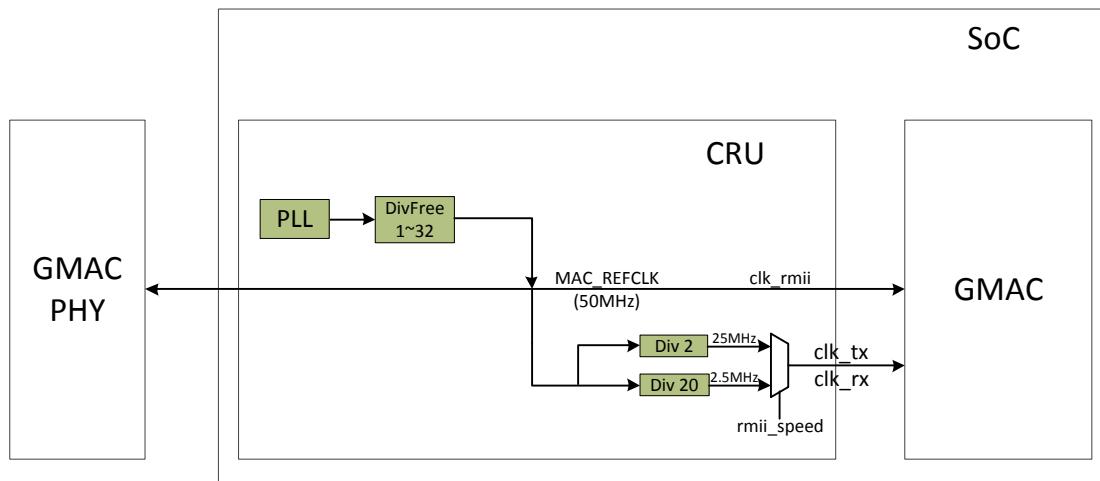


Fig. 44-12 RMII clock architecture when clock source from CRU

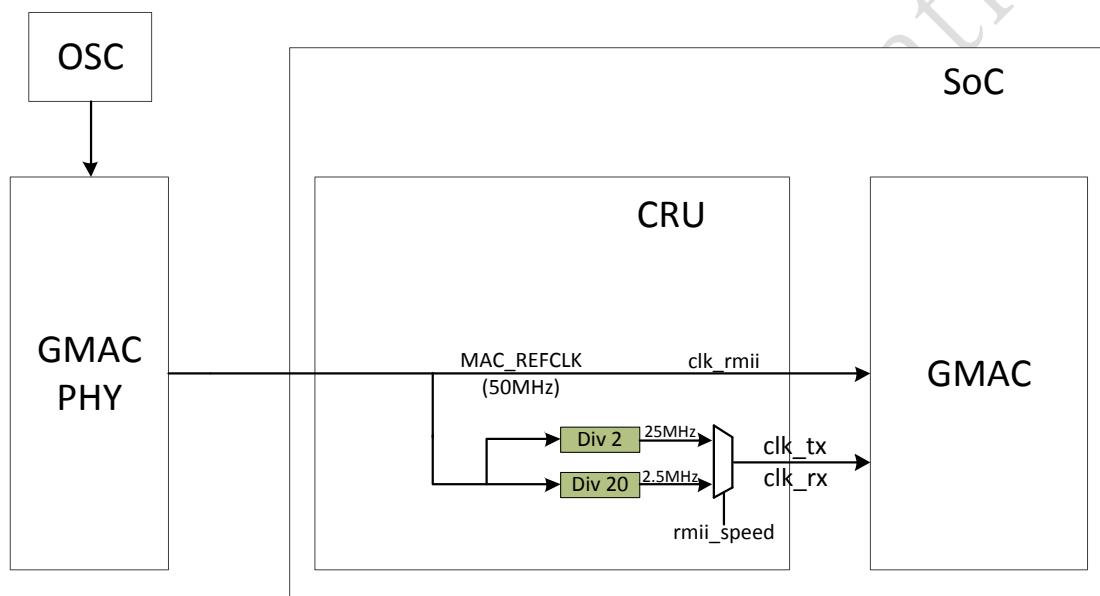


Fig. 44-13 RMII clock architecture when clock source from external OSC

In RGMII mode, clock architecture only supports that TX clock source is from CRU as following figure.

In order to dynamically adjust the timing between TX/RX clock with data, delayline is integrated in TX and RX clock path. Register GRF_MAC_CON0[15:14] can enable the delaylines, and GRF_MAC_CON0[13:0] is used to determine the delay length. There are 100 delay elements in each delayline, and it totally can adjust about 5ns typically.

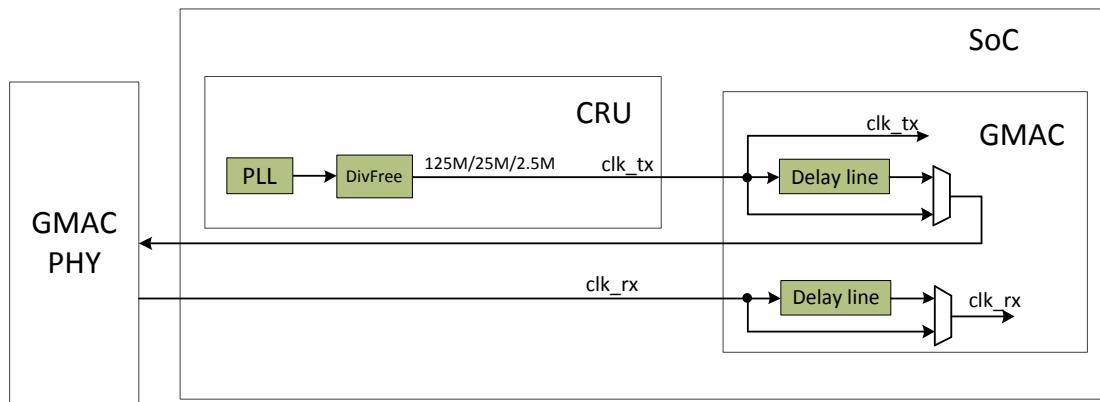


Fig. 44-14 RGMII clock architecture when clock source from CRU

44.6.6 Remote Wake-Up Frame Filter Register

The register `wkupfmfilter_reg`, address (028H), loads the Wake-up Frame Filter register. To load values in a Wake-up Frame Filter register, the entire register (`wkupfmfilter_reg`) must be written. The `wkupfmfilter_reg` register is loaded by sequentially loading the eight register values in address (028) for `wkupfmfilter_reg0`, `wkupfmfilter_reg1`, ... `wkupfmfilter_reg7`, respectively. `wkupfmfilter_reg` is read in the same way.

The internal counter to access the appropriate `wkupfmfilter_reg` is incremented when lane3 (or lane 0 in big-endian) is accessed by the CPU. This should be kept in mind if you are accessing these registers in byte or half-word mode.

<code>wkupfmfilter_reg0</code>	Filter 0 Byte Mask											
<code>wkupfmfilter_reg1</code>	Filter 1 Byte Mask											
<code>wkupfmfilter_reg2</code>	Filter 2 Byte Mask											
<code>wkupfmfilter_reg3</code>	Filter 3 Byte Mask											
<code>wkupfmfilter_reg4</code>	RSVD	Filter 3 Command	RSVD	Filter 2 Command	RSVD	Filter 1 Command	RSVD	Filter 0 Command				
<code>wkupfmfilter_reg5</code>	Filter 3 Offset		Filter 2 Offset		Filter 1 Offset		Filter 0 Offset					
<code>wkupfmfilter_reg6</code>	Filter 1 CRC - 16				Filter 0 CRC - 16							
<code>wkupfmfilter_reg7</code>	Filter 3 CRC - 16				Filter 2 CRC - 16							

Fig. 44-15 Wake-Up Frame Filter Register

Filter i Byte Mask

This register defines which bytes of the frame are examined by filter i (0, 1, 2, and 3) in order to determine whether or not the frame is a wake-up frame. The MSB (thirty-first bit) must be zero. Bit j [30:0] is the Byte Mask. If bit j (byte number) of the Byte Mask is set, then Filter i Offset + j of the incoming frame is processed by the CRC block; otherwise Filter i Offset + j is ignored.

Filter i Command

This 4-bit command controls the filter i operation. Bit 3 specifies the address type, defining the pattern's destination address type. When the bit is set, the pattern applies to only multicast frames; when the bit is reset, the pattern applies only to unicast frame. Bit 2 and Bit 1 are reserved. Bit 0 is the enable for filter i; if Bit 0 is not set, filter i is disabled.

Filter i Offset

This register defines the offset (within the frame) from which the frames are examined by filter i. This 8-bit pattern-offset is the offset for the filter i first byte to examined. The minimum allowed is 12, which refers to the 13th byte of the frame (offset value 0 refers to the first byte of the frame).

Filter i CRC-16

This register contains the CRC_16 value calculated from the pattern, as well as the byte mask programmed to the wake-up filter register block.

44.6.7 System Consideration During Power-Down

GMAC neither gates nor stops clocks when Power-Down mode is enabled. Power saving by clock gating must be done outside the core by the CRU. The receive data path must be clocked with clk_rx_i during Power-Down mode, because it is involved in magic packet/wake-on-LAN frame detection. However, the transmit path and the APB path clocks can be gated off during Power-Down mode.

The pmt interrupt is asserted when a valid wake-up frame is received. This interrupt is generated in the clk_rx domain.

The recommended power-down and wake-up sequence is as follows.

1. Disable the Transmit DMA (if applicable) and wait for any previous frame transmissions to complete. These transmissions can be detected when Transmit Interrupt (TI - Register GMAC_STATUS[0]) is received.
2. Disable the MAC transmitter and MAC receiver by clearing the appropriate bits in the MAC Configuration register.
3. Wait until the Receive DMA empties all the frames from the Rx FIFO (a software timer may be required).
4. Enable Power-Down mode by appropriately configuring the PMT registers.
5. Enable the MAC Receiver and enter Power-Down mode.
6. Gate the APB and transmit clock inputs to the core (and other relevant clocks in the system) to reduce power and enter Sleep mode.
7. On receiving a valid wake-up frame, the GMAC asserts the pmt interrupt signal and exits Power-Down mode.
8. On receiving the interrupt, the system must enable the APB and transmit clock inputs to the core.
9. Read the register GMAC_PMT_CTRL_STA to clear the interrupt, then enable the other modules in the system and resume normal operation.

44.6.8 GRF Register Summary

GRF Register	Register Description
GRF_MAC_CON1[8:6]	PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved
GRF_MAC_CON1[9]	GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the GMAC enables the

	Back-pressure function until this signal is made low again
GRF_MAC_CON1[10]	gmac_speed 1'b1: 100-Mbps 1'b0: 10-Mbps
GRF_MAC_CON1[11]	RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz
GRF_MAC_CON1[13:12]	RGMII clock selection 2'b00: 125MHz 2'b11: 25MHz 2'b10: 2.5MHz
GRF_MAC_CON1[14]	RMII mode selection 1'b1: RMII mode 1'b0: Reserved
GRF_MAC_CON0[6:0]	RGMII TX clock delayline value
GRF_MAC_CON0[13:7]	RGMII RX clock delayline value
GRF_MAC_CON0[14]	RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
GRF_MAC_CON0[15]	RGMII RX clock delayline enable 1'b1: enable 1'b0: disable

Rockchip Confidential

Chapter 45 MIPI Controller

45.1 Overview

The Display Serial Interface (DSI) is part of a group of communication protocols defined by the MIPI Alliance. The MIPI Controller is a digital core that implements all protocol functions defined in the MIPI DSI Specification. The MIPI Controller provides an interface between the system and the MIPI D-PHY, allowing the communication with a DSI-compliant display. The MIPI Controller supports one to four lanes for data transmission with MIPI D-PHY.

The MIPI Controller supports the following features:

- Compliant with MIPI Alliance standards
- Support the DPI interface color coding mappings into 24-bit Interface
 - 16 bits per pixel, configurations 1,2, and 3
 - 18 bits per pixel, configurations 1 and 2
 - 24 bits per pixel
- Programmable polarity of all DPI interface signals
- Extended resolutions beyond the DPI standard maximum resolution of 800x480 pixels:
 - Up to 2047 vertical active lines
 - Up to 63 vertical back porch lines
 - Up to 63 vertical front porch lines
 - Maximum resolution is limited by available DSI Physical link bandwidth which depends on the number of lanes and maximum speed per lane
- All commands defined in MIPI Alliance Specification for Display Command Set (DCS)
- Interface with MIPI D-PHY following PHY Protocol Interface (PPI), as defined in MIPI Alliance Specification for D-PHY
- Up to four D-PHY Data Lanes
- Bidirectional communication and escape mode support through data lane 0
- Transmission of all generic commands
- ECC and Checksum capabilities
- End of Transmission Packet(EOTP)
- Ultra Low-Power mode
- Fault recovery schemes

45.2 Block Diagram

The following diagram shows the MIPI Controller architecture.

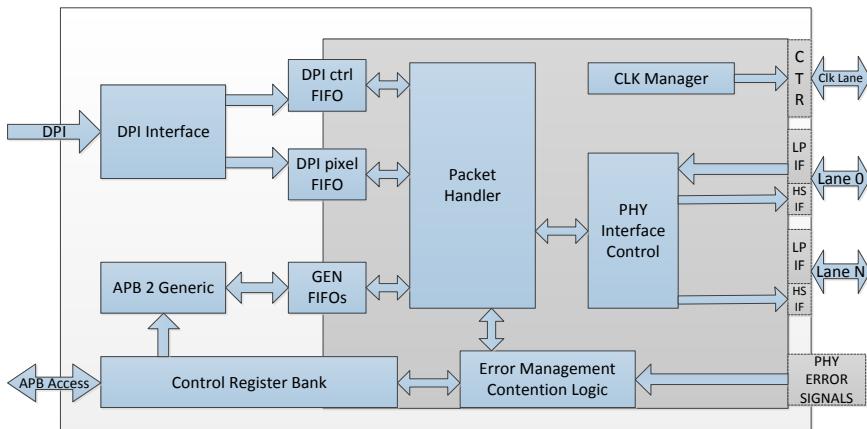


Fig. 45-1MIPI Controller architecture

The DPI interface captures the data and control signals and conveys them to a FIFO for video control signals and another one for pixel data. This data is then used to build Video packets, hen in Video mode.

The Register Bank is accessible through a standard AMBA-APB slave interface, providing access to the MIPI Controller registers for configuration and control. There is also a fully programmable interrupt generator to inform the system about certain events.

The PHY Interface Control is responsible for managing the D-PHY PPI interface. It acknowledges the current operation and enables low-power transmission/reception or a high-speed transmission. It also performs data splitting between available D-PHY lanes for high-speed transmission.

The Packet Handler schedules the activities inside the link. It performs several functions based on the interfaces that are currently DPI and the video transmission mode that is used (burst mode or non-burst mode with sync pulse or sync events). It builds long or short packet generating correspondent ECC and CRC codes. This block also performs the following functions: Packet reception, Validation of packet header by checking the ECC, Header correction and notification for single-bit errors, Termination of reception, Multiple header error notification.

The APB-to-Generic block bridges the APB operations into FIFOs holding the Generic commands. The block interfaces with the following FIFOs: Command FIFO, Write payload FIFO, Read payload FIFO.

The Error Management notifies and monitors the error conditions on the DSI link. It controls the timers used to determine if a timeout condition occurred, performing an internal soft reset and triggering an interruption notification.

45.3 Function Description

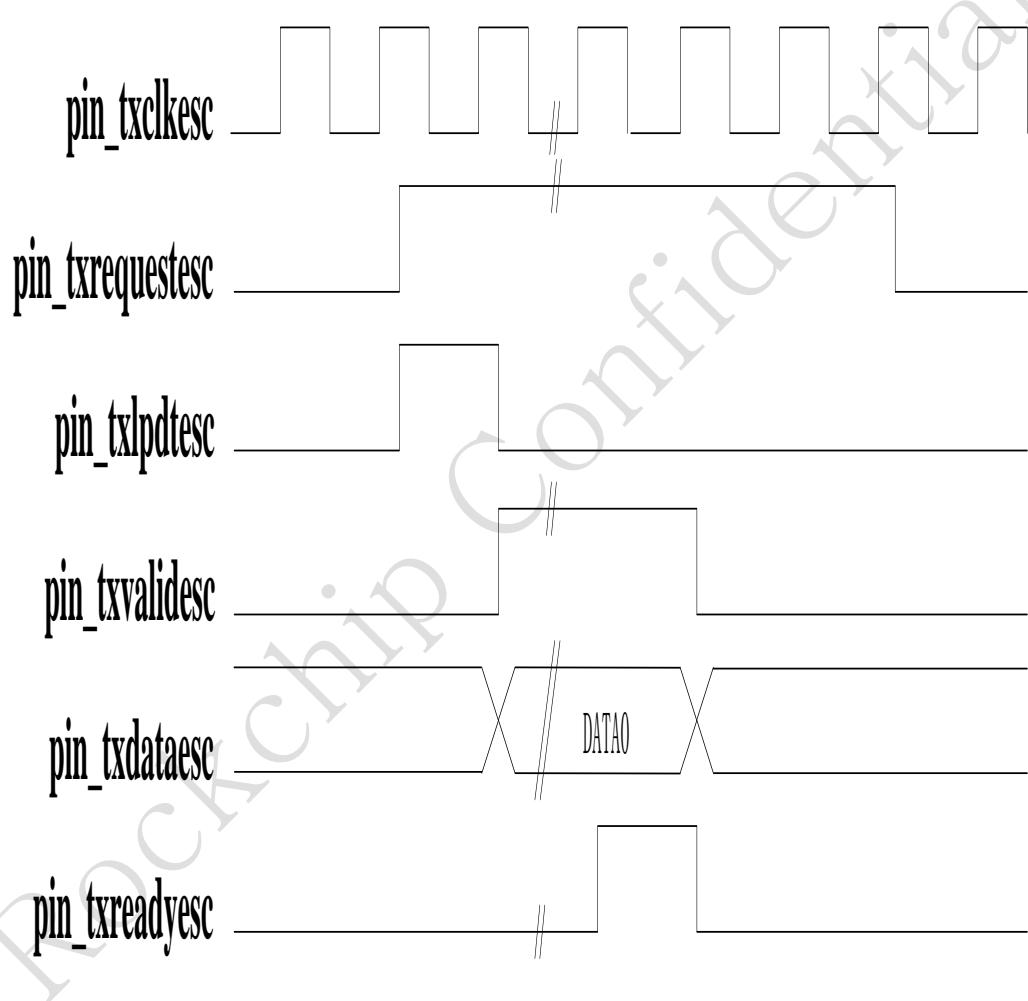
45.3.1 DPI interface function

The DPI interface follows the MIPI DPI specification with pixel data bus width up to 24 bits. It is used to transmit the information in Video mode in which the transfers from the host processor to the peripheral take the form of a real-time

pixel stream. This interface allows sending ShutDown (SD) and ColorMode (CM) commands, which are triggered directly by writing to the register of CFG_MISC_CON[2:1]. To transfer additional commands(for example, to initialize the display), use another interface such as APB Slave Generic Interface to complement the DPI interface.

The DPI interface captures the data and control signals and conveys them to the FIFO interfaces that transmit them to the DSI link. Two different streams of data are presented at the interface; video control signals and pixel data. Depending on the interface color coding, the pixel data is disposed differently throughout the dpipixdata bus. The following table shows the Interface pixel color coding.

Table 45-1Color table



The DPI interface can be configured to increase flexibility and promote correct usage of this interface for several systems. These configuration options are as follows:

- Polarity control:** All the control signals are programmable to change the polarity depending on system requirements.

After the MIPI Controller reset, DPI waits for the first VSYNC active transition to start signal sampling, including pixel data, and preventing image transmission in the middle of a frame.

If interface pixel color coding is 18 bits and the 18-bit loosely packed stream is disabled, the number of lines programmed in the pixels per lines configuration is a multiple of four. This means that in this mode, the two LSBs in the

configuration are always inferred as zero. The specification states that in this mode, the pixel line size should be a multiple of four.

45.3.2 APB Slave Generic Interface

The APB Slave interface allows the transmission of generic information in Command mode, and follows the proprietary register interface. Commands sent through this interface are not constrained to comply with the DCS specification, and can include generic commands described in the DSI specification as manufacturer-specific.

The MIPI Controller supports the transmission or write and read command mode packets as described in the DSI specification. These packets are built using the APB register access. The GEN_PLD_DATA register has two distinct functions based on the operation. Writing to this register sends the data as payload when sending a Command mode packet. Reading this register returns the payload of a read back operation. The GEN_HDR register contains the Command mode packet header type and header data. Writing to this register triggers the transmission of the packet implying that for a long Command mode packet, the packet's payload needs to be written in advance in the GEN_PLD_DATA register. The valid packets available to be transmitted through the Generic interface are as follows:

Generic Write Short Packet 0 Parameters

Generic Write Short Packet 1 Parameters

Generic Write Short Packet 2 Parameter

Generic Write Short Packet 0 Parameter

Generic Write Short Packet 1 Parameters

Generic Write Short Packet 2 Parameter

Maximum Read Packet Configuration

Generic Long Write Packet

DCS Write Short Packet 0 Parameter

DCS Write Short Packet 1 Parameter

DCS Write Short Packet 0 Parameter

DCS Write Long Packet

A set of bits in the CMD_PKT_STATUS register report the status of the FIFOs associated with APB interface support.

Generic interface packets are always transported using one of the DSI transmission modes; Video mode or Command mode. If neither of these mode are selected, the packets are not transmitted through the link and the released FIFOs eventually get overflowed.

The transfer of packets through the APB bus is based on the following conditions:

The APB protocol defines that the write and read procedure takes two clock cycles each to be executed. This means that the maximum input data rate through the APB interfaces is always half the speed of the APB clock.

The data input bus has a maximum width of 32 bits. This allows for a relation to be defined between the input APB clock frequency and maximum bi rate achievable by the APB interface.

The DSI link bit rate when using solely APB is equal to (APB clock frequency) *16 Mbps.

The bandwidth is dependent on the APB clock frequency; the available bandwidth increases with the clock frequency.

To drive the APB interface to achieve high bandwidth Command mode traffic transported by the DSI link, the MIPI Controller should operate in the Command

mode only and the APB interface should be the only data source that is currently in use. Thus, the APB interface has the entire bandwidth of the DSI link and does not share it with any another input interface source.

The memory write commands require maximum throughout from the APB interface, because they contain the most amount of data conveyed by the DSI link. While writing the packet information, first write the payload of a given packet into the payload FIFO using the GEN_PLD_DATA register. When the payload data is for the command parameters, place the first byte to be transmitted in the least significant byte position of the APB data bus.

After writing the payload, write the packet header into the command FIFO. For more information and it should follow the pixel to byte conversion organization referred in the Annexure A of the DCS specification. The follow figures show how the pixel data should be orgavized in the APB data write bus. The memory write commands are conveyed in DCS long packets. DCS long packets are encapsulated in a DSI packet. The DSI included in the diagrams. In the follow figures, the Write Memory Command can be replaced by the DCS command Write Memory Start and Write Memory Continue.

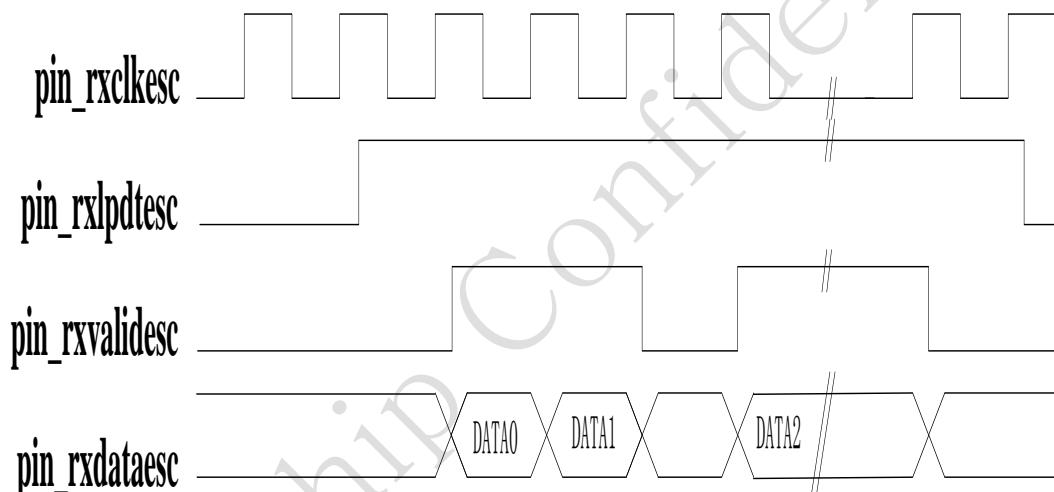


Fig. 45-224 bpp APB Pixel to Byte Organization

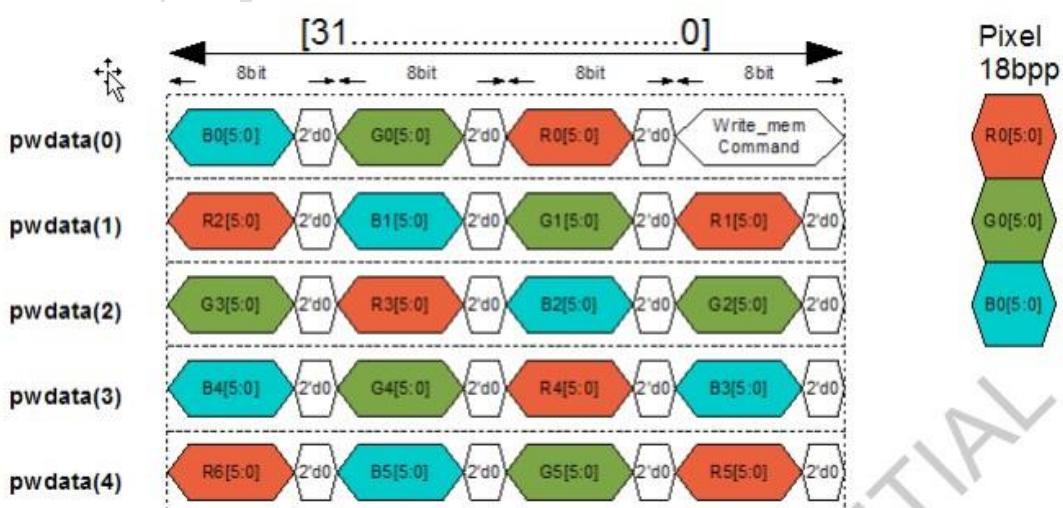


Fig. 45-318 bpp APB Pixel to Byte Organization



Fig. 45-416 bpp APB Pixel to Byte Organization

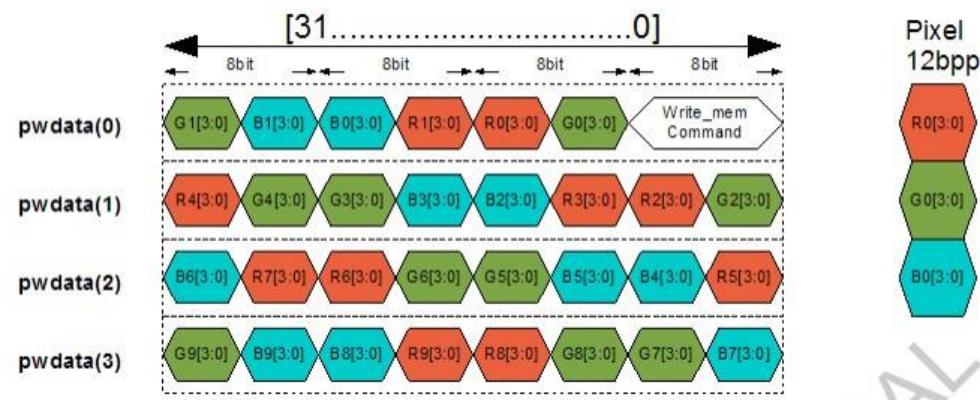


Fig. 45-512 bpp APB Pixel to Byte Organization

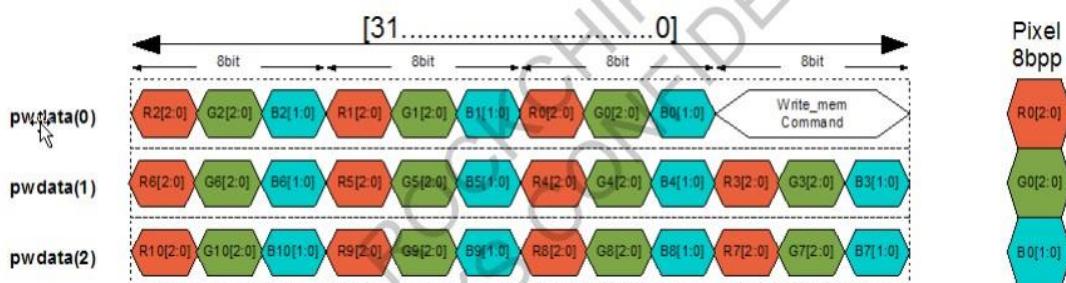


Fig. 45-68 bpp APB Pixel to Byte Organization

45.3.3 Transmission of Commands in Video Mode

The MIPI Controller supports the transmission of commands, both in high-speed and low-power, while in Video mode. The DSI controller uses Blanking or Low-Power(BLLP) periods to transmit commands inserted through the APB Generic interface. Those periods correspond to the shaded areas of the following figure.

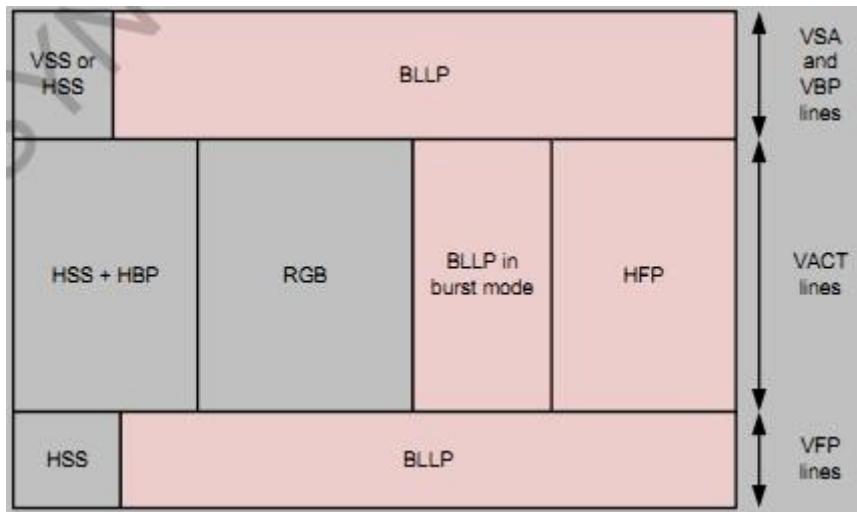


Fig. 45-7Command Transmission Periods within the Image Area

Commands are transmitted in the blanking periods after the following packet/states:

- Vertical Sync Start (VSS) packets, if the Video Sync pulses are not enabled
- Horizontal Sync End (HSE) packets, in the VSA, VBP, and VFP regions
- Horizontal Sync Start (HSS) packets, if the Video Sync pulses are not enabled in the VSA, VBP, and VFP regions
- Horizontal Active (HACT) state

Only one command is transmitted per line, even in the case of the last line of a frame but one command is possible for each line.

The MIPI Controller avoids sending commands in the last line because it is possible that the last line is shorter than the other ones. For instance, the line time (t_L) could be half a cycle longer than the t_L on the DPI interface, that is, each line in the frame taking half a cycle from time for the last line. This results in the last line being $(1/2 \text{ cycle}) * (\text{number of lines} - 1)$ shorter than t_L .

The dpicolorm and dpishutdn input signals are also able to trigger the sending of command packets. The commands are DSI data types Color Mode On, Color Mode Off, Shut Down Peripheral, and Turn on Peripheral. These commands are not sent in the VACT region. If the lpcmden bit of the VID_MODE_CFG register is 1, these commands are sent in LP mode. In LP mode, the ouvact_lpcmd_time field of the LP_CMD_TIM register is used to determine if these commands can be transmitted. It is assumed that outvact_lpcmd_time is greater than or equal to 4 bytes (number of bytes in a short packet), because the DWC_mipi_dsi_host does not transmit these commands on the last line.

If the frame_BTA_ack field is set in the VID_MODE_CFG register, a BTA is generated by DWC_mipi_dsi_host after the last line of a frame. This may coincide with a write command or a read command. In either case, the edpihalt signal is held asserted until an acknowledge has been received (control of the DSI bus is returned to the host).

If the lpcmden bit of the VID_MODE_CFG register is set to 1, the commands are sent in low-power in Video mode. In this case, it is necessary to calculate the time available, in bytes, to transmit a command in LP mode for Horizontal Front Porch (HFP), Vertical Sync Active (VSA), Vertical Back Porch (VBP), and Vertical

Front Porch(VFP) regions.

The outvact_lpcmd_time field of the LP_CMD_TIM register indicates the time available (in bytes) to transmit a command in LP mode, based on the escape clock, on a line during the VSA, VBP, and the VFP

$$\text{Outvact_lpcmd_time} = (\text{tL} - (\text{Time to transmit HSS and HSE frames} + \text{tHSA} + \text{Time to enter and leave LP mode} + \text{Time to send the D-PHY LPDT command})) / \text{escape clock period} / 8 / 2$$

Where,

tL=Line time

tHSA=Time to send a short packet (for sync events) or time of the HAS pulse (for sync pulses)

In the above equation, division by eight is done to convert the time available to bytes and division by two is done because one bit is transmitted once in every two escape clock cycles.

The outvact_lpcmd_time filed can be compared directly with the size of the command to be transmitted to determine if there is enough time to transmit the command. The maximum size of a command that can be transmitted in LP mode is limited to 255 bytes by this field. This register must be programmed to a value greater than or equal to 4 bytes for the transmission of the DCTRL commands such as shutdown and colorm in LP mode.

Consider an example with 12.6 μ s per line and assume an escape clock frequency of 15 MHz. In this case, 189 escape clock cycles are available to enter and exit LP mode and transmit command. The following are assumed:

Sync pulses are not being transmitted

Two lane byte clock ticks are required to transmit a short packet

phy_lp2hs_time=16

phy_lp2p_time=20

In this example, a 11-byte command can be transmitted as follows:

$$\text{outvact_lpcmd_time} = (12.6\mu\text{s} - (2*10\text{ ns}) - (16*10\text{ ns}) - (20*10\text{ ns}) - (8*66\text{ ns})) / 66\text{ ns} / 8 / 2 = 11 \text{ bytes}$$

The invact_lpcmd_time field of the LP_CMD_TIM register indicates the time available (in bytes) to transmite a command in LP mode (based on the escape clock) in the Vertical Active (VACT) region. This time is calculated as follows:

$$\text{Invact_lpcmd_time} = ((\text{tHFP} - \text{Time to enter and leave low-power mode} + \text{Blanking period before the HFP when in Burst mode} - \text{Time to send the D-PHY LPDT command}) / \text{escape clock period}) / 8$$

Where,

tHFP=line time-tHSA-tHBP-thACT

tHACT=vid_pkt_size*bits_per_pixel*lane_byte_clock_period / num_lanes

The invact_lpcmd_time field can be compared directly with the size of the command to be transmitted to determined if there is time to transmit the command.

Consider an example where the refresh rate is 60 Hz. The number of lines is 1320 (typical). The tL in this case is 12.6 μ s. With a lane byte clock of 100 MHz, 1260 clock ticks are availabel to transmit a single frame. If 800 ticks are used for pixel data then 460 ticks (4.6 μ s) are available for Horizontal Sync Start (HSS), HFP, and HBP. Assuming that 2.3 μ s is available for HFP and the escape clock is 15MHz, only 34 LP clock ticks are available to enter LP, transmit a command, and return from LP mode. Approximately 12 escape clock ticks are required to enter and leave LP mode. Therefore, only 1 byte could be transmitted in this

period.

A short packet (for example, generic short write) requires a minimum of 4 bytes. Therefore, in this example, commands are not sent in the VACT region. If Burst mode is enabled, more time is available to transmit commands in the VACT region. The following are assumed:

The controller is not in Burst mode

`phy_lp2hs_time=16`

`phy_lp2hs_time=16`

In this example `invact_lpcmd_time` is calculated as follows:

$$\text{Invact_lpcmd_time} = (2.3\mu\text{s} - (16*10 \text{ ns}) - (20*10 \text{ ns}) - (8*66 \text{ ns})) / 66 \text{ ns} / 8 = 2 \text{ bytes}$$

The `outvact_lpcmd_time` and `invact_lpcmd_time` fields allow a simple comparison to determine if a command can be transmitted in any of the BLLP periods.

Figure 5-21 illustrates the meaning of `invact_lpcmd_time` and `outvact_lpcmd_time`, matching them with the shaded areas and the VACT region.

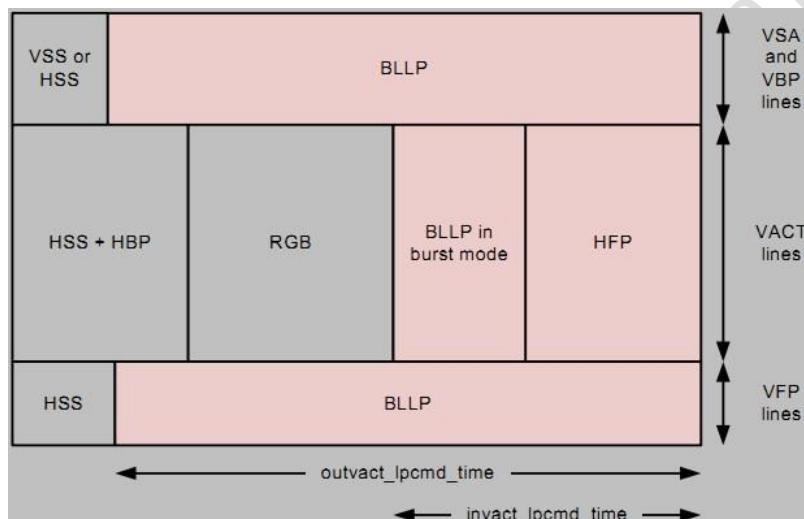


Fig. 45-8 Location of `outvact_lpcmd_time` and `invact_lpcmd_time` in the Image Area

If the `lpcmden` bit of the `VID_MODE_CFG` register is 0, the commands are sent in high_speed in Video Mode. In this case, the `DWC_mipi_dsi_host` automatically determines the area where each command can be sent and no programming or calculation is required.

On read command Transmission, the `max_rd_time` field of the `PHY_TMR_CFG` register configures the maximum amount of time required to perform a read command in lane byte clock cycles.

The maximum time required to perform a read command in Lane byte clock cycles (`max_rd_time`) = Time to transmit the read command in LP mode + Time to enter and leave LP mode + Time to return the read data packet from the peripheral device.

The time to return the read data packet from the peripheral depends on the number of bytes read and the escape clock frequency of the peripheral; not the escape clock of the host. The `max_rd_time` field is used in both HS and LP mode to determine if there is time to complete a read command in a BLLP period.

In high-speed mode (`lpcmden=0`), `max_rd_time` is calculated as follows:

$$\text{max_rd_time} = \text{phy_hs2lp_time} + \text{Time to return the read data packet from the peripheral device} + \text{phy_hs2hs_time}$$

In low-power mode (`lpcmden = 1`), `max_rd_time` is calculated as follows:
$$\text{max_rd_time} = \text{phy_hs2lp_time} + \text{LPDT command time} + \text{Read command time}$$
$$\text{in LP mode} + \text{Time to return the data read from the peripheral device} + \text{phy_lp2hs_time}$$

Where,

LPDT command time = $(8 * \text{Host escape clock period}) / \text{Lane byte clock period}$
Read command time in LP mode = $(32 * \text{host escape clock period}) / \text{lane byte clock period}$

It is recommended to keep the maximum number of bytes read from the peripheral to a minimum to have sufficient time available to issue the read commands on a line. Ensure that $\text{max_rd_time} * \text{Lane byte clock period}$ is less than `outvact_lpcmd_time * 8 * Escape clock period of the host`.

Otherwise, the read commands are serviced on the last line of a frame and the `edpihalt` signal may be asserted. If it is necessary to read a large number of parameters (> 16), increase the `max_rd_time` while the read command is being executed. When the read has completed, decrease the `max_rd_time` to a lower value.

45.4 Register Description

This section describes the control/status registers of the design.

45.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
MIPIC_VERSION	0x0000	W	0x3132302a	Version of the mi pi controller
MIPIC_PWR_UP	0x0004	W	0x00000000	Core power-up
MIPIC_CLKMGR_CFG	0x0008	W	0x00000000	Configuration of the internal clock dividers
MIPIC_DPI_VCID	0x000c	W	0x00000000	The DPI interface configuration.
MIPIC_DPI_COLOR_CODING	0x0010	W	0x00000000	
MIPIC_DPI_CFG_POL	0x0014	W	0x00000000	
MIPIC_LP_CMD_TIM	0x0018	W	0x00000000	Low-power Command Timing Configuration Register.
MIPIC_PCKHDL_CFG	0x002c	W	0x00000000	Packet handler configuration
MIPIC_GEN_VCID	0x0030	W	0x00000000	
MIPIC_MODE_CFG	0x0034	W	0x00000000	
MIPIC_VID_MODE_CFG	0x0038	W	0x00000000	Video mode configuration.
MIPIC_VID_PKT_SIZE	0x003c	W	0x00000000	
MIPIC_VID_NUM_CHUNKS	0x0040	W	0x00000000	
MIPIC_VID_NULL_SIZE	0x0044	W	0x00000000	
MIPIC_VID_HSA_TIM	0x0048	W	0x00000000	Line timing configuration.
MIPIC_VID_HBP_TIM	0x004c	W	0x00000000	
MIPIC_VID_HLINE_TIME	0x0050	W	0x00000000	
MIPIC_VID_VSA_LINES	0x0054	W	0x00000000	Vertical timing configuration.
MIPIC_VID_VBP_LINES	0x0058	W	0x00000000	
MIPIC_VID_VFP_LINES	0x005c	W	0x00000000	
MIPIC_VID_VACTIVE_LINES	0x0060	W	0x00000000	
MIPIC_EDPI_CMD_SIZE	0x0064	W	0x00000000	
MIPIC_CMD_MODE_CFG	0x0068	W	0x00000000	Command mode configuration
MIPIC_GEN_HDR	0x006c	W	0x00000000	Generic packet header configuration.
MIPIC_GEN_PLD_DATA	0x0070	W	0x00000000	Generic payload data in and out.
MIPIC_CMD_PKT_STATUS	0x0074	W	0x00000000	Command packet status

Name	Offset	Size	Reset Value	Description
MIPIC_TO_CNT_CFG	0x0078	W	0x00000000	Timeout timers configuration
MIPIC_HS_RD_TO_CNT	0x007c	W	0x00000000	
MIPIC_LP_RD_TO_CNT	0x0080	W	0x00000000	
MIPIC_HS_WR_TO_CNT	0x0084	W	0x00000000	
MIPIC_LP_WR_TO_CNT	0x0088	W	0x00000000	
MIPIC_BTA_TO_CNT	0x008c	W	0x00000000	
MIPIC_LPCLK_CTRL	0x0094	W	0x00000000	
MIPIC_PHY_TMR_LPC_LK_CFG	0x0098	W	0x00000000	
MIPIC_PHY_TMR_CFG	0x009c	W	0x00000000	D-PHY timing configuration
MIPIC_PHY_RSTZ	0x00a0	W	0x00000000	D-PHY reset control
MIPIC_PHY_IF_CFG	0x00a4	W	0x00000000	D-PHY interface configuration
MIPIC_PHY_ULPS_CTRL	0x00a8	W	0x00000000	D-PHY PPI interface control
MIPIC_PHY_TX_TRIGERS	0x00ac	W	0x00000000	
MIPIC_PHY_STATUS	0x00b0	W	0x00000000	D-PHY PPI status interface
MIPIC_RESERVED3	0x00b4	W	0x00000000	Reserved
MIPIC_RESERVED4	0x00b8	W	0x00000000	Reserved
MIPIC_ERROR_ST0	0x00bc	W	0x00000000	Interrupt status register 0
MIPIC_ERROR_ST1	0x00c0	W	0x00000000	Interrupt status register 1
MIPIC_MSK0	0x00c4	W	0x00000000	Masks the interrupt generation triggerd by the ERROR_ST0 reg
MIPIC_MSK1	0x00c8	W	0x00000000	Masks the interrupt generation triggerd by the ERROR_ST1 reg

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

45.4.2 Detail Register Description

MIPIC_VERSION

Address: Operational Base + offset (0x0000)

Version of the mihi controller

Bit	Attr	Reset Value	Description
31:0	RO	0x3132302a	version indicates the version of the mihi_controller

MIPIC_PWR_UP

Address: Operational Base + offset (0x0004)

Core power-up

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shutdownz This bit indicates the core power-up or the reset 0-Reset 1-Power-up

MIPIC_CLKMGR_CFG

Address: Operational Base + offset (0x0008)

Configuration of the internal clock dividers

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	TO_CLK_DIVISION This field indicates the division factor for the Time Out clock used as the timing unit in the configuration of HS to LP and LP to HS transition error.
7:0	RW	0x00	TX_ESC_CLK_DIVISION Field0000 Abstract This field indicates the division factor for the TX_Escape clock source(lanebyteclk).The value 0 and 1 stop the TX_ESC clock generation

MIPIC_DPI_VCID

Address: Operational Base + offset (0x000c)

The DPI interface configuration.

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	dpi_vid This field configures the DPI virtual channel id that is indexed to the Video mode packets.

MIPIC_DPI_COLOR_CODING

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	en18_loosely When set to 1, this bit enables 18 loosely packed pixel stream.
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	dpi_color_coding This field configures the DPI color coding as follows: 000:16bit configuration 1 001:16bit configuration 2 010:16bit configuration 3 011:18bit configuration 1 100:18bit configuration 2 101,110, and 111:24bits

MIPIC_DPI_CFG_POL

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	colorm_active_low When set to 1, this bit configures the color mode pin as active low
3	RW	0x0	shutd_active_low When set to 1, this bit configures the shut down pin as active low
2	RW	0x0	hsync_active_low When set to 1, this bit configures the horizontal synchronism pin as active low.
1	RW	0x0	vsync_active_low When set to 1, this bit configures the vertical synchronism pin as active low
0	RW	0x0	dataen_active_low When set to 1, this bit configures the data enable pin as active low

MIPIC_LP_CMD_TIM

Address: Operational Base + offset (0x0018)

Low-power Command Timing Configuration Register.

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	outvact_lpcmd_time outside VACT region command time. This field configures the time available to transmit a command in low-power mode. The time value is expressed in a number of bytes format. The number of bytes represents the maximum size of a packet that can fit in a line during the VSA, VBP, and VFP region. This field must be configured with a value greater than or equal to four bytes to allow the transmission of the DCTRL commands such as shutdown and colorm in low-power mode.
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	invact_lpcmd_time Inside VACT region command time.This field configures the time available to transmit a command in low-power mode.The time value is expressed in a number of bytes format.The number of bytes represents the maximum size of the packet that can fit a line during the VACT region.

MIPIC_PCKHDL_CFG

Address: Operational Base + offset (0x002c)

Packet handler configuration

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	en_CRC_rx When set to 1,this bit enables the CRC reception and error reporting
3	RW	0x0	en_ECC_rx When set to 1,this bit enables the ECC reception,error correction, and reporting
2	RW	0x0	en_BTA When set to 1,this bit enables the Bus Turn-Around(BTA) request.
1	RW	0x0	en_EOTP_rx When set to 1,this bit enables the EOTP reception
0	RW	0x0	en_EOTP_tx When set to 1,this bit enables the EOTP transmission

MIPIC_GEN_VCID

Address: Operational Base + offset (0x0030)

Packet handler configuration

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	gen_vid_rx This field indicates the Generic interface read-back virtual channel identification

MIPIC_MODE_CFG

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	en_video_mode When set to 1,this bit enables the DPI Video mode transmission.

MIPIC_VID_MODE_CFG

Address: Operational Base + offset (0x0038)

Video mode configuration.

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	lpcmden When set to 1, this bit enables the command transmission only in low-power mode
14	RW	0x0	frame_BTA_ack When set to 1, this bit enables the request for an acknowledge response at the end of a frame
13	RW	0x0	en_lp_hfp When set to 1, this bit enables the return to low-power inside the HFP period when timing allows.
12	RW	0x0	en_lp_hbp When set to 1, this bit enables the return to low-power inside the HBP period when timing allows.
11	RW	0x0	en_lp_vact When set to 1, this bit enables the return to low-power inside the VACT period when timing allows.
10	RW	0x0	en_lp_vfp When set to 1, this bit enables the return to low-power inside the VFP period when timing allows.
9	RW	0x0	en_lp_vbp When set to 1, this bit enables the return to low-power inside the VBP period when timing allows.
8	RW	0x0	en_lp_vsa When set to 1, this bit enables the return to low-power inside the VSA period when timing allows.
7:2	RO	0x0	reserved
1:0	RW	0x0	vid_mode_type This field indicates the video mode transmission type as follows: 00: Non-burst with sync pulses 01: Non-burst with sync events 10 and 11: Burst with sync pulses

MIPIC_VID_PKT_SIZE

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:0	RW	0x0000	vid_pkt_size This field configures the number of pixels on a single video packet. If you use the 18-bit mode and do not enable loosely packed stream, this value must be a multiple of 4.

MIPIC_VID_NUM_CHUNKS

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	num_chunks This field configures the number of chunks to be transmitted during a line period (a chunk is a video packet or a null packet)

MIPIC_VID_NULL_SIZE

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	null_pkt_size This field configures the number of bytes in a null packet

MIPIC_VID_HSA_TIME

Address: Operational Base + offset (0x0048)

Line timing configuration.

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	hsa_time This field configures the Horizontal Synchronization Active period in lane byte clock cycles.

MIPIC_VID_HBP_TIME

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	hbp_time This field configures the Horizontal Back Porch period in lane byte clock cycles

MIPIC_VID_HLINE_TIME

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:0	RW	0x0000	hline_time This field configures the size of the total lines counted in lane byte cycles.

MIPIC_VID_VSA_LINES

Address: Operational Base + offset (0x0054)

Vertical timing configuration.

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	vsa_lines This field configures the Vertical Synchronism Active period measured in number of horizontal lines.

MIPIC_VID_VBP_LINES

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	vbp_lines This field configures the Vertical Back Porch period measured in horizontal lines.

MIPIC_VID_VFP_LINES

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	vfp_lines This field configures the Vertical Front Porch period measured in horizontal lines.

MIPIC_VID_VACTIVE_LINES

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:0	RW	0x0000	v_active_line This field configures the Vertical Active period measured in horizontal lines.

MIPIC_EDPI_CMD_SIZE

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	edpi_allowed_cmd_size This field configures the maximum allowed size for an eDPI write memory command, measured in pixels. Automatic partitioning of data obtained from eDPI is permanently enabled.

MIPIC_CMD_MODE_CFG

Address: Operational Base + offset (0x0068)

Command mode configuration

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	max_rd_pkt_size This bit configures the maximum read packet size command transmission type: 0:High-speed 1:Low-power
23:20	RO	0x0	reserved
19	RW	0x0	dcs_lw_tx This bit configures the DCS long write packet command transmission type: 0:High-speed 1:Low-power
18	RW	0x0	dcs_sr_0p_tx This bit configures the DCS short read packet with zero parameter command transmission type: 0:High-speed 1:Low-power
17	RW	0x0	dcs_sw_1p_tx This bit configures the DCS short write packet with one parameter command transmission type: 0:High-speed 1:Low-power
16	RW	0x0	dcs_sw_0p_tx This bit configures the DCS short write packet with zero parameter command transmission type: 0:High-speed 1:Low-power
15	RO	0x0	reserved
14	RW	0x0	gen_lw_tx This bit configures the Generic long write packet command 0:High-speed 1:Low-power

Bit	Attr	Reset Value	Description
13	RW	0x0	gen_sr_2p_tx This bit configures the Generic short read packet with two parameter command transmission type: 0:High-speed 1:Low-power
12	RW	0x0	gen_sr_1p_tx This bit configures the Generic short read packet with one parameter command transmission type: 0:High-speed 1:Low-power
11	RW	0x0	gen_sr_0p_tx This bit configures the Generic short read packet with zero parameter command transmission type: 0:High-speed 1:Low-power
10	RW	0x0	gen_sw_2p_tx This bit configures the Generic short write packet with two parameter command transmission type: 0:High-speed 1:Low-power
9	RW	0x0	gen_sw_1p_tx This bit configures the Generic short write packet with one parameter command transmission type: 0:High-speed 1:Low-power
8	RW	0x0	gen_sw_0p_tx This bit configures the Generic short write packet with zero parameter command transmission type: 0:High-speed 1:Low-power
7:2	RO	0x0	reserved
1	RW	0x0	ack_rqst_en When
0	RW	0x0	tear_fx_en When set to 1, this bit enables the tearing effect acknowledge request

MIPIC_GEN_HDR

Address: Operational Base + offset (0x006c)

Generic packet header configuration.

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved

Bit	Attr	Reset Value	Description
23:16	RW	0x00	gen_WC_MSbyte This field configures the most significant byte of the header packet's Word count for long packets or data 1 for short packets.
15:8	RW	0x00	gen_WC_LSbyte This field configures the least significant byte of the header packet's Word count for long packets or data 0 for short packets.
7:6	RW	0x0	gen_VC This field configures the virtual channel id of the header packet.
5:0	RW	0x00	gen_DT This field configures the packet data type of the header packet

MIPIC_GEN_PLD_DATA

Address: Operational Base + offset (0x0070)

Generic payload data in and out.

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gen_pld_b4 This field indicates byte 4 of the packet payload.
23:16	RW	0x00	gen_pld_b3 This field indicates byte 3 of the packet payload.
15:8	RW	0x00	gen_pld_b2 This field indicates byte 2 of the packet payload.
7:0	RW	0x00	gen_pld_b1 This field indicates byte 1 of the packet payload.

MIPIC_CMD_PKT_STATUS

Address: Operational Base + offset (0x0074)

Command packet status

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	reserved reserved
6	RW	0x0	gen_rd_cmd_busy This bit is set when a read command is issued and cleared when the entire response is stored in the FIFO
5	RW	0x0	gen_pld_r_full This bit indicates the full status of the generic read payload FIFO Value after reset:0x0

Bit	Attr	Reset Value	Description
4	RO	0x0	gen_pld_r_empty This bit indicates the empty status of the generic read payload FIFO Value after reset:0x1
3	RO	0x0	gen_pld_w_full This bit indicates the full status of the generic write payload FIFO Value after reset:0x0
2	RO	0x0	gen_pld_w_empty This bit indicates the empty status of the generic write payload FIFO Value after reset:0x1
1	RO	0x0	gen_cmd_full This bit indicates the full status of the generic command FIFO Value after reset:0x0
0	RO	0x0	gen_cmd_empty This bit indicates the empty status of the generic command FIFO Value after reset:0x1

MIPIC_TO_CNT_CFG

Address: Operational Base + offset (0x0078)

Timeout timers configuration

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	hstx_to_cnt This field configures the timeout counter that triggers a high-speed transmission timeout contention detection(measured in TO_CLK_DIVISION cycles)
15:0	RW	0x0000	lprx_to_cnt This field configures the timeout counter that triggers a low-power reception timeout contention detection(measured in TO_CLK_DIVISION cycles)

MIPIC_HS_RD_TO_CNT

Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	hs_rd_to_cnt This field sets a period for which the MIPI Controller keeps the link still,after sending a high-speed read operation.This period is measured in cycles of lanebyteclk.The counting starts when the D-PHY enters the Stop state and causes no interrupts.

MIPIC_LP_RD_TO_CNT

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	lp_rd_to_cnt This field sets a period for which MIPI Controller keeps the link still, after sending a low-power read operation. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

MIPIC_HS_WR_TO_CNT

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	presp_to_mode When set to 1, this bit ensures that the peripheral response timeout caused by hs_wr_to_cnt is used only once per eDPI frame, when both the following conditions are met: .dpivsync_edpiwms has risen and fallen .packets originated from eDPI have been transmitted and its FIFO is empty again.
23:16	RO	0x0	reserved
15:0	RW	0x0000	hs_wr_to_cnt This field sets a period for which the MIPI Controller keeps the link inactive after sending a high-speed write operation. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

MIPIC_LP_WR_TO_CNT

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0	lp_wr_to_cnt This field sets a period for which the DSI Controller keeps the link still, after sending a low-power write operation. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

MIPIC_BTA_TO_CNT

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	bta_to_cnt This field sets a period for which the DSI Controller keeps the link still, after completing a Bus Turn-Around. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

MIPIC_LPCLK_CTRL

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	auto_clklane_ctrl This bit enables the automatic mechanism to stop providing clock in the clock lane when time allows.
0	RW	0x0	phy_txrequestclkhs This bit controls the D-PHY PPI tx requestclkhs signal

MIPIC_PHY_TMR_LPCLK_CFG

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:16	RW	0x000	phy_hs2lp_time This field configures the maximum time that the PHY takes to go from high-speed to low-power transmission measured in lane byte clock cycles.(clock lane)
15:10	RO	0x0	reserved
9:0	RW	0x000	phy_lp2hs_time This field configures the maximum time that the PHY takes to go from low-power to high-speed transmission measured in lane byte clock cycles.(clock lane)

MIPIC_PHY_TMR_CFG

Address: Operational Base + offset (0x009c)

D-PHY timing configuration

Bit	Attr	Reset Value	Description
31:24	RW	0x00	phy_hs2lp_time This field configures the maximum time that the PHY takes to go from high-speed to low-power transmission measured in lane byte clock cycles.

Bit	Attr	Reset Value	Description
23:16	RW	0x00	phy_lp2hs_time This field configures the maximum time that the PHY takes to go from low-power to high-speed transmission measured in lane byte clock cycles.
15	RW	0x0	reserved reserved for future use
14:0	RW	0x0000	max_rd_time This field configures the maximum time required to perform a read command in lane byte clock cycles. This register can only be modified when read commands are not in progress.

MIPIC_PHY_RSTZ

Address: Operational Base + offset (0x00a0)

D-PHY reset control

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	phy_enableclk When set to 1, this bit enables the D-PHY Clock Lane Module
1	RW	0x0	reserved1
0	RW	0x0	reserved

MIPIC_PHY_IF_CFG

Address: Operational Base + offset (0x00a4)

D-PHY interface configuration

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	phy_stop_wait_time This field configures the minimum wait period to request a high-speed transmission after the Stop state is accounted in clock lane cycles.
7:2	RO	0x0	reserved
1:0	RW	0x0	n_lanes This field configures the number of active data lanes: 00:One data lane(lane 0) 01:Two data lane(lanes 0 and 1) 10:Three data lanes(lanes 0,1, and 2) 11:Four data lanes(lanes 0,1,2, and 3)

MIPIC_PHY_ULPS_CTRL

Address: Operational Base + offset (0x00a8)

D-PHY PPI interface control

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	phy_txexitulpslan ULPS mode Exit on all active data lanes
2	RW	0x0	phy_txrequlpslan ULPS mode Request on all active data lanes
1	RW	0x0	phy_txexitulpsclk ULPS mode Exit on clock lane
0	RW	0x0	phy_txrequlpsclk ULPS mode Request on clock lane

MIPIC_PHY_TX_TRIGGER

Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	phy_tx_triggers This field controls the trigger transmissions.

MIPIC_PHY_STATUS

Address: Operational Base + offset (0x00b0)

D-PHY PPI status interface

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	ulpsactivenot3lane This bit indicates the status of ulpsactivenot3lane D-PHY signal
11	RO	0x0	phystopstate3lane This bit indicates the status of phystopstate3lane D-PHY signal
10	RO	0x0	ulpsactivenot2lane This bit indicates the status of ulpsactivenot2lane D-PHY signal
9	RO	0x0	phystopstate2lane This bit indicates the status of phystopstate2lane D-PHY signal
8	RO	0x0	ulpsactivenot1lane This bit indicates the status of ulpsactivenot1lane D-PHY signal
7	RO	0x0	phystopstate1lane This bit indicates the status of phystopstate1lane D-PHY signal
6	RW	0x0	rxulpsesc0lane This bit indicates the status of rxulpsesc0lane D-PHY signal
5	RO	0x0	ulpsactivenot0lane This bit indicates the status of ulpsactivenot0lane D-PHY signal
4	RO	0x0	phystopstate0lane This bit indicates the status of phystopstate0lane D-PHY signal

Bit	Attr	Reset Value	Description
3	RO	0x0	phyulpsactivenotclk This bit indicates the status of phyulpsactivenotclk D-PHY signal
2	RO	0x0	phystopstateclklane This bit indicates the status of phystopstateclklane D-PHY signal
1	RO	0x0	phydirection This bit indicates the status of phydirection D-PHY signal
0	RO	0x0	phylock This bit indicates the status of phylock D-PHY signal

MIPIC_RESERVED3

Address: Operational Base + offset (0x00b4)

Reserved

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	Reserved

MIPIC_RESERVED4

Address: Operational Base + offset (0x00b8)

Reserved

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	Reserved

MIPIC_ERROR_ST0

Address: Operational Base + offset (0x00bc)

Interrupt status register 0

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	RO	0x0	dphy_errors_4 This bit indicates LP1 contention error ErrContentionLP1 from Lane 0
19	RO	0x0	dphy_errors_3 This bit indicates LP0 contention error ErrContentionLP0 from Lane 0
18	RO	0x0	dphy_errors_2 This bit indicates control error ErrControl from Lane 0
17	RO	0x0	dphy_errors_1 This bit indicates ErrSyncEsc low-power data transmission synchronization error from Lane 0
16	RO	0x0	dphy_errors_0 This bit indicates ErrEsc escape entry error from Lane 0

Bit	Attr	Reset Value	Description
15	RO	0x0	ack_with_err_15 This bit retrieves the DSI protocol violation from the Display Acknowledge error report
14	RO	0x0	ack_with_err_14 This bit retrieves the reserved(specific to device) from the Display Acknowledge error report
13	RO	0x0	ack_with_err_13 This bit retrieves the invalid transmission length from the Display Acknowledge error report
12	RO	0x0	ack_with_err_12 This bit retrieves the DSI VC ID Invalid from the Display Acknowledge error report
11	RO	0x0	ack_with_err_11 This bit retrieves the not recognized DSI data type from the Display Acknowledge error report
10	RO	0x0	ack_with_err_10 This bit retrieves the checksum error(long packet only) from the Display Acknowledge error report
9	RO	0x0	ack_with_err_9 This bit retrieves the ECC error,multi-bit(detected and corrected) from the Display Acknowledge error report
8	RO	0x0	ack_with_err_8 This bit retrieves the ECC error,single-bit(detected and corrected) from the Display Acknowledge error report
7	RO	0x0	ack_with_err_7 This bit retrieves the reserved(specific to device) error from the Display Acknowledge error report
6	RO	0x0	ack_with_err_6 This bit retrieves the False Control error from the Display Acknowledge error report
5	RO	0x0	ack_with_err_5 This bit retrieves the HS Receive Timeout error from the Display Acknowledge error report
4	RO	0x0	ack_with_err_4 This bit retrieves the LP Transmit Sync error error from the Display Acknowledge error report
3	RO	0x0	ack_with_err_3 This bit retrieves the Escape Mode Entry command error from the Display Acknowledge error report
2	RO	0x0	ack_with_err_2 This bit retrieves the EoT sync error from the Display Acknowledge error report

Bit	Attr	Reset Value	Description
1	RO	0x0	ack_with_err_1 This bit retrieves the SoT Sync error from the Display Acknowledge error report
0	RO	0x0	ack_with_err_0 This bit retrieves the SoT error from the Display Acknowledge error report

MIPIC_ERROR_ST1

Address: Operational Base + offset (0x00c0)

Interrupt status register 1

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	gen_pld_recv_err This bit indicates that during a generic interface packet read back, the payload FIFO becomes full and the received data is corrupted.
11	RO	0x0	gen_pld_rd_err This bit indicates that during a DCS read data, the payload FIFO becomes empty and the data sent to the interface is corrupted
10	RO	0x0	gen_pld_send_err This bit indicates that during a Generic interface packet build, the payload FIFO becomes empty and corrupt data is sent.
9	RO	0x0	gen_pld_wr_err This bit indicates that the system tried to write a payload data through the Generic interface and the FIFO is full. Therefore, the command is not written.
8	RO	0x0	gen_cmd_wr_err This bit indicates that the system tried to write a command through the Generic interface and the FIFO is full. Therefore, the command is not written.
7	RO	0x0	dpi_pld_wr_err This bit indicates that during a DPI pixel line storage, the payload FIFO becomes full and the data stored is corrupted.
6	RO	0x0	eopt_err This bit indicates that the EOTP packet is not received at the end of the incoming peripheral transmission.
5	RO	0x0	pkt_size_err This bit indicates that the packet size error is detected during the packet reception.
4	RO	0x0	crc_err This bit indicates that the CRC error is detected in a received packet.

Bit	Attr	Reset Value	Description
3	RO	0x0	ecc_multi_err This bit indicates that the ECC multiple error is detected and corrected in a received packet.
2	RO	0x0	ecc_single_err This bit indicates that the ECC single error is detected and corrected in a received packet.
1	RO	0x0	to_lp_rx This bit indicates that the low-power reception timeout counter reached the end and contention detection is detected.
0	RO	0x0	to_hs_tx This bit indicates that the high-speed transmission timeout counter reached the end and contention detection is detected.

MIPIC_MSK0

Address: Operational Base + offset (0x00c4)

Masks the interrupt generation triggered by the ERROR_ST0 reg

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	RW	0x0	dphy_errors_4 This bit indicates LP1 contention error ErrContentionLP1 from Lane 0
19	RW	0x0	dphy_errors_3 This bit indicates LP0 contention error ErrContentionLP0 from Lane 0
18	RW	0x0	dphy_errors_2 This bit indicates control error ErrControl from Lane 0
17	RW	0x0	dphy_errors_1 This bit indicates ErrSyncEsc low-power data transmission synchronization error from Lane 0
16	RW	0x0	dphy_errors_0 This bit indicates ErrEsc escape entry error from Lane 0
15	RW	0x0	ack_with_err_15 This bit retrieves the DSI protocol violation from the Display Acknowledge error report
14	RW	0x0	ack_with_err_14 This bit retrieves the reserved(specific to device) from the Display Acknowledge error report
13	RW	0x0	ack_with_err_13 This bit retrieves the invalid transmission length from the Display Acknowledge error report
12	RW	0x0	ack_with_err_12 This bit retrieves the DSI VC ID Invalid from the Display Acknowledge error report

Bit	Attr	Reset Value	Description
11	RW	0x0	ack_with_err_11 This bit retrieves the not recognized DSI data type from the Display Acknowledge error report
10	RW	0x0	ack_with_err_10 This bit retrieves the checksum error(long packet only) from the Display Acknowledge error report
9	RW	0x0	ack_with_err_9 This bit retrieves the ECC error,multi-bit(detected and corrected) from the Display Acknowledge error report
8	RW	0x0	ack_with_err_8 This bit retrieves the ECC error,single-bit(detected and corrected) from the Display Acknowledge error report
7	RW	0x0	ack_with_err_7 This bit retrieves the reserved(specific to device) error from the Display Acknowledge error report
6	RW	0x0	ack_with_err_6 This bit retrieves the False Control error from the Display Acknowledge error report
5	RW	0x0	ack_with_err_5 This bit retrieves the HS Receive Timeout error from the Display Acknowledge error report
4	RW	0x0	ack_with_err_4 This bit retrieves the LP Transmit Sync error from the Display Acknowledge error report
3	RW	0x0	ack_with_err_3 This bit retrieves the Escape Mode Entry command error from the Display Acknowledge error report
2	RW	0x0	ack_with_err_2 This bit retrieves the EoT sync error from the Display Acknowledge error report
1	RW	0x0	ack_with_err_1 This bit retrieves the SoT Sync error from the Display Acknowledge error report
0	RW	0x0	ack_with_err_0 This bit retrieves the SoT error from the Display Acknowledge error report

MIPIC_MSK1

Address: Operational Base + offset (0x00c8)

Masks the interrupt generation triggered by the ERROR_ST1 reg

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RO	0x0	gen_pld_recv_err This bit indicates that during a generic interface packet read back, the payload FIFO becomes full and the received data is corrupted.
11	RO	0x0	gen_pld_rd_err This bit indicates that during a DCS read data, the payload FIFO becomes empty and the data sent to the interface is corrupted
10	RO	0x0	gen_pld_send_err This bit indicates that during a Generic interface packet build, the payload FIFO becomes empty and corrupt data is sent.
9	RO	0x0	gen_pld_wr_err This bit indicates that the system tried to write a payload data through the Generic interface and the FIFO is full. Therefore, the command is not written.
8	RO	0x0	gen_cmd_wr_err This bit indicates that the system tried to write a command through the Generic interface and the FIFO is full. Therefore, the command is not written.
7	RO	0x0	dpi_pld_wr_err This bit indicates that during a DPI pixel line storage, the payload FIFO becomes full and the data stored is corrupted.
6	RO	0x0	eopt_err This bit indicates that the EOTP packet is not received at the end of the incoming peripheral transmission.
5	RO	0x0	pkt_size_err This bit indicates that the packet size error is detected during the packet reception.
4	RO	0x0	crc_err This bit indicates that the CRC error is detected in a received packet.
3	RO	0x0	ecc_multi_err This bit indicates that the ECC multiple error is detected and corrected in a received packet.
2	RO	0x0	ecc_single_err This bit indicates that the ECC single error is detected and corrected in a received packet.
1	RO	0x0	to_lp_rx This bit indicates that the low-power reception timeout counter reached the end and contention detection is detected.
0	RO	0x0	to_hs_tx This bit indicates that the high-speed transmission timeout counter reached the end and contention detection is detected.

45.5 Application Notes

Low Power Mode is a special feature for D-PHY. You can control this function by using proper registers from the Innosilicon D-PHY with few operations. The following is a step by step instruction for low power mode in and out.

Perform the following steps to configure the DPI packet transmission:

Step1:Global configuration:

Configure n_lanes (PHY_IF_CFG-[1:0]) to define the number of lanes in which the controller has to perform high-speed transmissions.

Step2:Configure the DPI Interface to define how the DPI interface interacts with the controller.

Configure dpi_vid (DPI_CFG-[1:0]): This field configures the virtual channel that the packet generated by the DPI interface is indexed to.

Configure dpi_color_coding (DPI_CFG-[4:2]): This field configures the bits per pixels that the interface transmits and also the variant configuration of each bpp. If you select 18 bpp, and the Enable_18_loosely_packed is not active, the number of pixels per line should be a multiple of four.

Configure dataen_active_low (DPI_CFG-[5]): This bit configures the polarity of the dpidataen signal and enables if it is active low.

Configure vsync_active_low(DPI_CFG-[6]): This bit configures the polarity of the dpivsync signal and enables if it is active low.

Configure vsync_active_low(DPI_CFG-[7]): This bit configures the polarity of the dpivsync signal and enables if it is active low.

Configure vsync_active_low(DPI_CFG-[8]): This bit configures the polarity of the dpishutdn signal and enables if it is active low.

Configure vsync_active_low(DPI_CFG-[9]): This bit configures the polarity of the dpicolorm signal and enables if it is active low.

Configure en_18_loosely(DPI_CFG-[10]): This bit configures if the pixel packing is done loosely or packed when dpi_color_coding is 18 bpp. This bit enables loosely packing.

Step3: Select the Video Transmission Mode to define how the processor requires the video line to be transported through the DSI link.

Configure low-power transitions (VID_MODE_CFG-[8:3]): This defines the video line to be transported through the DSI link.

Configure low-power transitions (VID_MODE_CFG-[8:3]): This defines the video periods which are permitted to go to low-power if there is available time to do so.

Configure frame_BTA_ack (VID_MODE_CFG-[11]): This specifies if the controller should request the peripheral acknowledge message at the end of frames.

Burst mode: In this mode, the entire active pixel line is buffered into a FIFO and transmitted in a single packed with no interruptions. This transmission mode requires that the DPI Pixel FIFO has the capacity to store a full line of active pixel data inside it. This mode is optimally used if the difference between pixel required bandwidth and DSI link bandwidth is very different. This enables the DWC_mipi_dsi_host to quickly dispatch the entire active video line in a single burst of data and then return to low-power mode.

Configure the register field vid_mode_type (VID_MODE_CFG-[10]), num_chunks (VID_PKT_CFG-[20:11]), and null_pkt_size

(VID_PKT_CFG-[30:21]) are automatically ignored by the DWC_mipi_dsi_host.

Non-Burst mode: In this mode, the processor uses the partitioning properties of the DWC_mipi_dsi_host to divide the video line transmission into several DSI packets. This is done to match the pixel required bandwidth with the DSI link bandwidth. With this mode, the controller configuration does not require a full line of pixel data to be stored inside the DPI Pixel FIFO. It requires only the content of one video packet.

Configure the vid_mode_type field (VID_MODE_CFG-[2:1]) with 2'b0x.

Configure the vid_mode_type field (VID_MODE_CFG-[2:1]) with 2'b00x to enable the transmission of sync pulses.

Configure the vid_mode_type field (VID_MODE_CFG-[2:1]) with 2'b01 to enable the transmission of sync events.

Configure the vid_mode_type field (VID_MODE_CFG-[10:0]) with the number of pixels to be transmitted in a single packet.

Configure the en_multi_pkt field (VID_MODE_CFG-[9]) to enable the division of the active video transmission into more than one packet.

Configure the num_chunks field (VID_MODE_CFG-[20:11]) with the number of video chunks that the active video transmission is divided into.

Configure the en_null_pkt field (VID_MODE_CFG-[10]) to enable the insertion of null packets between video packets.

The field is effective only when en_multi_pkt field is activated, otherwise the controller ignores it and does not send the null packets.

Configure the null_pkt_size field (VID_MODE_CFG-[30:21]) with the actual size of the inserted null packet.

Step4: Define the DPI Horizontal timing configuration as follows:

Configure the hline_time field (TMR_LINE_CFG-[31:18]) with the time taken by a DPI video line accounted in Clock Lane bytes clock cycles (for a clock lane at 500 MHz the Lane byte clock period is 8 ns). When the DPI clock and Clock Lane clock are not multiples, the hline_time is a result of a round of a number. If the DWC_mipi_dsi_host is configured to go to low-power, it is possible that the error included in a line is incremented with the next one. At the end of several lines, the DWC_mipi_dsi_host can have a number of errors that can cause a malfunction of the video transmission.

Configure the hsa_time field (TMR_LINE_CFG-[8:0]) with the time taken by a DPI Horizontal Sync Active period accounted in Clock Lane byte clock cycles (normally a period of 8ns).

Configure the hbp_time field (TMR_LINE_CFG-[17:9]) with the time taken by a DPI Horizontal Sync Active period accounted in Clock Lane byte clock cycles (normally a period of 8ns). Special attention should be given to the calculation of this parameter.

Step5: Define the Vertical line configuration:

Configure the vsa_lines field (VTIMING_CFG-[3:0]) with the number of lines existing in the DPI Vertical Sync Active period.

Configure the vbp_lines field (VTIMING_CFG-[9:4]) with the number of lines existing in the DPI Vertical Back Porch period.

Configure the vfp_lines field (VTIMING_CFG-[15:10]) with the number of lines existing in the DPI Vertical Front Porch period.

Configure the v_active_lines field (VTIMING_CFG-[26:16]) with the number of lines existing in the DPI Vertical Active period.

Chapter 46 MIPI D-PHY

46.1 Overview

The MIPI D-PHY integrates a MIPI® V1.0 compatible PHY that supports up to 1GHz high speed data receiver, plus a MIPI® low-power low speed transceiver that supports data transfer in the bi-directional mode. It supports the full specifications described in V1.0 of the D-PHY spec. The D-PHY is built in with a standard digital interface to talk to MIPI Host controller. The architecture supports connection of multiple data lanes in parallel – up to 4 data lanes can be connected to increase the total through-put, customizable to user determined configurations. The MIPI D-PHY supports the electrical portion of MIPI D-PHY V1.0 standard, covering all transmission modes (ULP/LP/HS).

The MIPI D-PHY supports the following features:

- Mixed-signal D-PHY mixed-signal hard-macro- LS Transmitter and LS/HS Receiver solution
- Designed to MIPI® v1.0 Specifications
- Integrated PHY Protocol Interface (PPI) supports interface to CSI, DSI and UniPro™ MIPI® protocols
- 1.0GHz maximum data transfer rate per lane
- Expandable to support 4 data lanes, providing up to 4Gbps transfer rate
- HS, LP and ULPS modes supported
- 10Mbps per lane in low-power mode
- Unidirectional and bi-directional modes supported
- Automatic termination control for HS and LP modes
- Low-Power dissipation: HS less than 3mA/Lane
- Tx/Rx Buffers with tunable On-Die-Termination and advanced equalization.
- Embedded ESD, boundary scan support logic.

46.2 Block Diagram

The MIPI D-PHY configuration contains one Clock Lane Module and four Data Lane Modules. Each of these PHY Lane Modules communicates via two Lines to a complementary part at the other side of the Lane Interconnect. The following diagram shows the D-PHY architecture.

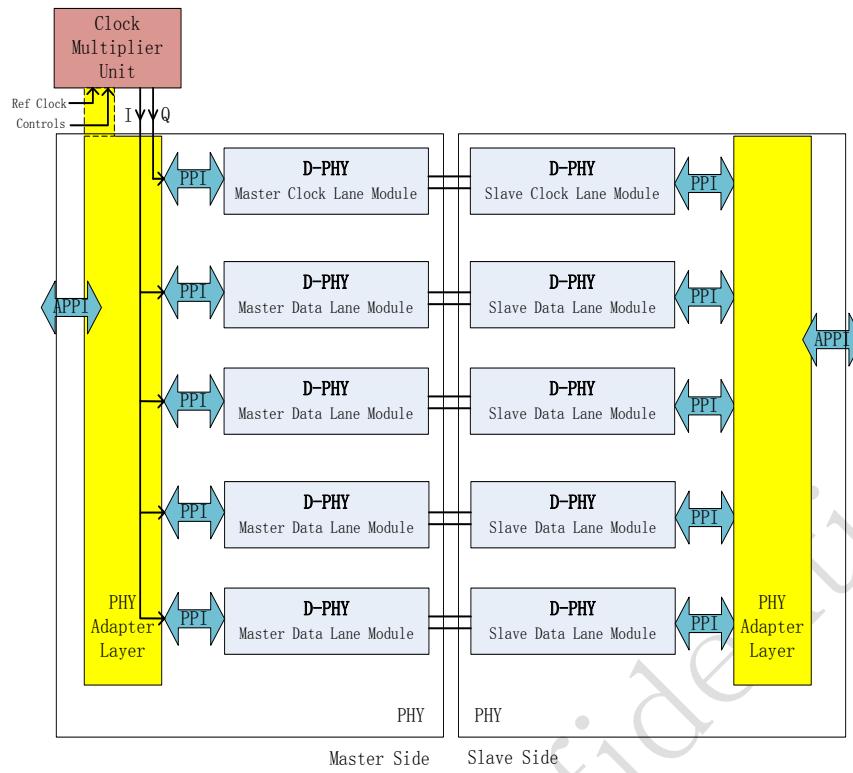


Fig.46-1 MIPI D-PHY simplified Block diagram with master to slave

The following diagram shows a Universal Lane Module Diagram with a global overview of internal functionality of the CIL function. This Universal Module can be used for all Lane types. The requirements for the 'Control and Interface Logic' (CIL) function depend on the Lane type and Lane side.

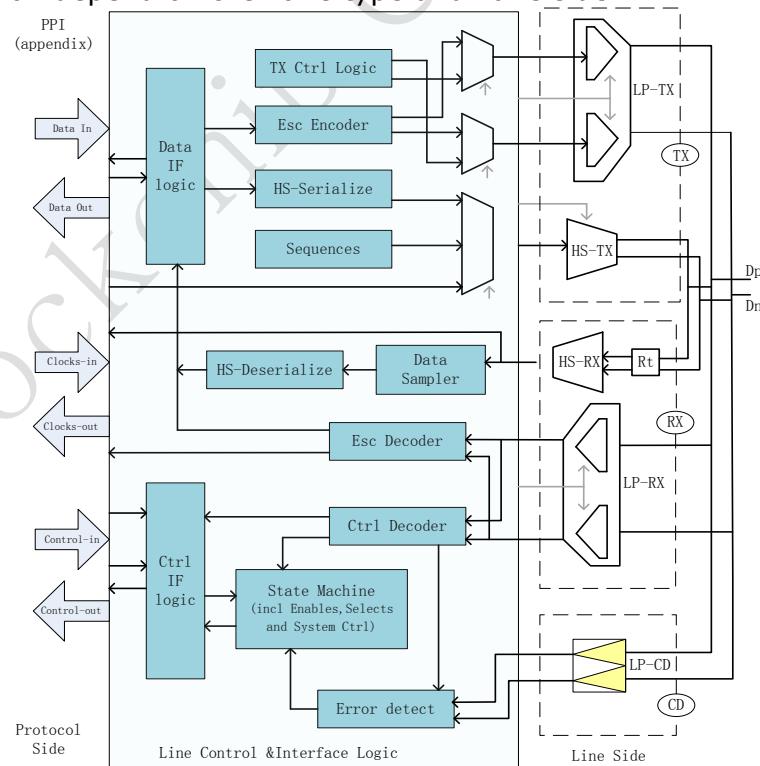


Fig.46-2 MIPI D-PHY V1.0 detailed block diagram

46.3 Function Description

The MIPI D-PHY transceiver is designed to reliably transmit HS and LP/ULP data/clock over the channel and recover the MIPI LP data stream from any MIPI input signal. It consists of 4 data transceiver paths and 1 clock transmitting path. For each data lane a HS transmitter and a LP transceiver is necessary to transmit/recover the data streams, for the clock lane, a HS/LP transmitter is designed to output the high speed clock signal over the channel.

A HS differential signal driven on the D_p and D_n pins is generated by a differential output driver. For reference, D_p is considered as the positive side and D_n as the negative side. High speed current switches are designed to output data streams over the channels.

The Low-Power receiver is an un-terminated, single-ended receiver circuit. LP receiver is used to detect the Low-Power state on each pin. For high robustness, the LP receiver can filter out noise pulses and RF interference. Furthermore, any spikes with a pulse width smaller than 20ns will be rejected.

Contention Detector (LP-CD) is designed in Data Lane to monitor the line voltage on each Low-Power signal. The LP-CD is used to detect an LP low fault when the LP transmitter is driving low and the pin voltage is greater than 450mV.

The Low-Power transmitter is a slew-rate controlled push-pull driver. The minimum pull-down and pull-up impedance of LP driver is 110 ohm. At the same time tunable slew rate control logic is available for eye pattern requirement.

46.4 Register Description

This section describes the control/status registers of the design. While you are reading this chapter please note that the offset address[7:0] is distributed two parts, one from the bit7 to bit5 is the first address, the other from the bit4 to bit0 is the second address. When you configure the registers, you must set both of them. The Clock Lane and Data Lane use the same registers with the same second address, but the first address is different. Itsapb base address is 0xc00.

46.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
MIPIPHY_REG0	0x0000	W	0x00000001	mipiphy register 0
MIPIPHY_REG1	0x0004	W	0x00000003	mipiphy register 1
MIPIPHY_REG3	0x000c	W	0x00000003	mipiphy register 3
MIPIPHY_REG4	0x0010	W	0x0000007d	mipiphy register 4
MIPIPHY_REG20	0x0080	W	0x00000001	mipiphy register 20
MIPIPHY_REG40	0x0100	W	0x0000000b	mipiphy register 40
MIPIPHY_REG45	0x0114	W	0x00000005	mipiphy register 45
MIPIPHY_REG46	0x0118	W	0x00000000	mipiphy register 46
MIPIPHY_REG47	0x011c	W	0x00000000	mipiphy register 47
MIPIPHY_REG48	0x0120	W	0x00000000	mipiphy register 48
MIPIPHY_REG49	0x0124	W	0x00000000	mipiphy register 49
MIPIPHY_REG4A	0x0128	W	0x00000000	mipiphy register 4a
MIPIPHY_REG4B	0x012c	W	0x00000000	mipiphy register 4b
MIPIPHY_REG4C	0x0130	W	0x00000000	mipiphy register 4c
MIPIPHY_REG4D	0x0134	W	0x00000000	mipiphy register 4d
MIPIPHY_REG4E	0x0138	W	0x00000000	mipiphy register 4e
MIPIPHY_REG50	0x0140	W	0x00000000	mipiphy register 50
MIPIPHY_REG51	0x0144	W	0x00000000	mipiphy register 51
MIPIPHY_REG52	0x0148	W	0x00000000	mipiphy register 52
MIPIPHY_REG60	0x0180	W	0x0000000b	mipiphy register 60
MIPIPHY_REG65	0x0194	W	0x00000005	mipiphy register 65
MIPIPHY_REG66	0x0198	W	0x00000000	mipiphy register 66
MIPIPHY_REG67	0x019c	W	0x00000000	mipiphy register 67
MIPIPHY_REG68	0x01a0	W	0x00000000	mipiphy register 68
MIPIPHY_REG69	0x01a4	W	0x00000000	mipiphy register 69
MIPIPHY_REG6A	0x01a8	W	0x00000000	mipiphy register 6a
MIPIPHY_REG6B	0x01ac	W	0x00000000	mipiphy register 6b
MIPIPHY_REG6C	0x01b0	W	0x00000000	mipiphy register 6c
MIPIPHY_REG6D	0x01b4	W	0x00000000	mipiphy register 6d
MIPIPHY_REG6E	0x01b8	W	0x00000000	mipiphy register 6e
MIPIPHY_REG70	0x01c0	W	0x00000000	mipiphy register 70
MIPIPHY_REG71	0x01c4	W	0x00000000	mipiphy register 71
MIPIPHY_REG72	0x01c8	W	0x00000000	mipiphy register 72
MIPIPHY_REG80	0x0200	W	0x0000000b	mipiphy register 80
MIPIPHY_REG85	0x0214	W	0x00000005	mipiphy register 85
MIPIPHY_REG86	0x0218	W	0x00000000	mipiphy register 86
MIPIPHY_REG87	0x021c	W	0x00000000	mipiphy register 87
MIPIPHY_REG88	0x0220	W	0x00000000	mipiphy register 88
MIPIPHY_REG89	0x0224	W	0x00000000	mipiphy register 89
MIPIPHY_REG8A	0x0228	W	0x00000000	mipiphy register 8a
MIPIPHY_REG8B	0x022c	W	0x00000000	mipiphy register 8b
MIPIPHY_REG8C	0x0230	W	0x00000000	mipiphy register 8c
MIPIPHY_REG8D	0x0234	W	0x00000000	mipiphy register 8d
MIPIPHY_REG8E	0x0238	W	0x00000000	mipiphy register 8e
MIPIPHY_REG90	0x0240	W	0x00000000	mipiphy register 90
MIPIPHY_REG91	0x0244	W	0x00000000	mipiphy register 91
MIPIPHY_REG92	0x0248	W	0x00000000	mipiphy register 92

Name	Offset	Size	Reset Value	Description
MIPIPHY_REGA0	0x0280	W	0x0000000b	mipiphy register a0
MIPIPHY_REGA5	0x0294	W	0x00000005	mipiphy register a5
MIPIPHY_REGA6	0x0298	W	0x00000000	mipiphy register a6
MIPIPHY_REGA7	0x029c	W	0x00000000	mipiphy register a7
MIPIPHY_REGA8	0x02a0	W	0x00000000	mipiphy register a8
MIPIPHY_REGA9	0x02a4	W	0x00000000	mipiphy register a9
MIPIPHY_REGAA	0x02a8	W	0x00000000	mipiphy register aa
MIPIPHY_REGAB	0x02ac	W	0x00000000	mipiphy register ab
MIPIPHY_REGAC	0x02b0	W	0x00000000	mipiphy register ac
MIPIPHY_REGAD	0x02b4	W	0x00000000	mipiphy register ad
MIPIPHY_REGAE	0x02b8	W	0x00000000	mipiphy register ae
MIPIPHY_REGB0	0x02c0	W	0x00000000	mipiphy register b0
MIPIPHY_REGB1	0x02c4	W	0x00000000	mipiphy register b1
MIPIPHY_REGB2	0x02c8	W	0x00000000	mipiphy register b2
MIPIPHY_REGC0	0x0300	W	0x0000000b	mipiphy register c0
MIPIPHY_REGC5	0x0314	W	0x00000005	mipiphy register c5
MIPIPHY_REGC6	0x0318	W	0x00000000	mipiphy register c6
MIPIPHY_REGC7	0x031c	W	0x00000000	mipiphy register c7
MIPIPHY_REGC8	0x0320	W	0x00000000	mipiphy register c8
MIPIPHY_REGC9	0x0324	W	0x00000000	mipiphy register c9
MIPIPHY_REGCA	0x0328	W	0x00000000	mipiphy register ca
MIPIPHY_REGCB	0x032c	W	0x00000000	mipiphy register cb
MIPIPHY_REGCC	0x0330	W	0x00000000	mipiphy register cc
MIPIPHY_REGCD	0x0334	W	0x00000000	mipiphy register cd
MIPIPHY_REGCE	0x0338	W	0x00000000	mipiphy register ce
MIPIPHY_REGD0	0x0340	W	0x00000000	mipiphy register d0
MIPIPHY_REGD1	0x0344	W	0x00000000	mipiphy register d1
MIPIPHY_REGD2	0x0348	W	0x00000000	mipiphy register d2
MIPIPHY_REGE0	0x0380	W	0x00000000	mipiphy register e0
MIPIPHY_REGEA	0x03a8	W	0x00000000	mipiphy register ea

Notes: **Size:** **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

46.4.2 Detail Register Description

MIPIPHY_REG0

Address: Operational Base + offset (0x0000)
mipiphy register 0

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	lane_en_ck 1: enable 0: disable
5	RW	0x0	lane_en_3 1: enable 0: disable
4	RW	0x0	lane_en_2 1: enable 0: disable

Bit	Attr	Reset Value	Description
3	RW	0x0	lane_en_1 1: enable 0: disable
2	RW	0x0	lane_en_0 1: enable 0: disable
1	RW	0x0	reserved1
0	RO	0x1	reserved

MIPIPHY_REG1

Address: Operational Base + offset (0x0004)

mipiphy register 1

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	reg_da_syncrst 1: reset 0: normal
1	RW	0x1	reg_da_ldopd 1: power down 0: power on
0	RW	0x1	reg_da_pllpd 1: power down 0: power on

MIPIPHY_REG3

Address: Operational Base + offset (0x000c)

mipiphy register 3

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	reg_fbdv reg_fbdv[8] PLL input reference clock divider
4:0	RW	0x03	reg_pdiv reg_pdiv[4:0] Integer value programmed into feedback divider

MIPIPHY_REG4

Address: Operational Base + offset (0x0010)

mipiphy register 4

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x7d	reg_fbdv reg_fbdv[7:0] PLL input reference clock divider

MIPIPHY_REG20

Address: Operational Base + offset (0x0080)

mipiphy register 20

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	reg_dig_rstn 1: normal 0: reset

MIPIPHY_REG40

Address: Operational Base + offset (0x0100)
 mipiphy register 40

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0xb	reg_ths_settle Clock Lane Configure the count time of the THS-SETTLE by protocol. After the count done, D-PHY will begin to receive the high speed data. (Can be configured from 4'h0 to 4'hd) 4'b0000 80-110 MHz 4'b0001 110-150 MHz 4'b0010 150-200 MHz 4'b0011 200-250 MHz 4'b0100 250-300 MHz 4'b0101 300-400 MHz 4'b0110 400-500 MHz 4'b0111 500-600 MHz 4'b1000 600-700 MHz 4'b1001 700-800 MHz 4'b1010 800-1000 MHz 4'b1011 additional adjust 4'b1100 additional adjust 4'b1101 additional adjust 4'b1110 additional adjust

MIPIPHY_REG45

Address: Operational Base + offset (0x0114)
 mipiphy register 45

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x05	reg_hs_tlpk Clock Lane The value of counter for HS Tlpk Time (\geq Tlpk) $= \text{Tpin_txbyteclkhs} * \text{value}$

MIPIPHY_REG46

Address: Operational Base + offset (0x0118)
 mipiphy register 46

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	reg_hs_ths_prepare Clock Lane The value of counter for HS Ths-prepare For clock lane, Ths-prepare(38ns~95ns) For data lane, Ths-prepare(40ns+4UI~85ns+6UI) =Ttxddrclkhs*value

MIPIPHY_REG47

Address: Operational Base + offset (0x011c)
mipiphy register 47

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_the_zero Clock Lane The value of counter for HS Ths-zero For clock lane, Ths-prepare+Ths-zero (>=300ns) For data lane, Ths-prepare+Ths-zero (>= 145 ns + 10*UI) = Tpin_txbyteclkhs*value For clock lane, S_HSTXTHSZERO[5:0] = 6'b100000 For data lane, S_HSTXTHSZERO[5:0] = 6'b001001 Frequency(1/UI) Value(Decimal) 80 -110 MHz 3 110-150 MHz 4 150-200 MHz 4 200-250 MHz 5 250-300 MHz 6 300-400 MHz 7 400-500 MHz 8 500-600 MHz 10 600-700 MHz 11 700-800 MHz 12 800-1000 MHz 15

MIPIPHY_REG48

Address: Operational Base + offset (0x0120)
mipiphy register 48

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description																								
6:0	RW	0x00	<p>reg_hs_ths_trail Clock Lane The value of counter for HS Ths-trail For clock lane, Ths-trail (\geq60ns) For data lane, Ths-trail (\geqmax(8UI, 60ns+4UI)) = Tad_txddrclkhs_i* value For clock lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111 For data lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111</p> <table> <tr> <td>Frequency(1/UI)</td> <td>Value(Decimal)</td> </tr> <tr> <td>80 -110 MHz</td> <td>12</td> </tr> <tr> <td>110-150 MHz</td> <td>13</td> </tr> <tr> <td>150-200 MHz</td> <td>17</td> </tr> <tr> <td>200-250 MHz</td> <td>20</td> </tr> <tr> <td>250-300 MHz</td> <td>24</td> </tr> <tr> <td>300-400 MHz</td> <td>29</td> </tr> <tr> <td>400-500 MHz</td> <td>35</td> </tr> <tr> <td>500-600 MHz</td> <td>41</td> </tr> <tr> <td>600-700 MHz</td> <td>49</td> </tr> <tr> <td>700-800 MHz</td> <td>52</td> </tr> <tr> <td>800-1000 MHz</td> <td>64</td> </tr> </table>	Frequency(1/UI)	Value(Decimal)	80 -110 MHz	12	110-150 MHz	13	150-200 MHz	17	200-250 MHz	20	250-300 MHz	24	300-400 MHz	29	400-500 MHz	35	500-600 MHz	41	600-700 MHz	49	700-800 MHz	52	800-1000 MHz	64
Frequency(1/UI)	Value(Decimal)																										
80 -110 MHz	12																										
110-150 MHz	13																										
150-200 MHz	17																										
200-250 MHz	20																										
250-300 MHz	24																										
300-400 MHz	29																										
400-500 MHz	35																										
500-600 MHz	41																										
600-700 MHz	49																										
700-800 MHz	52																										
800-1000 MHz	64																										

MIPPHY_REG49

Address: Operational Base + offset (0x0124)
mipiphy register 49

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>reg_hs_ths_exit Clock Lane The value of counter for HS Ths-exit Ths-exit = Tpin_txbyteclkhs*value</p>

MIPPHY_REG4A

Address: Operational Base + offset (0x0128)
mipiphy register 4a

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>reg_hs_tclk_post Clock Lane The value of counter for HS Tclk-post Tclk-post = Tpin_txbyteclkhs*value</p>

MIPPHY_REG4B

Address: Operational Base + offset (0x012c)

mipi phy register 4b

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reserved

MIPIPHY_REG4C

Address: Operational Base + offset (0x0130)

mipi phy register 4c

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	reg_hs_twakup Clock Lane The value[9:8] of counter for HS Twakup also see REG4D

MIPIPHY_REG4D

Address: Operational Base + offset (0x0134)

mipi phy register 4d

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reg_hs_twakup Clock Lane The value[7:0] of counter for HS Twakup Twakup for ulpm, Twakup = Tpin_sys_clk*value[9:0]

MIPIPHY_REG4E

Address: Operational Base + offset (0x0138)

mipi phy register 4e

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	reg_hs_tclk_pre Clock Lane The value of counter for HS Tclk-pre Tclk-pre for clock lane Tclk-pre = Tpin_txbyteclkhs*value

MIPIPHY_REG50

Address: Operational Base + offset (0x0140)

mipi phy register 50

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	reg_hs_tta_go Clock Lane The value of counter for HS Tta-go Tta-go for turnaround Tta-go $= \text{Txclkesc} * \text{value}$

MIPIPHY_REG51

Address: Operational Base + offset (0x0144)

mipiphy register 51

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_tta_sure Clock Lane The value of counter for HS Tta-sure Tta-sure for turnaround Tta-sure $= \text{Txclkesc} * \text{value}$

MIPIPHY_REG52

Address: Operational Base + offset (0x0148)

mipiphy register 52

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_tta_wait Clock Lane The value of counter for HS Tta-wait Tta-wait for turnaround Interval from receiving ppi turnaround request to sending esc request. Tta-wait $= \text{Txclkesc} * \text{value}$

MIPIPHY_REG60

Address: Operational Base + offset (0x0180)

mipiphy register 60

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description																														
3:0	RW	0xb	<p>reg_ths_settle Data0 Lane Configure the count time of the THS-SETTLE by protocol. After the count done, D-PHY will begin to receive the high speed data. (Can be configured from 4'h0 to 4'hd)</p> <table> <tr><td>4'b0000</td><td>80-110 MHz</td></tr> <tr><td>4'b0001</td><td>110-150 MHz</td></tr> <tr><td>4'b0010</td><td>150-200 MHz</td></tr> <tr><td>4'b0011</td><td>200-250 MHz</td></tr> <tr><td>4'b0100</td><td>250-300 MHz</td></tr> <tr><td>4'b0101</td><td>300-400 MHz</td></tr> <tr><td>4'b0110</td><td>400-500 MHz</td></tr> <tr><td>4'b0111</td><td>500-600 MHz</td></tr> <tr><td>4'b1000</td><td>600-700 MHz</td></tr> <tr><td>4'b1001</td><td>700-800 MHz</td></tr> <tr><td>4'b1010</td><td>800-1000 MHz</td></tr> <tr><td>4'b1011</td><td>additional adjust</td></tr> <tr><td>4'b1100</td><td>additional adjust</td></tr> <tr><td>4'b1101</td><td>additional adjust</td></tr> <tr><td>4'b1110</td><td>additional adjust</td></tr> </table>	4'b0000	80-110 MHz	4'b0001	110-150 MHz	4'b0010	150-200 MHz	4'b0011	200-250 MHz	4'b0100	250-300 MHz	4'b0101	300-400 MHz	4'b0110	400-500 MHz	4'b0111	500-600 MHz	4'b1000	600-700 MHz	4'b1001	700-800 MHz	4'b1010	800-1000 MHz	4'b1011	additional adjust	4'b1100	additional adjust	4'b1101	additional adjust	4'b1110	additional adjust
4'b0000	80-110 MHz																																
4'b0001	110-150 MHz																																
4'b0010	150-200 MHz																																
4'b0011	200-250 MHz																																
4'b0100	250-300 MHz																																
4'b0101	300-400 MHz																																
4'b0110	400-500 MHz																																
4'b0111	500-600 MHz																																
4'b1000	600-700 MHz																																
4'b1001	700-800 MHz																																
4'b1010	800-1000 MHz																																
4'b1011	additional adjust																																
4'b1100	additional adjust																																
4'b1101	additional adjust																																
4'b1110	additional adjust																																

MIPIPHY_REG65

Address: Operational Base + offset (0x0194)
mipiphy register 65

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x05	<p>reg_hs_tlpix Data0 Lane The value of counter for HS Tlpix Time (\geqTlpix) $= Tpin_txbyteclkhs * value$</p>

MIPIPHY_REG66

Address: Operational Base + offset (0x0198)
mipiphy register 66

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	<p>reg_hs_ths_prepare Data0 Lane The value of counter for HS Ths-prepare For clock lane, Ths-prepare(38ns~95ns) For data lane, Ths-prepare(40ns+4UI~85ns+6UI) $= Ttxddrclkhs * value$</p>

MIPIPHY_REG67

Address: Operational Base + offset (0x019c)
 mipiphy register 67

Bit	Attr	Reset Value	Description																								
31:6	RO	0x0	reserved																								
5:0	RW	0x00	<p>reg_hs_the_zero Data0 Lane The value of counter for HS Ths-zero For clock lane, Ths-prepare+Ths-zero (>=300ns) For data lane, Ths-prepare+Ths-zero (>= 145 ns + 10*UI) = Tpin_txbyteclkhs*value For clock lane, S_HSTXTHSZERO[5:0] = 6'b100000 For data lane, S_HSTXTHSZERO[5:0] = 6'b001001</p> <table> <thead> <tr> <th>Frequency(1/UI)</th> <th>Value(Decimal)</th> </tr> </thead> <tbody> <tr> <td>80 -110 MHz</td> <td>3</td> </tr> <tr> <td>110-150 MHz</td> <td>4</td> </tr> <tr> <td>150-200 MHz</td> <td>4</td> </tr> <tr> <td>200-250 MHz</td> <td>5</td> </tr> <tr> <td>250-300 MHz</td> <td>6</td> </tr> <tr> <td>300-400 MHz</td> <td>7</td> </tr> <tr> <td>400-500 MHz</td> <td>8</td> </tr> <tr> <td>500-600 MHz</td> <td>10</td> </tr> <tr> <td>600-700 MHz</td> <td>11</td> </tr> <tr> <td>700-800 MHz</td> <td>12</td> </tr> <tr> <td>800-1000 MHz</td> <td>15</td> </tr> </tbody> </table>	Frequency(1/UI)	Value(Decimal)	80 -110 MHz	3	110-150 MHz	4	150-200 MHz	4	200-250 MHz	5	250-300 MHz	6	300-400 MHz	7	400-500 MHz	8	500-600 MHz	10	600-700 MHz	11	700-800 MHz	12	800-1000 MHz	15
Frequency(1/UI)	Value(Decimal)																										
80 -110 MHz	3																										
110-150 MHz	4																										
150-200 MHz	4																										
200-250 MHz	5																										
250-300 MHz	6																										
300-400 MHz	7																										
400-500 MHz	8																										
500-600 MHz	10																										
600-700 MHz	11																										
700-800 MHz	12																										
800-1000 MHz	15																										

MIPPHY_REG68

Address: Operational Base + offset (0x01a0)
 mipiphy register 68

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description																								
6:0	RW	0x00	<p>reg_hs_ths_trail Data0 Lane The value of counter for HS Ths-trail For clock lane, Ths-trail (\geq60ns) For data lane, Ths-trail (\geqmax(8UI, 60ns+4UI)) = Tad_txddrclkhs_i* value For clock lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111 For data lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111</p> <table> <tr> <td>Frequency(1/UI)</td> <td>Value(Decimal)</td> </tr> <tr> <td>80 -110 MHz</td> <td>12</td> </tr> <tr> <td>110-150 MHz</td> <td>13</td> </tr> <tr> <td>150-200 MHz</td> <td>17</td> </tr> <tr> <td>200-250 MHz</td> <td>20</td> </tr> <tr> <td>250-300 MHz</td> <td>24</td> </tr> <tr> <td>300-400 MHz</td> <td>29</td> </tr> <tr> <td>400-500 MHz</td> <td>35</td> </tr> <tr> <td>500-600 MHz</td> <td>41</td> </tr> <tr> <td>600-700 MHz</td> <td>49</td> </tr> <tr> <td>700-800 MHz</td> <td>52</td> </tr> <tr> <td>800-1000 MHz</td> <td>64</td> </tr> </table>	Frequency(1/UI)	Value(Decimal)	80 -110 MHz	12	110-150 MHz	13	150-200 MHz	17	200-250 MHz	20	250-300 MHz	24	300-400 MHz	29	400-500 MHz	35	500-600 MHz	41	600-700 MHz	49	700-800 MHz	52	800-1000 MHz	64
Frequency(1/UI)	Value(Decimal)																										
80 -110 MHz	12																										
110-150 MHz	13																										
150-200 MHz	17																										
200-250 MHz	20																										
250-300 MHz	24																										
300-400 MHz	29																										
400-500 MHz	35																										
500-600 MHz	41																										
600-700 MHz	49																										
700-800 MHz	52																										
800-1000 MHz	64																										

MIPPHY_REG69

Address: Operational Base + offset (0x01a4)
mipiphy register 69

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>reg_hs_ths_exit Data0 Lane The value of counter for HS Ths-exit Ths-exit = Tpin_txbyteclkhs*value</p>

MIPPHY_REG6A

Address: Operational Base + offset (0x01a8)
mipiphy register 6a

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>reg_hs_tclk_post Data0 Lane The value of counter for HS Tclk-post Tclk-post = Tpin_txbyteclkhs*value</p>

MIPPHY_REG6B

Address: Operational Base + offset (0x01ac)

mipiphy register 6b

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reserved Data0 Lane reserved

MIPIPHY_REG6C

Address: Operational Base + offset (0x01b0)

mipiphy register 6c

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	reg_hs_twakup Data0 Lane The value[9:8] of counter for HS Twakup also see REG6D

MIPIPHY_REG6D

Address: Operational Base + offset (0x01b4)

mipiphy register 6d

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reg_hs_twakup Data0 Lane The value[7:0] of counter for HS Twakup Twakup for ulpm, Twakup = Tpin_sys_clk*value[9:0]

MIPIPHY_REG6E

Address: Operational Base + offset (0x01b8)

mipiphy register 6e

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	reg_hs_tclk_pre Data0 Lane The value of counter for HS Tclk-pre Tclk-pre for clock lane Tclk-pre = Tpin_txbyteclkhs*value

MIPIPHY_REG70

Address: Operational Base + offset (0x01c0)

mipiphy register 70

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	<p>reg_hs_tta_go Data0 Lane The value of counter for HS Tta-go Tta-go for turnaround Tta-go = Ttxclkesc*value</p>

MIPIPHY_REG71

Address: Operational Base + offset (0x01c4)

mipiphy register 71

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	<p>reg_hs_tta_sure Data0 Lane The value of counter for HS Tta-sure Tta-sure for turnaround Tta-sure = Ttxclkesc*value</p>

MIPIPHY_REG72

Address: Operational Base + offset (0x01c8)

mipiphy register 72

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	<p>reg_hs_tta_wait Data0 Lane The value of counter for HS Tta-wait Tta-wait for turnaround Interval from receiving ppi turnaround request to sending esc request. Tta-wait = Ttxclkesc*value</p>

MIPIPHY_REG80

Address: Operational Base + offset (0x0200)

mipiphy register 80

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description																														
3:0	RW	0xb	<p>reg_ths_settle Data1 Lane Configure the count time of the THS-SETTLE by protocol. After the count done, D-PHY will begin to receive the high speed data. (Can be configured from 4'h0 to 4'hd)</p> <table> <tr><td>4'b0000</td><td>80-110 MHz</td></tr> <tr><td>4'b0001</td><td>110-150 MHz</td></tr> <tr><td>4'b0010</td><td>150-200 MHz</td></tr> <tr><td>4'b0011</td><td>200-250 MHz</td></tr> <tr><td>4'b0100</td><td>250-300 MHz</td></tr> <tr><td>4'b0101</td><td>300-400 MHz</td></tr> <tr><td>4'b0110</td><td>400-500 MHz</td></tr> <tr><td>4'b0111</td><td>500-600 MHz</td></tr> <tr><td>4'b1000</td><td>600-700 MHz</td></tr> <tr><td>4'b1001</td><td>700-800 MHz</td></tr> <tr><td>4'b1010</td><td>800-1000 MHz</td></tr> <tr><td>4'b1011</td><td>additional adjust</td></tr> <tr><td>4'b1100</td><td>additional adjust</td></tr> <tr><td>4'b1101</td><td>additional adjust</td></tr> <tr><td>4'b1110</td><td>additional adjust</td></tr> </table>	4'b0000	80-110 MHz	4'b0001	110-150 MHz	4'b0010	150-200 MHz	4'b0011	200-250 MHz	4'b0100	250-300 MHz	4'b0101	300-400 MHz	4'b0110	400-500 MHz	4'b0111	500-600 MHz	4'b1000	600-700 MHz	4'b1001	700-800 MHz	4'b1010	800-1000 MHz	4'b1011	additional adjust	4'b1100	additional adjust	4'b1101	additional adjust	4'b1110	additional adjust
4'b0000	80-110 MHz																																
4'b0001	110-150 MHz																																
4'b0010	150-200 MHz																																
4'b0011	200-250 MHz																																
4'b0100	250-300 MHz																																
4'b0101	300-400 MHz																																
4'b0110	400-500 MHz																																
4'b0111	500-600 MHz																																
4'b1000	600-700 MHz																																
4'b1001	700-800 MHz																																
4'b1010	800-1000 MHz																																
4'b1011	additional adjust																																
4'b1100	additional adjust																																
4'b1101	additional adjust																																
4'b1110	additional adjust																																

MIPIPHY_REG85

Address: Operational Base + offset (0x0214)
mipiphy register 85

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x05	<p>reg_hs_tlpix Data1 Lane The value of counter for HS Tlpix Time (\geqTlpix) = Tpin_txbyteclkhs * value</p>

MIPIPHY_REG86

Address: Operational Base + offset (0x0218)
mipiphy register 86

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	<p>reg_hs_ths_prepare Data1 Lane The value of counter for HS Ths-prepare For clock lane, Ths-prepare(38ns~95ns) For data lane, Ths-prepare(40ns+4UI~85ns+6UI) = Ttxddrclkhs*value</p>

MIPIPHY_REG87

Address: Operational Base + offset (0x021c)
 mipiphy register 87

Bit	Attr	Reset Value	Description																						
31:6	RO	0x0	reserved																						
5:0	RW	0x00	<p>reg_hs_the_zero Data1 Lane The value of counter for HS Ths-zero For clock lane, Ths-prepare+Ths-zero ($\geq 300\text{ns}$) For data lane, Ths-prepare+Ths-zero ($\geq 145\text{ ns} + 10*\text{UI}$) $= \text{Tpin_txbyteclkhs} * \text{value}$ For clock lane, S_HSTXTHSZERO[5:0] = 6'b100000 For data lane, S_HSTXTHSZERO[5:0] = 6'b001001 Frequency(1/UI) Value(Decimal)</p> <table> <tbody> <tr><td>80 -110 MHz</td><td>3</td></tr> <tr><td>110-150 MHz</td><td>4</td></tr> <tr><td>150-200 MHz</td><td>4</td></tr> <tr><td>200-250 MHz</td><td>5</td></tr> <tr><td>250-300 MHz</td><td>6</td></tr> <tr><td>300-400 MHz</td><td>7</td></tr> <tr><td>400-500 MHz</td><td>8</td></tr> <tr><td>500-600 MHz</td><td>10</td></tr> <tr><td>600-700 MHz</td><td>11</td></tr> <tr><td>700-800 MHz</td><td>12</td></tr> <tr><td>800-1000 MHz</td><td>15</td></tr> </tbody> </table>	80 -110 MHz	3	110-150 MHz	4	150-200 MHz	4	200-250 MHz	5	250-300 MHz	6	300-400 MHz	7	400-500 MHz	8	500-600 MHz	10	600-700 MHz	11	700-800 MHz	12	800-1000 MHz	15
80 -110 MHz	3																								
110-150 MHz	4																								
150-200 MHz	4																								
200-250 MHz	5																								
250-300 MHz	6																								
300-400 MHz	7																								
400-500 MHz	8																								
500-600 MHz	10																								
600-700 MHz	11																								
700-800 MHz	12																								
800-1000 MHz	15																								

MIPPHY_REG88

Address: Operational Base + offset (0x0220)
 mipiphy register 88

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description																								
6:0	RW	0x00	<p>reg_hs_ths_trail Data1 Lane The value of counter for HS Ths-trail For clock lane, Ths-trail (\geq60ns) For data lane, Ths-trail (\geqmax(8UI, 60ns+4UI)) = Tad_txddrclkhs_i* value For clock lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111 For data lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111</p> <table> <tr> <td>Frequency(1/UI)</td> <td>Value(Decimal)</td> </tr> <tr> <td>80 -110 MHz</td> <td>12</td> </tr> <tr> <td>110-150 MHz</td> <td>13</td> </tr> <tr> <td>150-200 MHz</td> <td>17</td> </tr> <tr> <td>200-250 MHz</td> <td>20</td> </tr> <tr> <td>250-300 MHz</td> <td>24</td> </tr> <tr> <td>300-400 MHz</td> <td>29</td> </tr> <tr> <td>400-500 MHz</td> <td>35</td> </tr> <tr> <td>500-600 MHz</td> <td>41</td> </tr> <tr> <td>600-700 MHz</td> <td>49</td> </tr> <tr> <td>700-800 MHz</td> <td>52</td> </tr> <tr> <td>800-1000 MHz</td> <td>64</td> </tr> </table>	Frequency(1/UI)	Value(Decimal)	80 -110 MHz	12	110-150 MHz	13	150-200 MHz	17	200-250 MHz	20	250-300 MHz	24	300-400 MHz	29	400-500 MHz	35	500-600 MHz	41	600-700 MHz	49	700-800 MHz	52	800-1000 MHz	64
Frequency(1/UI)	Value(Decimal)																										
80 -110 MHz	12																										
110-150 MHz	13																										
150-200 MHz	17																										
200-250 MHz	20																										
250-300 MHz	24																										
300-400 MHz	29																										
400-500 MHz	35																										
500-600 MHz	41																										
600-700 MHz	49																										
700-800 MHz	52																										
800-1000 MHz	64																										

MIPIPHY_REG89

Address: Operational Base + offset (0x0224)
mipiphy register 89

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>reg_hs_ths_exit Data1 Lane The value of counter for HS Ths-exit Ths-exit = Tpin_txbyteclkhs*value</p>

MIPIPHY_REG8A

Address: Operational Base + offset (0x0228)
mipiphy register 8a

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>reg_hs_tclk_post Data1 Lane The value of counter for HS Tclk-post Tclk-post = Tpin_txbyteclkhs*value</p>

MIPIPHY_REG8B

Address: Operational Base + offset (0x022c)

mipi phy register 8b

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reserved

MIPIPHY_REG8C

Address: Operational Base + offset (0x0230)

mipi phy register 8c

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	reg_hs_twakup Data1 Lane The value[9:8] of counter for HS Twakup also see REG8D

MIPIPHY_REG8D

Address: Operational Base + offset (0x0234)

mipi phy register 8d

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reg_hs_twakup Data1 Lane The value[7:0] of counter for HS Twakup Twakup for ulpm, Twakup = Tpin_sys_clk*value[9:0]

MIPIPHY_REG8E

Address: Operational Base + offset (0x0238)

mipi phy register 8e

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	reg_hs_tclk_pre Data1 Lane The value of counter for HS Tclk-pre Tclk-pre for clock lane Tclk-pre = Tpin_txbyteclkhs*value

MIPIPHY_REG90

Address: Operational Base + offset (0x0240)

mipi phy register 90

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	reg_hs_tta_go Data1 Lane The value of counter for HS Tta-go Tta-go for turnaround Tta-go $= \text{Txclkesc} * \text{value}$

MIPIPHY_REG91

Address: Operational Base + offset (0x0244)

mipiphy register 91

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_tta_sure Data1 Lane The value of counter for HS Tta-sure Tta-sure for turnaround Tta-sure $= \text{Txclkesc} * \text{value}$

MIPIPHY_REG92

Address: Operational Base + offset (0x0248)

mipiphy register 92

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_tta_wait Data1 Lane The value of counter for HS Tta-wait Tta-wait for turnaround Interval from receiving ppi turnaround request to sending esc request. Tta-wait $= \text{Txclkesc} * \text{value}$

MIPIPHY_REGA0

Address: Operational Base + offset (0x0280)

mipiphy register a0

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description																														
3:0	RW	0xb	<p>reg_ths_settle Data2 Lane Configure the count time of the THS-SETTLE by protocol. After the count done, D-PHY will begin to receive the high speed data. (Can be configured from 4'h0 to 4'hd)</p> <table> <tr><td>4'b0000</td><td>80-110 MHz</td></tr> <tr><td>4'b0001</td><td>110-150 MHz</td></tr> <tr><td>4'b0010</td><td>150-200 MHz</td></tr> <tr><td>4'b0011</td><td>200-250 MHz</td></tr> <tr><td>4'b0100</td><td>250-300 MHz</td></tr> <tr><td>4'b0101</td><td>300-400 MHz</td></tr> <tr><td>4'b0110</td><td>400-500 MHz</td></tr> <tr><td>4'b0111</td><td>500-600 MHz</td></tr> <tr><td>4'b1000</td><td>600-700 MHz</td></tr> <tr><td>4'b1001</td><td>700-800 MHz</td></tr> <tr><td>4'b1010</td><td>800-1000 MHz</td></tr> <tr><td>4'b1011</td><td>additional adjust</td></tr> <tr><td>4'b1100</td><td>additional adjust</td></tr> <tr><td>4'b1101</td><td>additional adjust</td></tr> <tr><td>4'b1110</td><td>additional adjust</td></tr> </table>	4'b0000	80-110 MHz	4'b0001	110-150 MHz	4'b0010	150-200 MHz	4'b0011	200-250 MHz	4'b0100	250-300 MHz	4'b0101	300-400 MHz	4'b0110	400-500 MHz	4'b0111	500-600 MHz	4'b1000	600-700 MHz	4'b1001	700-800 MHz	4'b1010	800-1000 MHz	4'b1011	additional adjust	4'b1100	additional adjust	4'b1101	additional adjust	4'b1110	additional adjust
4'b0000	80-110 MHz																																
4'b0001	110-150 MHz																																
4'b0010	150-200 MHz																																
4'b0011	200-250 MHz																																
4'b0100	250-300 MHz																																
4'b0101	300-400 MHz																																
4'b0110	400-500 MHz																																
4'b0111	500-600 MHz																																
4'b1000	600-700 MHz																																
4'b1001	700-800 MHz																																
4'b1010	800-1000 MHz																																
4'b1011	additional adjust																																
4'b1100	additional adjust																																
4'b1101	additional adjust																																
4'b1110	additional adjust																																

MIPIPHY_REGA5

Address: Operational Base + offset (0x0294)
mipiphy register a5

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x05	<p>reg_hs_tlpix Data2 Lane The value of counter for HS Tlpix Time (\geqTlpix) $= Tpin_txbyteclkhs * value$</p>

MIPIPHY_REGA6

Address: Operational Base + offset (0x0298)
mipiphy register a6

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	<p>reg_hs_ths_prepare Data2 Lane The value of counter for HS Ths-prepare For clock lane, Ths-prepare(38ns~95ns) For data lane, Ths-prepare(40ns+4UI~85ns+6UI) $= Ttxddrclkhs * value$</p>

MIPIPHY_REGA7

Address: Operational Base + offset (0x029c)
 mipiphy register a7

Bit	Attr	Reset Value	Description																						
31:6	RO	0x0	reserved																						
5:0	RW	0x00	<p>reg_hs_the_zero Data2 Lane The value of counter for HS Ths-zero For clock lane, Ths-prepare+Ths-zero (>=300ns) For data lane, Ths-prepare+Ths-zero (>= 145 ns + 10*UI) = Tpin_txbyteclkhs*value For clock lane, S_HSTXTHSZERO[5:0] = 6'b100000 For data lane, S_HSTXTHSZERO[5:0] = 6'b001001 Frequency(1/UI) Value(Decimal)</p> <table> <tbody> <tr><td>80 -110 MHz</td><td>3</td></tr> <tr><td>110-150 MHz</td><td>4</td></tr> <tr><td>150-200 MHz</td><td>4</td></tr> <tr><td>200-250 MHz</td><td>5</td></tr> <tr><td>250-300 MHz</td><td>6</td></tr> <tr><td>300-400 MHz</td><td>7</td></tr> <tr><td>400-500 MHz</td><td>8</td></tr> <tr><td>500-600 MHz</td><td>10</td></tr> <tr><td>600-700 MHz</td><td>11</td></tr> <tr><td>700-800 MHz</td><td>12</td></tr> <tr><td>800-1000 MHz</td><td>15</td></tr> </tbody> </table>	80 -110 MHz	3	110-150 MHz	4	150-200 MHz	4	200-250 MHz	5	250-300 MHz	6	300-400 MHz	7	400-500 MHz	8	500-600 MHz	10	600-700 MHz	11	700-800 MHz	12	800-1000 MHz	15
80 -110 MHz	3																								
110-150 MHz	4																								
150-200 MHz	4																								
200-250 MHz	5																								
250-300 MHz	6																								
300-400 MHz	7																								
400-500 MHz	8																								
500-600 MHz	10																								
600-700 MHz	11																								
700-800 MHz	12																								
800-1000 MHz	15																								

MIPPHY_REGA8

Address: Operational Base + offset (0x02a0)
 mipiphy register a8

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description																								
6:0	RW	0x00	<p>reg_hs_ths_trail Data2 Lane The value of counter for HS Ths-trail For clock lane, Ths-trail (\geq60ns) For data lane, Ths-trail (\geqmax(8UI, 60ns+4UI)) = Tad_txddrclkhs_i* value For clock lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111 For data lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111</p> <table> <tr> <td>Frequency(1/UI)</td> <td>Value(Decimal)</td> </tr> <tr> <td>80 -110 MHz</td> <td>12</td> </tr> <tr> <td>110-150 MHz</td> <td>13</td> </tr> <tr> <td>150-200 MHz</td> <td>17</td> </tr> <tr> <td>200-250 MHz</td> <td>20</td> </tr> <tr> <td>250-300 MHz</td> <td>24</td> </tr> <tr> <td>300-400 MHz</td> <td>29</td> </tr> <tr> <td>400-500 MHz</td> <td>35</td> </tr> <tr> <td>500-600 MHz</td> <td>41</td> </tr> <tr> <td>600-700 MHz</td> <td>49</td> </tr> <tr> <td>700-800 MHz</td> <td>52</td> </tr> <tr> <td>800-1000 MHz</td> <td>64</td> </tr> </table>	Frequency(1/UI)	Value(Decimal)	80 -110 MHz	12	110-150 MHz	13	150-200 MHz	17	200-250 MHz	20	250-300 MHz	24	300-400 MHz	29	400-500 MHz	35	500-600 MHz	41	600-700 MHz	49	700-800 MHz	52	800-1000 MHz	64
Frequency(1/UI)	Value(Decimal)																										
80 -110 MHz	12																										
110-150 MHz	13																										
150-200 MHz	17																										
200-250 MHz	20																										
250-300 MHz	24																										
300-400 MHz	29																										
400-500 MHz	35																										
500-600 MHz	41																										
600-700 MHz	49																										
700-800 MHz	52																										
800-1000 MHz	64																										

MIPIPHY_REGA9

Address: Operational Base + offset (0x02a4)
mipiphy register a9

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>reg_hs_ths_exit Data2 Lane The value of counter for HS Ths-exit Ths-exit = Tpin_txbyteclkhs*value</p>

MIPIPHY_REGAA

Address: Operational Base + offset (0x02a8)
mipiphy register aa

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>reg_hs_tclk_post Data2 Lane The value of counter for HS Tclk-post Tclk-post = Tpin_txbyteclkhs*value</p>

MIPIPHY_REGAB

Address: Operational Base + offset (0x02ac)

mipi phy register ab

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reserved Data2 Lane

MIPIPHY_REGAC

Address: Operational Base + offset (0x02b0)

mipi phy register ac

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	reg_hs_twakeup Data2 Lane The value[9:8] of counter for HS Twakeup also see REGAD

MIPIPHY_REGAD

Address: Operational Base + offset (0x02b4)

mipi phy register ad

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reg_hs_twakeup Data2 Lane The value[7:0] of counter for HS Twakeup Twakeup for ulpm, Twakeup = Tpin_sys_clk * value[9:0]

MIPIPHY_REGAE

Address: Operational Base + offset (0x02b8)

mipi phy register ae

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	reg_hs_tclk_pre Data2 Lane The value of counter for HS Tclk-pre Tclk-pre for clock lane Tclk-pre = Tpin_txbyteclkhs * value

MIPIPHY_REGBO

Address: Operational Base + offset (0x02c0)

mipi phy register b0

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	reg_hs_tta_go Data2 Lane The value of counter for HS Tta-go Tta-go for turnaround Tta-go $= \text{Txclkesc} * \text{value}$

MIPIPHY_REGB1

Address: Operational Base + offset (0x02c4)

mipiphy register b1

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_tta_sure Data2 Lane The value of counter for HS Tta-sure Tta-sure for turnaround Tta-sure $= \text{Txclkesc} * \text{value}$

MIPIPHY_REGB2

Address: Operational Base + offset (0x02c8)

mipiphy register b2

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_tta_wait Data2 Lane The value of counter for HS Tta-wait Tta-wait for turnaround Interval from receiving ppi turnaround request to sending esc request. Tta-wait $= \text{Txclkesc} * \text{value}$

MIPIPHY_REGC0

Address: Operational Base + offset (0x0300)

mipiphy register c0

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description																														
3:0	RW	0xb	<p>reg_ths_settle Data3 Lane Configure the count time of the THS-SETTLE by protocol. After the count done, D-PHY will begin to receive the high speed data. (Can be configured from 4'h0 to 4'hd)</p> <table> <tr><td>4'b0000</td><td>80-110 MHz</td></tr> <tr><td>4'b0001</td><td>110-150 MHz</td></tr> <tr><td>4'b0010</td><td>150-200 MHz</td></tr> <tr><td>4'b0011</td><td>200-250 MHz</td></tr> <tr><td>4'b0100</td><td>250-300 MHz</td></tr> <tr><td>4'b0101</td><td>300-400 MHz</td></tr> <tr><td>4'b0110</td><td>400-500 MHz</td></tr> <tr><td>4'b0111</td><td>500-600 MHz</td></tr> <tr><td>4'b1000</td><td>600-700 MHz</td></tr> <tr><td>4'b1001</td><td>700-800 MHz</td></tr> <tr><td>4'b1010</td><td>800-1000 MHz</td></tr> <tr><td>4'b1011</td><td>additional adjust</td></tr> <tr><td>4'b1100</td><td>additional adjust</td></tr> <tr><td>4'b1101</td><td>additional adjust</td></tr> <tr><td>4'b1110</td><td>additional adjust</td></tr> </table>	4'b0000	80-110 MHz	4'b0001	110-150 MHz	4'b0010	150-200 MHz	4'b0011	200-250 MHz	4'b0100	250-300 MHz	4'b0101	300-400 MHz	4'b0110	400-500 MHz	4'b0111	500-600 MHz	4'b1000	600-700 MHz	4'b1001	700-800 MHz	4'b1010	800-1000 MHz	4'b1011	additional adjust	4'b1100	additional adjust	4'b1101	additional adjust	4'b1110	additional adjust
4'b0000	80-110 MHz																																
4'b0001	110-150 MHz																																
4'b0010	150-200 MHz																																
4'b0011	200-250 MHz																																
4'b0100	250-300 MHz																																
4'b0101	300-400 MHz																																
4'b0110	400-500 MHz																																
4'b0111	500-600 MHz																																
4'b1000	600-700 MHz																																
4'b1001	700-800 MHz																																
4'b1010	800-1000 MHz																																
4'b1011	additional adjust																																
4'b1100	additional adjust																																
4'b1101	additional adjust																																
4'b1110	additional adjust																																

MIPIPHYS_REGC5

Address: Operational Base + offset (0x0314)
mipiphy register c5

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x05	<p>reg_hs_tlpix Data3 Lane The value of counter for HS Tlpix Time (\geqTlpix) $= Tpin_txbyteclkhs * value$</p>

MIPIPHYS_REGC6

Address: Operational Base + offset (0x0318)
mipiphy register c6

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	<p>reg_hs_ths_prepare Data3 Lane The value of counter for HS Ths-prepare For clock lane, Ths-prepare(38ns~95ns) For data lane, Ths-prepare(40ns+4UI~85ns+6UI) $= Ttxddrclkhs * value$</p>

MIPIPHYS_REGC7

Address: Operational Base + offset (0x031c)
 mipiphy register c7

Bit	Attr	Reset Value	Description																						
31:6	RO	0x0	reserved																						
5:0	RW	0x00	<p>reg_hs_the_zero Data3 Lane The value of counter for HS Ths-zero For clock lane, Ths-prepare+Ths-zero (>=300ns) For data lane, Ths-prepare+Ths-zero (>= 145 ns + 10*UI) = Tpin_txbyteclkhs*value For clock lane, S_HSTXTHSZERO[5:0] = 6'b100000 For data lane, S_HSTXTHSZERO[5:0] = 6'b001001 Frequency(1/UI) Value(Decimal)</p> <table> <tbody> <tr><td>80 -110 MHz</td><td>3</td></tr> <tr><td>110-150 MHz</td><td>4</td></tr> <tr><td>150-200 MHz</td><td>4</td></tr> <tr><td>200-250 MHz</td><td>5</td></tr> <tr><td>250-300 MHz</td><td>6</td></tr> <tr><td>300-400 MHz</td><td>7</td></tr> <tr><td>400-500 MHz</td><td>8</td></tr> <tr><td>500-600 MHz</td><td>10</td></tr> <tr><td>600-700 MHz</td><td>11</td></tr> <tr><td>700-800 MHz</td><td>12</td></tr> <tr><td>800-1000 MHz</td><td>15</td></tr> </tbody> </table>	80 -110 MHz	3	110-150 MHz	4	150-200 MHz	4	200-250 MHz	5	250-300 MHz	6	300-400 MHz	7	400-500 MHz	8	500-600 MHz	10	600-700 MHz	11	700-800 MHz	12	800-1000 MHz	15
80 -110 MHz	3																								
110-150 MHz	4																								
150-200 MHz	4																								
200-250 MHz	5																								
250-300 MHz	6																								
300-400 MHz	7																								
400-500 MHz	8																								
500-600 MHz	10																								
600-700 MHz	11																								
700-800 MHz	12																								
800-1000 MHz	15																								

MIPPHY_REGC8

Address: Operational Base + offset (0x0320)
 mipiphy register c8

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description																								
6:0	RW	0x00	<p>reg_hs_ths_trail Data3 Lane The value of counter for HS Ths-trail For clock lane, Ths-trail (\geq60ns) For data lane, Ths-trail (\geqmax(8UI, 60ns+4UI)) = Tad_txddrclkhs_i* value For clock lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111 For data lane, S_HXTXTHSTRAIL[6:0] = 7'b0100111</p> <table> <tr> <td>Frequency(1/UI)</td> <td>Value(Decimal)</td> </tr> <tr> <td>80 -110 MHz</td> <td>12</td> </tr> <tr> <td>110-150 MHz</td> <td>13</td> </tr> <tr> <td>150-200 MHz</td> <td>17</td> </tr> <tr> <td>200-250 MHz</td> <td>20</td> </tr> <tr> <td>250-300 MHz</td> <td>24</td> </tr> <tr> <td>300-400 MHz</td> <td>29</td> </tr> <tr> <td>400-500 MHz</td> <td>35</td> </tr> <tr> <td>500-600 MHz</td> <td>41</td> </tr> <tr> <td>600-700 MHz</td> <td>49</td> </tr> <tr> <td>700-800 MHz</td> <td>52</td> </tr> <tr> <td>800-1000 MHz</td> <td>64</td> </tr> </table>	Frequency(1/UI)	Value(Decimal)	80 -110 MHz	12	110-150 MHz	13	150-200 MHz	17	200-250 MHz	20	250-300 MHz	24	300-400 MHz	29	400-500 MHz	35	500-600 MHz	41	600-700 MHz	49	700-800 MHz	52	800-1000 MHz	64
Frequency(1/UI)	Value(Decimal)																										
80 -110 MHz	12																										
110-150 MHz	13																										
150-200 MHz	17																										
200-250 MHz	20																										
250-300 MHz	24																										
300-400 MHz	29																										
400-500 MHz	35																										
500-600 MHz	41																										
600-700 MHz	49																										
700-800 MHz	52																										
800-1000 MHz	64																										

MIPIPHY_REGC9

Address: Operational Base + offset (0x0324)
mipiphy register c9

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>reg_hs_ths_exit Data3 Lane The value of counter for HS Ths-exit Ths-exit = Tpin_txbyteclkhs*value</p>

MIPIPHY_REGCA

Address: Operational Base + offset (0x0328)
mipiphy register ca

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>reg_hs_tclk_post Data3 Lane The value of counter for HS Tclk-post Tclk-post = Tpin_txbyteclkhs*value</p>

MIPIPHY_REGCB

Address: Operational Base + offset (0x032c)

mipiphy register cb

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reserved Data3 Lane

MIPIPHY_REGCC

Address: Operational Base + offset (0x0330)

mipiphy register cc

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	reg_hs_twakup Data3 Lane The value[9:8] of counter for HS Twakup also see REGCD

MIPIPHY_REGCD

Address: Operational Base + offset (0x0334)

mipiphy register cd

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	reg_hs_twakup Data3 Lane The value[7:0] of counter for HS Twakup Twakup for ulpm, Twakup = Tpin_sys_clk*value[9:0]

MIPIPHY_REGCE

Address: Operational Base + offset (0x0338)

mipiphy register ce

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	reg_hs_tclk_pre Data3 Lane The value of counter for HS Tclk-pre Tclk-pre for clock lane Tclk-pre = Tpin_txbyteclkhs*value

MIPIPHY_REGDO

Address: Operational Base + offset (0x0340)

mipiphy register d0

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	reg_hs_tta_go Data3 Lane The value of counter for HS Tta-go Tta-go for turnaround Tta-go $= \text{Txclkesc} * \text{value}$

MIPIPHY_REGD1

Address: Operational Base + offset (0x0344)

miiphy register d1

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_tta_sure Data3 Lane The value of counter for HS Tta-sure Tta-sure for turnaround Tta-sure $= \text{Txclkesc} * \text{value}$

MIPIPHY_REGD2

Address: Operational Base + offset (0x0348)

miiphy register d2

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	reg_hs_tta_wait Data3 Lane The value of counter for HS Tta-wait Tta-wait for turnaround Interval from receiving ppi turnaround request to sending esc request. Tta-wait $= \text{Txclkesc} * \text{value}$

MIPIPHY_REGE0

Address: Operational Base + offset (0x0380)

miiphy register e0

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x1	mipi_mode_en 1: enable mipi mode 0: disable mipi mode
6	RW	0x0	ttl_mode_en 1: enable ttl mode 0: disable ttl mode
5	RW	0x0	lvds_mode_en 1: enable lvds mode 0: disable lvds mode

Bit	Attr	Reset Value	Description
4:3	RW	0x00	reserved
2	RW	0x1	reg_rstn reset the LVDS PHY configuration 1: none 0: reset
1:0	RW	0x00	reserved

MIPIPHYS_REGEA

Address: Operational Base + offset (0x03A8)
mipiphy register ea

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x1	lvds_pllpd lvds pll power down 1: power down 0: power up
1	RW	0x0	reserved
0	RW	0x0	lvds_bgpd lvds bandgap power down 1: power down 0: power up

46.5 Interface Timing

This section shows a PPI timing relationship at high-speed transmission. While pin_txrequesths is low, the Lane Module ignores the value of pin_txdatahs. To begin the transmission, the protocol drives pin_txdatahs with the first byte of data and asserts pin_txrequesths. This data byte is accepted by the PHY on the first rising edge of pin_txbyteclkhs with pin_txreadyhs also asserted. At this point, the protocol logic drives the next data byte onto pin_txdatahs. After every rising clock cycle with pin_txreadyhs active, the protocol supplies a new valid data byte or ends the transmission. After the last data byte has been transferred to the Lane Module, pin_txrequesths is driven low to cause the Lane Module to stop the transmission and enter Stop state. The minimum number of bytes transmitted could be as small as one.

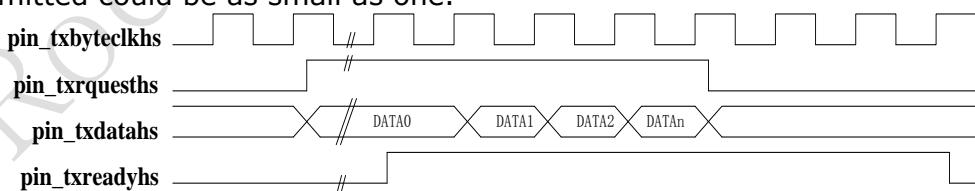


Fig.46-3HS-TX PPI Timing

This section shows a PPI timing relationship at low-power data transmission operation. The Protocol directs the Data Lane to enter Low-Power data transmission Escape mode by asserting pin_txrequestesc with pin_txlpdtesc high. The Low-Power transmit data is transferred on the pin_txdataesc lines when pin_txvalidesc and pin_txreadyesc are both active at a rising edge of pin_txclkesc.

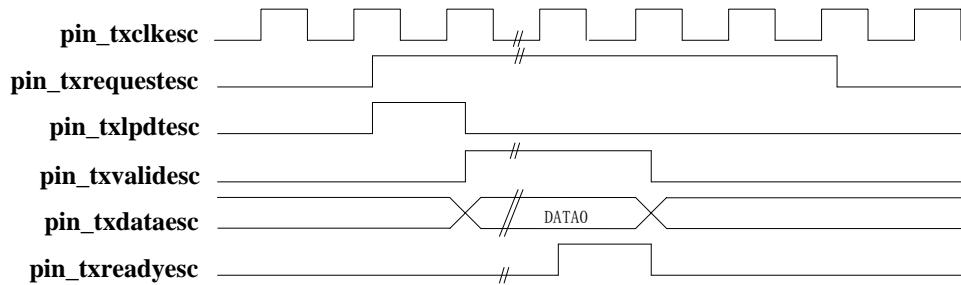


Fig.46-4LPDT TX PPI Timing

This section shows a PPI timing relationship at low-power data reception. The signal pin_rxlpdtesc is asserted when the escape entry command is detected and stays high until the Lane returns to stop state, indicating that the transmission has finished.

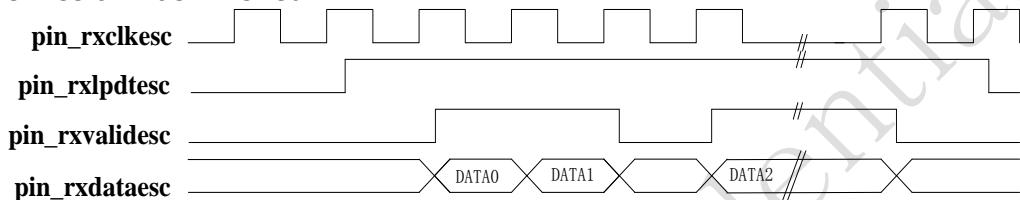


Fig.46-5LPDT RX PPI Timing

46.6 Application Notes

46.6.1 Low power mode

Low Power Mode is a special feature for D-PHY. You can control this function by using proper registers from the Innosilicon D-PHY with few operations. The following is a step by step instruction for low power mode in and out.

Low Power in Steps:

- Step1: Send 0x01 to register 0x00. Disable all lanes on analog part.
- Step2: Send 0xe3 to register 0x01. Disable PLL and LDO.
- Step3: Wait a period before reference clock have been disabled.
- Step4: Disable reference clock.

Low Power out Steps:

- Step1: Enable reference clock.
- Step2: Wait a period after reference clock have been enabled.
- Step3: Send 0xe4 to register 0x01. Enable PLL and LDO.
- Step4: Send 0x7d to register 0x00. Enable all lanes on analog part.
- Step5: Send 0xe0 to register 0x01. Reset analog.
- Step6: Wait a period after analog has been reset.
- Step7: Send 0x1e to register 0x20. Reset digital.
- Step8: Send 0x1f to register 0x20. Reset digital.
- Step9: Wait a period before normal transmission.

46.6.2 Programmable PLL IN DSI TX

Frequency Calculating Formula

The PLL output frequency can be calculated using a simple formula:

PLL_Output_Frequency = FREF/PREDIV*FBDIV

PLL_Output_Frequency: It is equal to DDR- Clock-Frequency * 2

FREF :PLL input reference frequency which equals to the frequency of the pin_clkhtref

PREDIV : PLL input reference clock divider which can be configured by the register of reg_prediv

FBDIV :Integer value programmed into feedback divider which can be configured by the register of reg_fbdv

For example,

FREF =20MHz, PLL output frequency = 800Hz, so set PREDIV=1, FBDIV=40

Additional Programming Considerations

1. The divided reference frequency (FREF/PREDIV) should be less than 40MHz.
2. The all possible settings of feedback divider are 12,13,14,16~511.

46.6.3 LVDS mode

Lvds source from LCDC0 or LCDC1.

Step1: configure lvds_format

8bit mode format-1 : GRF_LVDS_CON0[2:1]=0x00;

8bit mode format-2 : GRF_LVDS_CON0[2:1]=0x01;

8bit mode format-3 : GRF_LVDS_CON0[2:1]=0x10;

6bit mode : GRF_LVDS_CON0[2:1]=0x11;

MSB is on D0 : GRF_LVDS_CON0[3]=0x0;

MSB is on D7 : GRF_LVDS_CON0[3]=0x1;

Step3: configure MIPI-PHY

Configure PLL:

MIPIPHY_REG3=0x1;

MIPIPHY_REG4=0x7;

Configure LVDS Interface:

MIPIPHY_REGE0=0x25;

MIPIPHY_REGEA=0xf8;

Step4: enable lvds

GRF_LVDS_CON0[6]=0x1;

46.6.4 Other mipi_phy grf

Other mipi_phy control registers description reference to GRF_LVDS_CON0.

46.7 ELECTRICAL SPECIFICATIONS

46.7.1 DC SPECIFICATIONS

Table 46-1 HS Transmitter DC specifications

Parameter	Description	Min	Nom	Max	Unit	Note
V _{CMTX}	HS TX staticCommon-mode voltage	150	200	250	mV	1
ΔV _{CMTX(1,0)}	V _{CMTX} mismatch when output is Differential-1 or Differential-0			5	mV	2
V _O	HS transmit differential voltage	140	200	270	mV	1
ΔV _O	V _O mismatch when output is Differential-1 or Differential-0			10	mV	2
V _{OHHS}	HS output high voltage			360	mV	1
Z _{os}	Single ended output impedance	40	50	62.5	ohm	
Δ Z _{os}	Single ended output impedance mismatch			10	%	

- Value when driving into load impedance anywhere in the ZID range.
- It is recommended the implementer minimize ΔV_O and ΔV_{CMTX(1,0)} in order to minimize radiation and optimize signal integrity.

Table 46-2 HS Transmitter DC specifications

Parameter	Description	Min	Nom	Max	Unit	Note
V _{IH}	Logic 1 input voltage	880			mV	
V _{IL}	Logic 0 input voltage, not in ULPState			550	mV	
V _{IL-ULPS}	Logic 0 input voltage, ULP State			300	mV	
V _{HYST}	Input hysteresis	25			mV	

Table 46-3 LP Transmitter DC Specifications

Parameter	Description	Min	Nom	Max	Unit	Note
V _{OH}	The venin output high level	1.1	1.2	1.3	V	

VOL	The venin output low level	-50		50	mV	
ZOLP	Output impedance of LP transmitter	110			Ω	1

1. Though no maximum value for ZOLP is specified, the LP transmitter output impedance shall ensure theTRLP/TFLP specification is met.

46.7.2 AC specifications

Table 46-4 HS receiver AC specifications

Parameter	Description	Min	Nom	Max	Unit	Note
$\Delta V_{CMRX(HF)}$	Common-mode interference beyond 450 MHz			100	mV	2
$\Delta V_{CMRX(LF)}$	Common-mode interference 50MHz – 450MHz	-50		50	mV	1,4
C _{CM}	Common-mode termination			60	pF	3

1. Excluding 'static' ground shift of 50mV
2. $\Delta V_{CMRX(HF)}$ is the peak amplitude of a sine wave superimposed on the receiver inputs.
3. For higher bit rates a 14pF capacitor will be needed to meet the common-mode return loss specification.
4. Voltage difference compared to the DC average common-mode potential.

Table 46-5 LP receiver AC specifications

Parameter	Description	Min	Nom	Max	Unit	Note
e_{SPIKE}	Input pulse rejection			300	V.ps	1, 2,3
T _{MIN-RX}	Minimum pulse width response	20			ns	4
V _{INT}	Peak interference amplitude			200	mV	
f _{INT}	Interference frequency	450			MHz	

1. Time-voltage integration of a spike above VIL when being in LP-0 state or below VIH when being in LP-1 state
2. An impulse less than this will not change the receiver state.
3. In addition to the required glitch rejection, implementers shall ensure rejection of known RF-interferers.
4. An input pulse greater than this shall toggle the output.

Table 46-6 LP Transmitter AC specifications

Parameter	Description	Min	Nom	Max	Unit	Note
T _{RLP/TFLP}	15%-85% rise time and fall time			25	ns	1

TREOT	30%-85% rise time and fall time				35	ns	1,5,6
TLP-PULSE-TX	Pulse width of exclusive-O R clock the LP	First LP exclusive-OR clock pulse after Stop state or last pulse before Stop state	40			ns	4
	All other pulses		20				4
TLP-PER-TX	Period of the LP exclusive-OR clock		90			ns	
$\delta V/\delta t_{SR}$	Slew rate @ CLOAD = 0pF				500	mV/n s	1,3,7,8
	Slew rate @ CLOAD = 5pF				300	mV/n s	1,3,7,8
	Slew rate @ CLOAD = 20pF				250	mV/n s	1,3,7,8
	Slew rate @ CLOAD = 70pF				150	mV/n s	1,3,7,8
	Slew rate @ CLOAD = 0 to 70pF(Falling Edge Only)		30			mV/n s	1,2,3
	Slew rate @ CLOAD = 0 to 70pF(Rising Edge Only)		30			mV/n s	1,3,9
	Slew rate @ CLOAD = 0 to 70pF(Rising Edge Only)		30-0.07 5 * (VO,INST - 700)			mV/n s	1,10,11
CLOAD	Load capacitance		0		70	pF	1

1. CLOAD includes the low-frequency equivalent transmission line capacitance. The capacitance of TX and RX are assumed to always be <10pF. The distributed line capacitance can be up to 50pF for a transmission line with 2ns delay.
2. When the output voltage is between 400 mV and 930 mV.
3. Measured as average across any 50 mV segment of the output signal transition.
4. This parameter value can be lower than TLPX due to differences in rise vs. fall signal slopes and trip levels and mismatches between Dp and Dn LP transmitters. Any LP exclusive-OR pulse observed during HS EoT (transition from HS level to LP-11) is glitch behavior as described in section 8.2.2.
5. The rise-time of TREOT starts from the HS common-level at the moment the differential amplitude drops below 70mV, due to stopping the differential drive.
6. With an additional load capacitance CCM between 0 and 60pF on the termination center tap at RX side of the Lane
7. This value represents a corner point in a piecewise linear curve.
8. When the output voltage is in the range specified by VPIN(absmax).
9. When the output voltage is between 400 mV and 700 mV.
10. Where VO,INST is the instantaneous output voltage, VDP or VDN, in millivolts.

-
11. When the output voltage is between 700 mV and 930 mV.

Rockchip Confidential

Chapter 47 I2S 8-channel

47.1 Overview

The I2S/PCM controller is designed for interfacing between the AHB bus and the I2S bus.

The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and be invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.

Devices often use the I2S bus are ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

47.1.1 Features

Not only I2S but also PCM mode surround audio output (up to 7.1channel) and stereo input are supported in I2S/PCM controller.

- Support five internal 32-bit wide and 32-location deep FIFOs, four for transmitting and one for receiving audio data
- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support 2,4,6,8 channels audio transmitting in I2S and PCM mode
- Support 2 channels audio receiving in I2S and PCM mode
- Support up to 192kHz sample rate
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 3 independent LRCK signals, one for receiving and two for transmitting audio data
- Support configurable SCLK and LRCK polarity
- Support SCLK is equivalent to MCLK divided by an even number range from 2 to 64 in master mode

47.2 Block Diagram

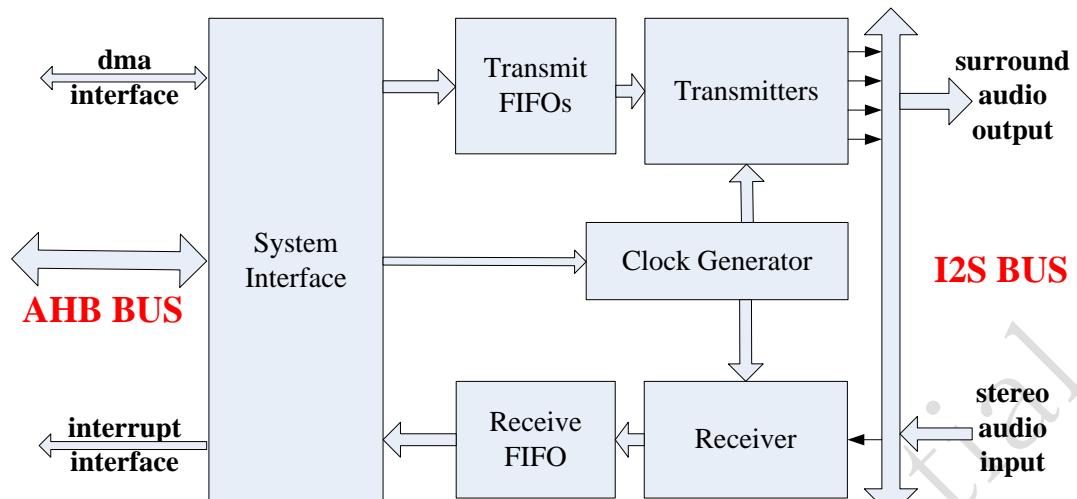


Fig.47-1 I2S/PCM controller (8 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshake interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either master or slave, with I2S or PCM mode surround (up to 7.1 channel) serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either master or slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFOs

The Transmit FIFOs are the buffer to store transmitted audio data. Each of the size of the four FIFOs is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

47.3 Function description

In the I2S/PCM controller, there are four conditions: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

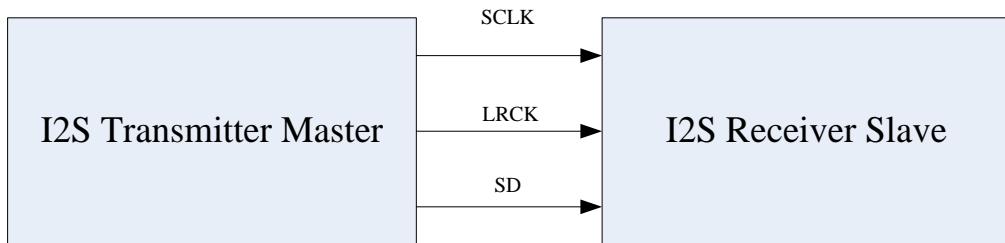


Fig.47-2 I2S transmitter-master & receiver-slave condition

When transmitter acts as a master, it sends all signals to receiver (slave), and CPU control when to send clock and data to the receiver. When acting as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from receiver (master) to transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when to send data.

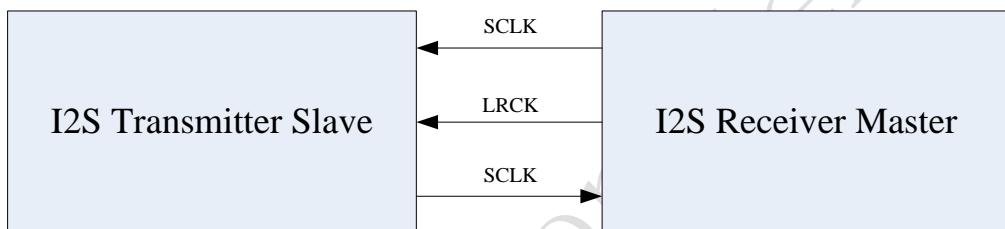


Fig.47-3 I2S transmitter-slave& receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

47.3.1 i2s normal mode

This is the waveform of I2S normal mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx0) signal, it goes low to indicate left channel and high to right channel. For SD (i2s_sdo0, i2s_sdo1, i2s_sdo2, i2s_sdo3, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

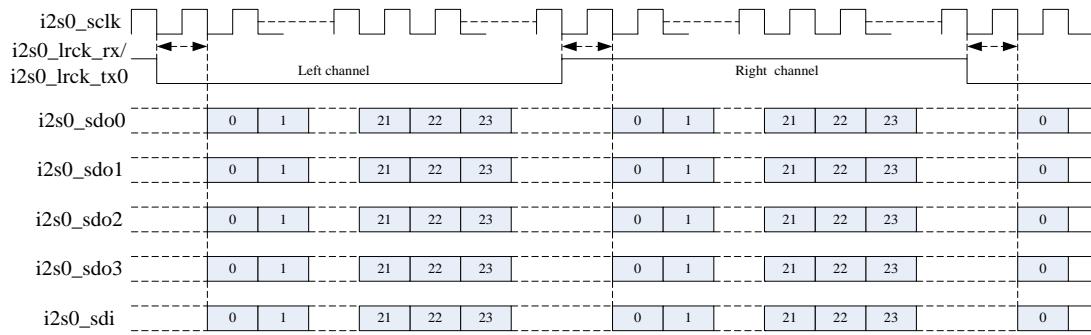


Fig.47-4 I2S normal mode timing format

47.3.2 i2s left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo0, i2s_sdo1, i2s_sdo2, i2s_sdo3, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

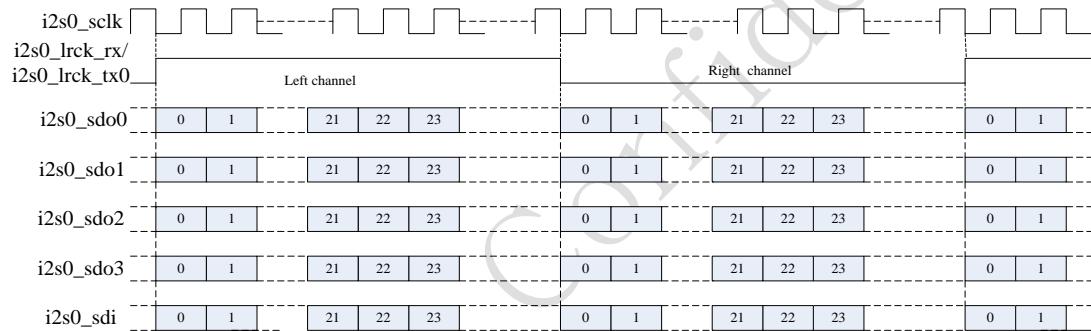


Fig.47-5 I2S left justified mode timing format

47.3.3 i2s right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo0, i2s_sdo1, i2s_sdo2, i2s_sdo3, i2s_sdi) signal, it transfers MSB or LSB first; but different from I2S normal or left justified mode, its data is aligned to last bit at the edge of the LRCK signal. The range of SD signal width is from 16 to 32bits.

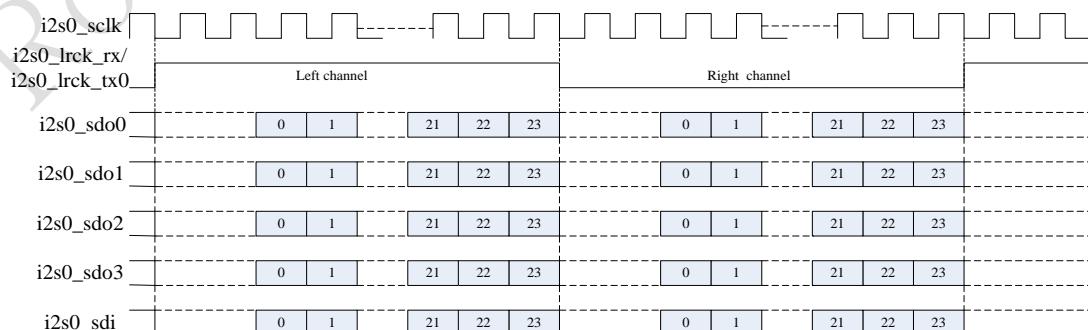


Fig.47-6 I2S right justified mode timing format

47.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo0, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

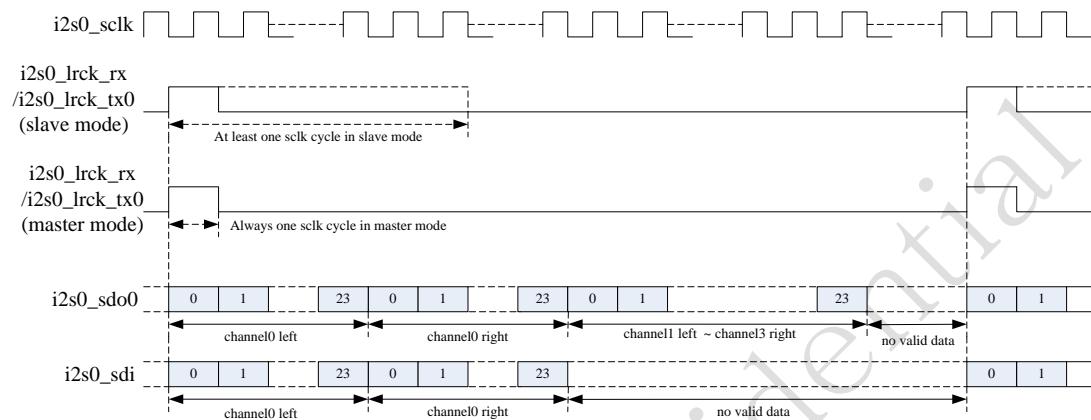


Fig.47-7 PCM early modetiming format

47.3.5 PCM late1 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo0, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

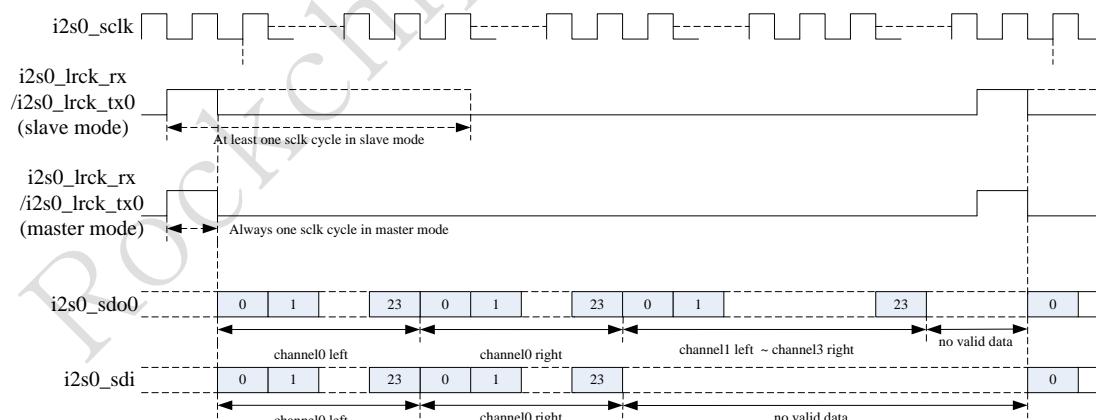


Fig.47-8 PCM late1 modetiming format

47.3.6 PCM late2 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo0, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

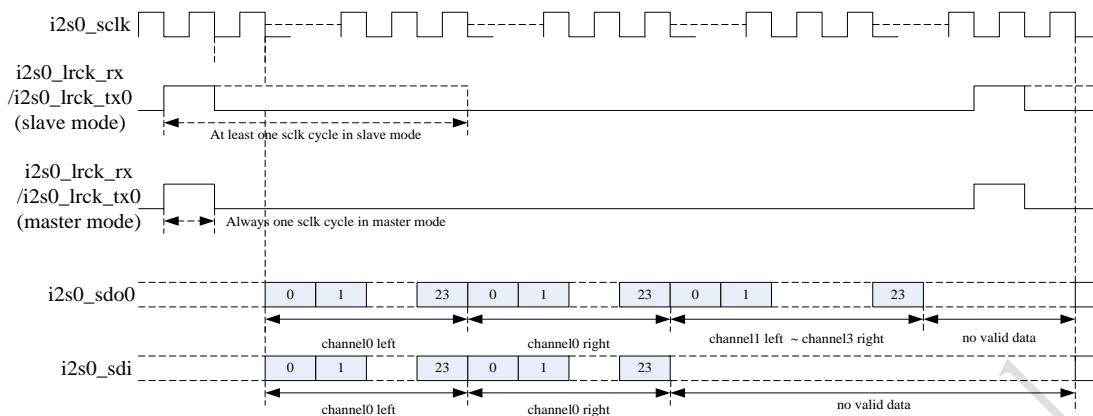


Fig.47-9 PCM late2 modetiming format

47.3.7 PCM late3 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo0, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

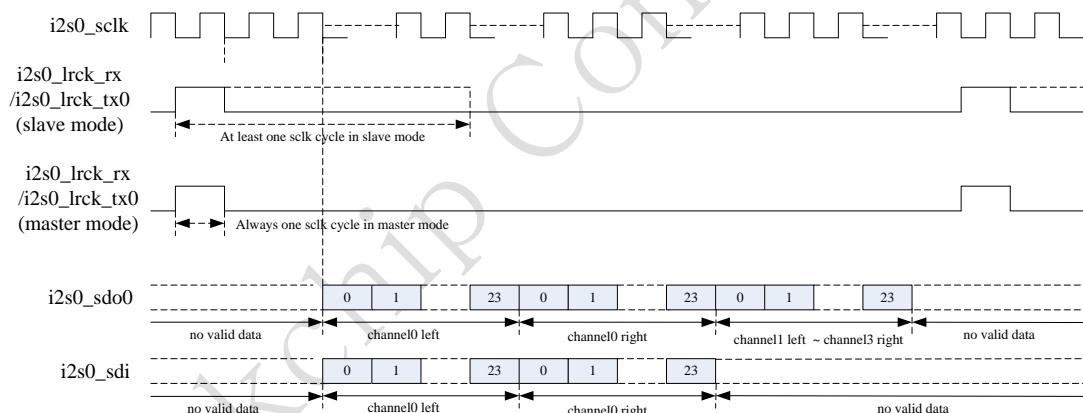


Fig.47-10 PCM late3 modetiming format

47.4 Register Description

This section describes the control/status registers of the design.

47.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_TXCR	0x0000	W	0x0000000f	transmit operation control register
I2S_RXCR	0x0004	W	0x0000000f	receive operation control register
I2S_CKR	0x0008	W	0x00071f1f	clock generation register
I2S_FIFOLR	0x000c	W	0x00000000	FIFO level register
I2S_DMACR	0x0010	W	0x001f0000	DMA control register
I2S_INTCR	0x0014	W	0x01f00000	interrupt control register
I2S_INTSR	0x0018	W	0x00000000	interrupt status register
I2S_XFER	0x001c	W	0x00000000	Transfer Start Register

Name	Offset	Size	Reset Value	Description
I2S_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2S_TXDR	0x0024	W	0x00000000	Transimt FIFO Data Register
I2S_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

47.4.2 Detail Register Description

I2S_TXCR

Address: Operational Base + offset (0x0000)
transmit operation control register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT right jusitified counter (Can be written only when XFER[0] bit is 0.) Only vailid in I2S Right justified format and slave tx mode is selected. Start to tramsmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	CSR Channel select register (Can be written only when XFER[0] bit is 0.) 0:channel 0 enable 1:channel 0 & channel 1 enable 2:channel 0 & channel 1 & channel 2 enable 3:channel 0 & channel 1 & channel 2 & channel 3 enable
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[0] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified

Bit	Attr	Reset Value	Description
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[0] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[0] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[0] bit is 0.) 0: I2S format 1: PCM format
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[0] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit 28:29bit 29:30bit 30:31bit 31:32bit

I2S_RXCR

Address: Operational Base + offset (0x0004)
receive operation control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[1] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[1] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[1] bit is 0.) 0:i2s 1:pcm

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit 28:29bit 29:30bit 30:31bit 31:32bit

I2S_CKR

Address: Operational Base + offset (0x0008)
 clock generation register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	MSS Master/slave mode select (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:master mode(sclk output) 1:slave mode(sclk input)
26	RW	0x0	CKP Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedgesclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedgesclk
25	RW	0x0	RLP Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: normal polartiy (I2S normal: low for left channel, high for right channel) I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1: oppsite polarity (I2S normal: high for left channel, low for right channel) I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)

Bit	Attr	Reset Value	Description
24	RW	0x0	<p>TLP Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: normal polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1: oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)</p>
23:16	RW	0x07	<p>MDIV mclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Serial Clock Divider = Fmclk / Ftxsclk-1.(mclkfrequency / txaclk frequency-1) 0 :Fmclk=Ftxsclk; 1 :Fmclk=2*Ftxsclk; 2,3 :Fmclk=4*Ftxsclk; 4,5 :Fmclk=6*Ftxsclk; 60,61:Fmclk=62*Ftxsclk; 62,63:Fmclk=64*Ftxsclk; 252,253:Fmclk=254*Ftxsclk; 254,255:Fmclk=256*Ftxsclk;</p>
15:8	RW	0x1f	<p>RSD Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Receive sclk divider= Fsclk/Frxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs 253: 254fs 254: 255fs 255: 256fs</p>

Bit	Attr	Reset Value	Description
7:0	RW	0x1f	<p>TSD Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Transmit sclk divider=Ftxsclk/Ftxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs 253: 254fs 254: 255fs 255: 256fs</p>

I2S_FIFOLR

Address: Operational Base + offset (0x000c)

FIFO level register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	RO	0x00	RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO.
23:18	RO	0x00	TFL3 Transmit FIFO3 Level Contains the number of valid data entries in the transmit FIFO3.
17:12	RO	0x00	TFL2 Transmit FIFO2 Level Contains the number of valid data entries in the transmit FIFO2.
11:6	RO	0x00	TFL1 Transmit FIFO1 Level Contains the number of valid data entries in the transmit FIFO1.
5:0	RO	0x00	TFL0 Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO0.

I2S_DMACR

Address: Operational Base + offset (0x0010)

DMA control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 0 : Receive DMA disabled 1 : Receive DMA enabled

Bit	Attr	Reset Value	Description
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if CSR=00;TXFIFO1 if CSR=01,TXFIFO2 if CSR=10,TXFIFO3 if CSR=11)is equal to or below this field value.

I2S_INTCR

Address: Operational Base + offset (0x0014)

interrupt control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x1f	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX overrun interrupt enable 0:disable 1:enable
16	RW	0x0	RXFIE RX full interrupt enable 0:disable 1:enable
15:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO (TXFIFO0 if CSR=00; TXFIFO1 if CSR=01, TXFIFO2 if CSR=10, TXFIFO3 if CSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC TX underrun interrupt clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX underrun interrupt enable 0:disable 1:enable
0	RW	0x0	TXEIE TX empty interrupt enable 0:disable 1:enable

I2S_INTSR

Address: Operational Base + offset (0x0018)
interrupt status register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI RX overrun interrupt 0:inactive 1:active
16	RO	0x0	RXFI RX full interrupt 0:inactive 1:active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX underrun interrupt 0:inactive 1:active
0	RO	0x0	TXEI TX empty interrupt 0:inactive 1:active

I2S_XFER

Address: Operational Base + offset (0x001c)
Transfer Start Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	RXS RX Transfer start bit 0:stop RX transfer. 1:start RX transfer
0	RW	0x0	TXS TX Transfer start bit 0:stop TX transfer. 1:start TX transfer

I2S_CLR

Address: Operational Base + offset (0x0020)

SCLK domain logic clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC RX logic clear This is a self cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX logic clear This is a self cleared bit. Write 1 to clear all transmit logic.

I2S_TXDR

Address: Operational Base + offset (0x0400~0x7FC)

Transmit FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transmit FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2S_RXDR

Address: Operational Base + offset (0x0800~0xBFC)

Receive FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR Receive FIFO Data Register When the register is read, data in the receive FIFO is accessed.

47.5 Application Notes

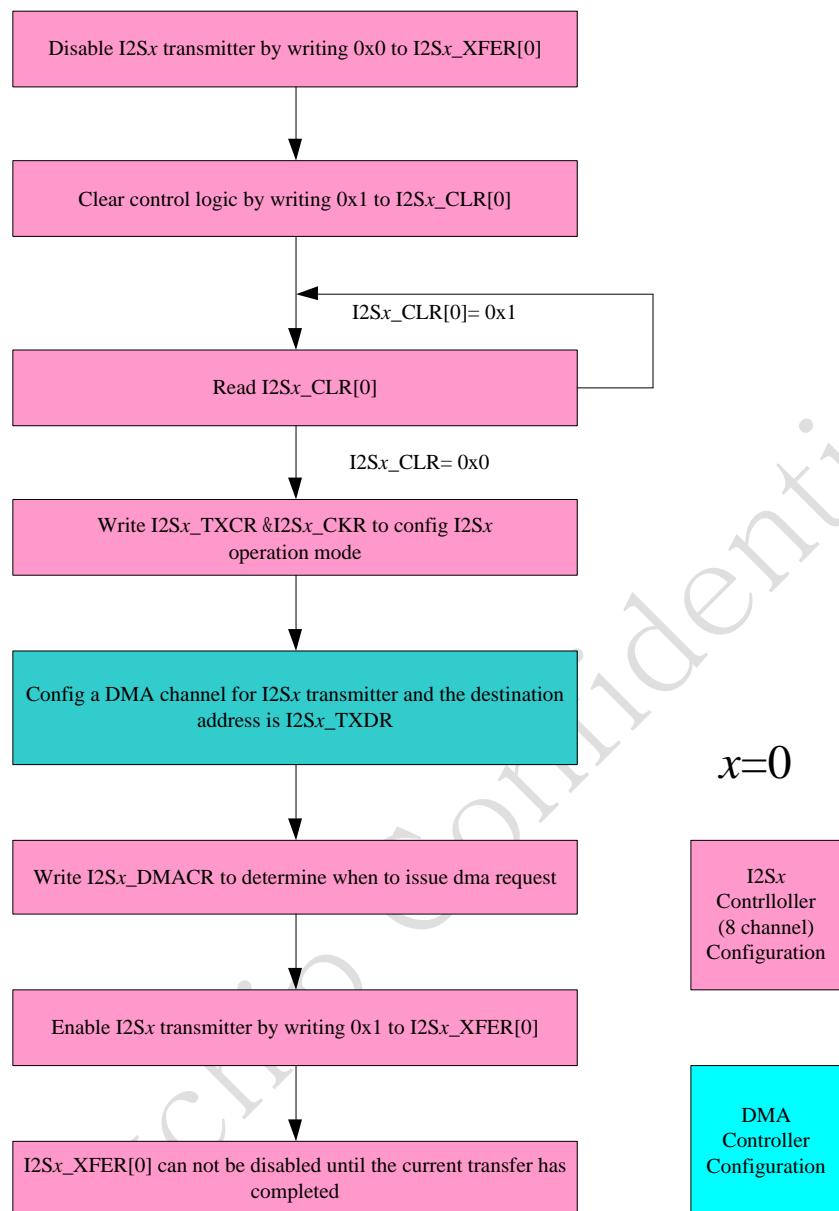


Fig.47-11 I2S/PCM controller (8 channel) transmit operation flow chart

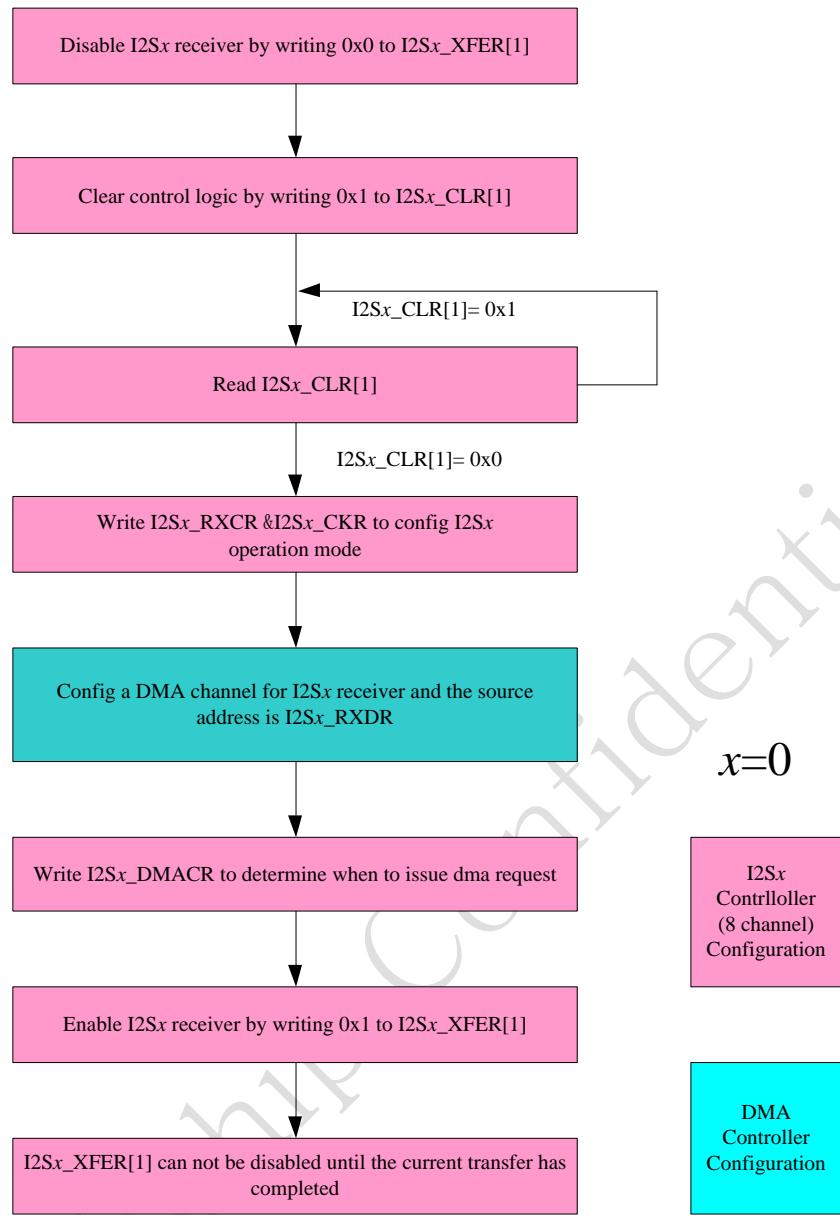


Fig.47-12 I2S/PCM controller (8 channel) receive operation flow chart