



GPL32900A Programming Guide

Preliminary V0.1 – Dec. 21, 2010

**Important Notice**

Generalplus Technology reserves the right to change this documentation without prior notice. Information provided by Generalplus Technology is believed to be accurate and reliable. However, Generalplus Technology makes no warranty for any errors which may appear in this document. Contact Generalplus Technology to obtain the latest version of device specifications before placing your order. No responsibility is assumed by Generalplus Technology for any infringement of patent or other rights of third parties which may result from its use. In addition, Generalplus products are not authorized for use as critical components in life support devices/systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Generalplus.

Table of Content

	PAGE
1 CONFIRMATION SHEET.....	16
2 INTRODUCTION.....	17
2.1 General Description	17
2.2 Significant Features	17
2.3 Block Diagram.....	19
2.4 Applications.....	19
3 SYSTEM CONTROL.....	20
3.1 Introduction	20
3.2 Features	20
3.3 Register Summary	20
3.4 Register Definition.....	22
3.4.1 SCUA.....	22
3.4.2 SCUB.....	39
3.4.3 SCUC.....	59
3.4.4 SCUD.....	72
3.5 CLOCK.....	77
3.6 SPLL	78
3.6.1 SPLL Clock Output	78
3.6.2 SPLL Spread Spectrum Mode	78
3.6.3 SPLL Modification Procedure	79
3.7 ARM Subsystem	80
3.8 CEVA Subsystem	81
3.9 SYS Subsystem	82
3.10 Clock Configuration.....	84
4 DIGITAL SIGNAL PROCESSOR (DSP)	85
4.1 Introduction	85
4.2 Features	85
5 VECTOR INTERRUPT CONTROLER (VIC).....	86
5.1 Introduction	86
5.2 Features.....	86
5.3 Block Diagram.....	86
5.4 Interrupt Source of VIC0	86
5.5 Interrupt Source of VIC1	87
5.6 Register Summary	88
5.7 Register Definition.....	90
6 NAND FLASH & BCH CONTROLLER.....	100

6.1	Introduction	100
6.2	Features	100
6.3	Block Diagram.....	101
6.4	Descriptor Defines.....	101
6.5	Register Summary	103
6.6	Register Definition.....	104
7	WATCH-DOG TIMER (WDT)	113
7.1	Introduction	113
7.2	Features	113
7.3	Register Summary	113
7.4	Register Definition.....	114
8	APBDMA0	117
8.1	Introduction	117
8.2	Features	117
8.3	Operation Mode	117
8.4	DMA End.....	118
8.5	Peripheral.....	118
8.6	Register Summary	118
8.7	Register Definition.....	119
9	APBDMA1	124
9.1	Introduction	124
9.2	Features	124
9.3	Operation Mode	124
9.4	DMA End.....	125
9.5	Control Register	125
10	SDMA	131
10.1	Introduction	131
10.2	Features	131
10.3	Block Diagram.....	131
10.4	Transfer Operation Control.....	132
10.4.1	<i>Channel Enable</i>	132
10.4.2	<i>Pause Operation</i>	132
10.4.3	<i>Resume Operation</i>	132
10.4.4	<i>Stop Operation</i>	132
10.4.5	<i>Endian Switch</i>	132
10.4.6	<i>Data Mask</i>	133
10.4.7	<i>Addressing Mode</i>	134
10.4.8	<i>Column Copy</i>	135
10.4.9	<i>TCP/IP Checksum Offload (TOE)</i>	136
10.4.10	<i>Linked List</i>	137
10.5	Register Summary	140

10.6	Register Definition.....	142
11	XDMA	156
11.1	Introduction	156
11.2	Features.....	156
12	PPU INTRODUCTION	157
12.1	General Description	157
12.2	Significant Features	157
12.3	Block Diagram.....	160
12.4	PPU Registers.....	160
12.5	TV Registers	163
12.6	Sensor Registers.....	164
12.7	PPU Special Effect RAM	165
12.7.1	<i>Horizontal Movement Control RAM</i>	165
12.7.2	<i>Horizontal Compression Control RAM</i>	165
12.7.3	<i>TEXT3 Virtual 3D Control Registers</i>	165
12.8	PPU Palette RAM.....	166
12.9	PPU Sprite Attribute RAM	167
12.10	PPU Color Mapping RAM	168
13	PPU CONTROL.....	169
13.1	Screen Structure	169
13.2	Character	171
13.2.1	<i>Color Mode</i>	174
13.2.2	<i>Palette Control</i>	180
13.2.3	<i>Depth</i>	185
13.2.4	<i>Character Size</i>	186
13.2.5	<i>Character Flip</i>	188
13.2.6	<i>Blending Effect</i>	190
13.2.7	<i>Rotate Effect</i>	192
13.2.8	<i>Zoom Effect</i>	193
13.2.9	<i>Mosaic Effect</i>	195
13.2.10	<i>Fade Effect</i>	196
13.2.11	<i>Character Number</i>	197
13.3	TEXT Control	205
13.3.1	<i>TEXT Register Summary</i>	205
13.3.2	<i>TEXT SRAM Summary</i>	208
13.3.3	<i>TEXT Layer Structure</i>	210
13.3.4	<i>TEXT Mode Control</i>	214
13.3.5	<i>TEXT Horizontal Movement</i>	230
13.3.6	<i>TEXT Calculation Order</i>	232
13.3.7	<i>TEXT Character 0 Auto Transparent Mode</i>	233
13.4	Sprite Control	234
13.4.1	<i>Sprite Register Summary</i>	234

13.4.2	Sprite SRAM Summary.....	235
13.4.3	Sprite General Control.....	239
13.4.4	Sprite Coordinate.....	241
13.4.5	Sprite DMA	246
13.4.6	Sprite Virtual 3D Mode.....	248
13.4.7	Sprite Color Dither Mode	253
13.4.8	Extended Sprite General Control.....	256
13.4.9	Sprite Group Function.....	261
13.5	PPU Interrupt Control.....	263
13.6	Light Pen Interface.....	269
13.7	TV Line Counter	270
13.8	Frame Base Mode.....	271
13.9	Window Mode	283
13.10	Random Number Generator.....	287
13.11	GTE Controller	288
13.12	ROM Saving Mode.....	303
13.13	Color Mapping.....	304
13.14	PPU Special Effect.....	305
13.15	Horizontal Blank Interrupt.....	309
13.16	PPU Bi-linear Interpolation Mode	310
13.17	TV up-scaling Function.	311
13.18	De-flicker Control	312
13.19	PPU Color Mask function.....	313
14	OVG INTRODUCTION.....	315
14.1	Genearal Description	315
15	DISPLAY CONTROLLER.....	316
15.1	Introduction	316
15.2	Features	316
15.3	Block Diagram.....	316
15.4	Register Summary	318
15.5	Register Definition.....	318
15.6	APPENDIX.....	342
16	ON SCREEN DISPLAY (OSD) CONTROLLER.....	346
16.1	Introduction	346
16.2	Register Summary	346
16.3	Register Definition.....	346
17	INDEPENDENT SCALING CONTROLLER.....	355
17.1	Introduction	355
17.2	Register Summary	355
17.3	Register Definition.....	355

18 BUILT-IN SCALING CONTROLLER	368
18.1 Introduction	368
18.2 Register Summary	368
18.3 Register Definition.....	368
19 SCALER2	372
19.1 Introduction	372
19.2 Features	372
19.3 Block Diagram.....	373
19.4 Register Summary	373
19.5 Register Definition.....	375
19.5.1 <i>P_SCALER2_CTRL</i>	375
19.5.2 <i>P_SCALER2_OB_COLOR</i>	384
19.5.3 <i>P_SCALER2_OUTX_WIDTH</i>	384
19.5.4 <i>P_SCALER2_OUTY_WIDTH</i>	385
19.5.5 <i>P_SCALER2_X_FACTOR</i>	386
19.5.6 <i>P_SCALER2_Y_FACTOR</i>	386
19.5.7 <i>P_SCALER2_X_START</i>	387
19.5.8 <i>P_SCALER2_Y_START</i>	387
19.5.9 <i>P_SCALER2_IN_X_Width</i>	388
19.5.10 <i>P_SCALER2_IN_Y_Width</i>	389
19.5.11 <i>P_SCALER2_Y_INPTR</i>	389
19.5.12 <i>P_SCALER2_U_INPTR</i>	390
19.5.13 <i>P_SCALER2_V_INPTR</i>	390
19.5.14 <i>P_SCALER2_Y_OUTPTR</i>	391
19.5.15 <i>P_SCALER2_U_OUTPTR</i>	392
19.5.16 <i>P_SCALER2_V_OUTPTR</i>	392
19.5.17 <i>P_SCALER2_CUR_LINE</i>	394
19.5.18 <i>P_SCALER2_A11</i>	394
19.5.19 <i>P_SCALER2_A12</i>	395
19.5.20 <i>P_SCALER2_A13</i>	395
19.5.21 <i>P_SCALER2_A21</i>	396
19.5.22 <i>P_SCALER2_A22</i>	396
19.5.23 <i>P_SCALER2_A23</i>	397
19.5.24 <i>P_SCALER2_A31</i>	397
19.5.25 <i>P_SCALER2_A32</i>	398
19.5.26 <i>P_SCALER2_A33</i>	398
19.5.27 <i>P_SCALER2_IN_RX_WIDTH</i>	399
19.5.28 <i>P_SCALER2_IN_RY_WIDTH</i>	399
19.5.29 <i>P_SCALER2_OUT_OFFSET</i>	400
19.5.30 <i>P_SCALER2_LB_PTR</i>	400
19.5.31 <i>P_SCALER2_INT</i>	401
19.5.32 <i>P_SCALER2_POST</i>	402
19.5.33 <i>P_SCALER2_MAX_Y</i>	403

19.5.34 <i>P_SCALER2_MIN_Y</i>	403
19.5.35 <i>P_SCALER2_SUM_Y</i>	404
19.5.36 <i>P_SCALER2_MAX_U</i>	404
19.5.37 <i>P_SCALER2_MIN_U</i>	405
19.5.38 <i>P_SCALER2_SUM_U</i>	405
19.5.39 <i>P_SCALER2_MAX_V</i>	406
19.5.40 <i>P_SCALER2_MIN_V</i>	406
19.5.41 <i>P_SCALER2_SUM_V</i>	407
19.5.42 <i>P_SCALER2_Y_HIS0</i>	407
19.5.43 <i>P_SCALER2_Y_HIS1</i>	408
19.5.44 <i>P_SCALER2_Y_HIS2</i>	409
19.5.45 <i>P_SCALER2_Y_HIS3</i>	409
19.5.46 <i>P_SCALER2_Y_HIS4</i>	410
19.5.47 <i>P_SCALER2_Y_HIS5</i>	411
19.5.48 <i>P_SCALER2_Y_HIS6</i>	411
19.5.49 <i>P_SCALER2_Y_HIS7</i>	412
19.5.50 <i>P_SCALER2_Y_HIS8</i>	412
19.5.51 <i>P_SCALER2_Y_HIS9</i>	413
19.5.52 <i>P_SCALER2_Y_HIS10</i>	414
19.5.53 <i>P_SCALER2_Y_HIS11</i>	414
19.5.54 <i>P_SCALER2_Y_HIS12</i>	415
19.5.55 <i>P_SCALER2_Y_HIS13</i>	415
19.5.56 <i>P_SCALER2_Y_HIS14</i>	416
19.5.57 <i>P_SCALER2_Y_HIS15</i>	417
19.5.58 <i>P_SCALER2_CONTI_W1</i>	417
19.5.59 <i>P_SCALER2_CONTI_R1</i>	418
19.5.60 <i>P_SCALER2_CONTI_R2</i>	419
19.5.61 <i>P_SCALER2_Y_GAMMA_0~P_SCALER2_Y_GAMMA_255</i>	419
19.5.62 <i>P_SCALER2_Y_HIS0~P_SCALER2_Y_HIS255</i>	420
20 CMOS SENSOR INTERFACE (CSI0).....	421
20.1 Introduction	421
20.2 Features	421
20.3 Block Diagram.....	421
20.4 Sensor Control Pin Configuration.....	421
20.5 Register Summary	422
20.6 Register Definition.....	422
21 CMOS SENSOR INTERFACE 1	429
21.1 Features	429
21.2 Block Diagram.....	429
21.3 CSI1 Register.....	430
21.4 Register Definition.....	430
21.4.1 <i>Sensor General Control Register</i>	430

21.4.2	<i>CSI1 CONTROL REGISTER0 Register</i>	430
21.4.3	<i>CSI1 CONTROL REGISTER1 Register</i>	432
21.4.4	<i>CSI1 HORIZONTAL SETTING Register</i>	433
21.4.5	<i>CSI1 VERTICAL SETTING Register</i>	434
21.4.6	<i>CSI1 PER LINE OFFSET IN FRAME BUFFER Register</i>	434
21.4.7	<i>CSI1 FRAME BUFFER ADDRESS0 Register</i>	435
21.4.8	<i>CSI1 FRAME BUFFER ADDRESS1 Register</i>	436
21.4.9	<i>CSI1 FRAME BUFFER ADDRESS2 Register</i>	436
21.4.10	<i>CSI1 HOLD NEXT FRAME CONTROL BIT Register</i>	437
21.4.11	<i>CSI1 IRQ ENABLE Register</i>	437
21.4.12	<i>CSI1 IRQ STATUS Register</i>	438
21.5	FUNCTION / DESIGN Description	439
22	COLOR DSP (CDSP)	440
22.1	Introduction	440
22.2	Features	440
23	MOBILE INDUSTRY PROCESSOR INTERFACE (MIPI)	441
23.1	Introduction	441
23.2	Features	441
23.3	Block Diagram	441
23.4	MIPI Control Pin Configuration	442
23.5	Register Summary	442
23.6	Register Definition	442
24	TFT1 LCD	453
24.1	Introduction	453
24.2	Features	453
24.3	Interface Signals	454
24.4	Register Summary	454
24.5	Register Definition	455
24.6	Programming Guide	473
24.7	TFT1 Timing Diagram	476
24.8	TCON Timing Diagram	478
25	STN	479
25.1	Introduction	479
25.2	Features	479
25.3	STN Interface Configuration	479
25.4	Register Summary	480
25.5	Register Definition	480
25.5.1	<i>Control Registers</i>	480
25.5.2	<i>STN Panel without the Clipping Function</i>	485
25.5.3	<i>STN Panel with Clipping Function Enabled</i>	485
25.5.4	<i>STN Timing</i>	486

26 DC-DC.....	488
26.1 Introduction	488
26.2 Features	488
26.3 Register Summary	488
26.4 Register Definition.....	488
27 TV0 ENCODER INTERFACE	490
27.1 Introduction	490
27.2 Features	490
27.3 Block Diagram.....	491
27.4 TVE0 Registers	492
27.5 Register Definition.....	493
27.5.1 Output TV0 Resolution Register	493
27.5.2 TV0 Blanking (Vertical) Register.....	494
27.5.3 TV0 Blanking (Horizontal) Register.....	495
27.5.4 TV0 Blanking Pattern Register.....	496
27.5.5 TV0 Frame Buffer Address Register.....	497
27.5.6 TV0 Frame Buffer Pitch Register.....	497
27.5.7 TV0 Frame Buffer Resolution Register	498
27.5.8 TV0 Scaling Resolution Register	499
27.5.9 TV0 Scaling Parameter (Horizontal) Register.....	499
27.5.10 TV0 Scaling Parameter (Vertical) Register	500
27.5.11 TV0 Scaling Control Register.....	501
27.5.12 TV0 OSDx Base Address Register	502
27.5.13 TV0 OSDx Pitch Register Register	503
27.5.14 TV0 OSDx Resolution Register	503
27.5.15 TV0 OSDx Starting Coordinate Register.....	504
27.5.16 TV0 OSDx Format Register	505
27.5.17 TV0 OSDx Control Register.....	506
27.5.18 TV0 OSDx Scaling Parameter (Horizontal) Register	507
27.5.19 TV0 OSDx Scaling Parameter (Vertical) Register.....	508
27.5.20 TV0 OSDx Scaling Parameter (field 1) Register.....	508
27.5.21 TV0 OSDx Scaling Resolution Register.....	509
27.5.22 TV0 Interrupt Enable Register	510
27.5.23 TV0 Interrupt Status Register	511
27.5.24 TV0 Colorbar Control Register.....	512
27.5.25 TV0 Colorbar Color Register.....	513
27.5.26 TV0 Color Matrix Param0 Register	514
27.5.27 TV0 Color Matrix Param1 Register	514
27.5.28 TV0 Color Matrix Param2 Register	515
27.5.29 TV0 Color Matrix Param3 Register	515
27.5.30 TV0 Color Matrix Param4 Register	516
27.5.31 TV0 Color Matrix Param5 Register	517
27.5.32 TV0 Amplitude Adjust Register	517

27.5.33 <i>TV0 Position Adjust Register</i>	518
27.5.34 <i>TV0 LCM AC Timing Register</i>	519
27.5.35 <i>TV0 Display Format Register</i>	521
27.5.36 <i>TV0 Display Control Register</i>	522
27.5.37 <i>TV0 OSDx Palette Pattern Register File</i>	524
27.6 Recommended Settings	525
28 TV ENCODER INTERFACE 1	527
28.1 Introduction	527
28.2 Features	527
28.3 Block Diagram	527
28.4 Register Summary	528
28.5 Register Definition	529
28.5.1 <i>Control Register</i>	529
28.5.2 <i>Saturation Register</i>	531
28.5.3 <i>Hue Register</i>	531
28.5.4 <i>Brightness Register</i>	531
28.5.5 <i>Sharpness Register</i>	532
28.5.6 <i>Y Gain Register</i>	533
28.5.7 <i>Y Delay Register</i>	533
28.5.8 <i>Vertical Position Register</i>	534
28.5.9 <i>Horizontal Position Register</i>	534
28.5.10 <i>Video DAC Control Register</i>	534
28.6 Recommended Settings	535
29 SPU INTRODUCTION	538
29.1 General Description	538
29.2 Feature	538
29.3 Internal Memory Mapping	539
29.4 Gain Control	540
29.5 Fundamentals	540
29.6 Musical Sound	541
29.7 Control Register	542
29.8 Control Register List	545
29.9 Internal SRAM List	549
29.10 SPU Control Register	551
29.10.1 <i>SPU Control Flag</i>	551
29.10.2 <i>Channel Enable Control</i>	553
29.10.3 <i>Main Volume Control</i>	555
29.10.4 <i>Beat Control</i>	556
29.10.5 <i>Tone Control</i>	557
29.10.6 <i>SoftChannel Control</i>	560
29.10.7 <i>PitchBend Enable Control</i>	562
29.10.8 <i>Envelope Control</i>	563

29.10.9 Compressor Control.....	573
29.10.10 Post Process Control	577
29.10.11 Frequency Vibration Control	580
29.11 Internal SRAM.....	585
29.11.1 Tone-Color Register.....	587
29.11.2 Volume Register	593
29.11.3 Envelope Register	595
29.11.4 Phase Register	603
29.11.5 PitchBend Register	605
29.11.6 DAM Register	607
29.11.7 LFO Register	609
29.12 Programming Tips	610
29.13 Programming Flow Example	611
30 SAR ADC (ADC).....	612
30.1 Introduction	612
30.2 Features	612
30.3 Block Diagram.....	612
30.4 SAR ADC Measurement Flow	613
30.5 Register Summary	615
30.6 Register Definition.....	615
31 I2S.....	622
31.1 Introduction	622
31.2 Features	622
31.3 Block Diagram.....	623
31.4 I2S Tx.....	624
31.5 Control Register	624
32 AUDIO ADC	630
32.1 Introduction	630
32.2 Register Summary	630
32.3 Register Definition.....	630
33 USB INTERFACE	634
33.1 Introduction	634
33.2 Features	634
34 SD/MMC CARD CONTROLLER	635
34.1 Introduction	635
34.2 Features	635
34.3 Block Diagram.....	636
34.4 Control Architecture.....	636
34.4.1 Command Line Control.....	636
34.4.2 Data Line Control.....	637

34.4.3	<i>Card Busy Detection</i>	637
34.4.4	<i>Card Insertion Detection</i>	637
34.4.5	<i>Multi-Block Read/Write</i>	637
34.4.6	<i>Transmit / Receive Data via APBDMA Channel</i>	638
34.5	Control Pin Configuration	639
34.5.1	<i>SD 0 Pin Configuration</i>	639
34.5.2	<i>SD 1 Pin Configuration</i>	639
34.6	Register Summary	639
34.7	Register Definition.....	640
35	MEMORY STICK	649
35.1	Introduction	649
35.2	Features	649
36	UART	650
36.1	Introduction	650
36.2	Features	650
36.3	Block Diagram.....	650
36.4	Transmit / Receive Data via APBDMA Channel	651
36.5	Control Pin Configuration	652
36.5.1	<i>UART 0 Pin Configuration</i>	652
36.5.2	<i>UART 1 Pin Configuration</i>	652
36.6	Register Summary	652
36.7	Register Definition.....	653
37	SERIAL PERIPHERAL INTERFACE (SPI).....	661
37.1	Introduction	661
37.2	Features	661
37.3	Block Diagram.....	662
37.4	Register Summary	662
37.5	Register Definition.....	663
38	SERIAL PERIPHERAL INTERFACE (I2C).....	674
38.1	Introduction	674
38.2	Features	675
38.3	Block Diagram.....	675
38.4	I2C Control Pin Configuration	675
38.5	I2C Transfer and Receiver Mode	676
38.6	Register Summary	678
38.7	Register Definition.....	678
39	TIMER.....	682
39.1	Introduction	682
39.2	Features	682
39.3	Register Summary	682

39.4	Register Definition.....	683
39.5	Operation Mode	686
40	RTC.....	687
40.1	Introduction	687
40.2	Features.....	688
40.3	Register Summary	688
40.4	Register Definition.....	689
40.5	RTC Macro Internal Register.....	690
41	I/O PORTS.....	693
41.1	Introduction	693
41.2	Register Summary	702
41.3	Register Definition.....	704
42	AES.....	732
42.1	Features	732
43	RANDOM NUMBER GENERATOR.....	733
43.1	Introduction	733
43.2	Register Summary	733
43.3	Register Definition.....	733
44	POWER CONTROL (PWRC).....	736
44.1	Introduction	736
44.2	Features	736
44.3	Block Diagram.....	737
44.4	Power Control Pin Configuration.....	738
44.5	Power Control Mode	739



Revision History

Revision	Date	By	Remark
0.1	2010/12/21	Bruce Huang	Original



1 Confirmation Sheet

The confirmation sheet, as a requisite document before placing orders, contains useful information and checklist that help to avoid mistakes during development. Due to the updates of the confirmation sheet, please download a newest confirmation sheet from <http://www.generalplus.com/>.

2 Introduction

2.1 General Description

GPL32900A is a highly integrated SoC (System-On a Chip) that offers a great cost-effective and high performance ratio solution for Multimedia Applications. It is embedded the ARM1176JZFS with 32K-byte I-cache, 32K-byte D-cache and many tremendous features such as high performance DSP, 2D graphic accelerator, picture process unit (PPU), TFT-LCD/STN interface, TV encoder, CMOS/MIPI sensor interface, Color DSP engine, scalar engine, 32-channel sound process unit (SPU), SDR/DDR/mDDR/DDRII controller, NAND flash controller, BCH, AES-128, DMA controller, programmable I/O ports, timer, RTC, USB 2.0 device, USB 2.0 UHCI/EHCI host controller, UART/IrDA interface, SPI, I2C, I2S, SD/MMC card interface, MS/MS pro card interface, Power control macro, Stereo Audio DAC, Stereo line-in, Microphone line-in, 5-channel 10-bit ADC, PLL, DC-DC Boost control and 16K-byte embedded SRAM.

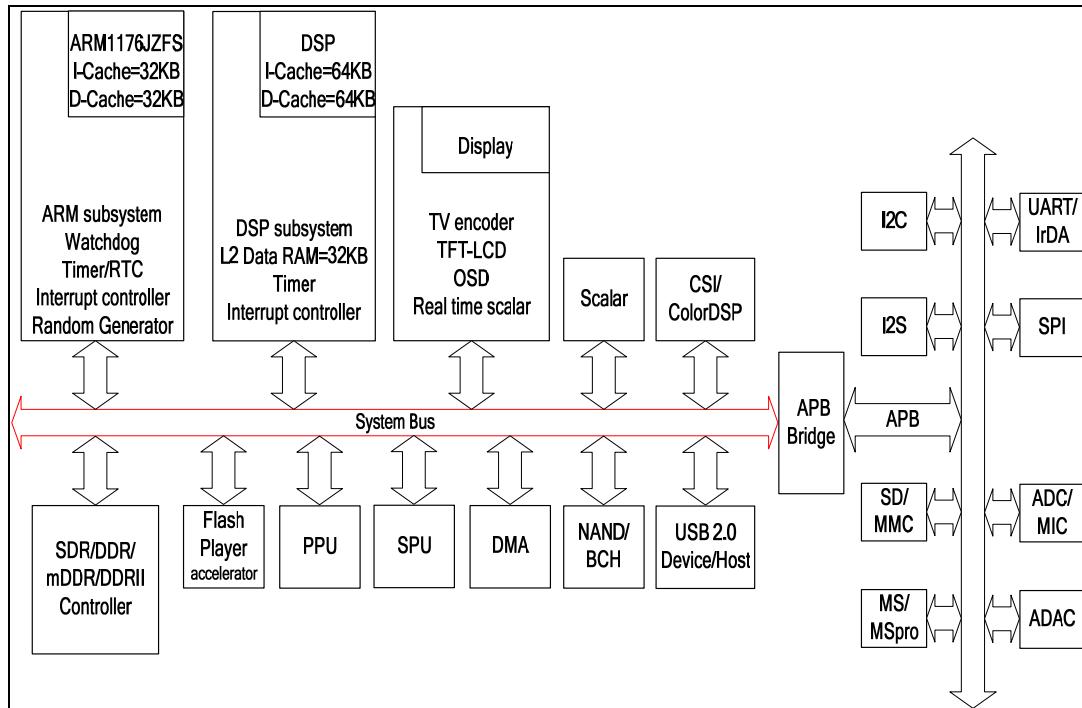
With a complete set of common system peripherals, GPL32900A chip minimizes overall system costs and no additional component needs to be added. GPL32900A features not only the high-speed performance, but it is also a cost-effective system and the most important - compatible with all ARM based programs.

2.2 Significant Features

- High performance CPU + DSP dual-core architecture
- ARM1176JZFS CPU with 32KB I-cache and 32KB D-cache, Vector Floating Point (VFP) Coprocessor, embedded JTAG ICE, and working frequency up to 400MHz
- High performance DSP with L1 64KB I-cache, L1 64KB D-cache and L2 32KB data ram, and working frequency up to 324MHz
- 16KB SRAM for local data buffer
- 2D graphic hardware accelerator, compatible with OpenVG 1.1 standard
- Picture Process Unit (PPU)
- Virtual 3D effect for text (or background slide show) and sprite (or OSD)
- Up to 4 Text layers + 5K Sprites
- Max. 1920x1080 LCD Resolution output
- Video-in & CMOS sensor interface, MIPI sensor interface and CCIR601/CCIR656 standard support
- Color DSP inside, support CMOS sensor RAW data interface with up to 5 million pixels
- Edge-enhance and de-noise function for Image post Processing
- High quality Scalar engine
- Sound Process Unit
- 32 hardware PCM/ADPCM channels(SPU)
- Built-in dynamic volume compressor
- Variety of audio codec
- 324 MHz DDR/mDDR/DDRII controller, 162 MHz SDR controller support
- NAND flash controller with BCH up to 60-bit error correction

- AES-128 encryption and decryption
- 4 sets of DMA controller for audio, peripheral, DSP and public system application
- Display engine
- TFT-LCD controller
- UPS051 / UPS052 (serial RGB)
- Parallel RGB (5-6-5, 6-6-6, 8-8-8)
- I80 (8-bit/16-bit system bus) I/F type
- Tearing effect control
- CCIR601/CCIR656
- Digital Timing Controller for TFT-LCD drivers
- Mono and 16 gray levels STN-LCD controller
- TV encoder
- NTSC/PAL output
- YPbPr output
- De-flicker function
- Real time OSD, which also supports OSD scaling up or scaling down
- Real time (on screen) scalar engine inside to fit the display screen
- Programmable RGB gamma correction
- Color convert matrix for display
- Supports TV and TFT-LCD enable at the same time
- Watchdog timer
- 5 16-bit timers/counters
- Real-time clock
- Programmable general I/O ports (GPIO) with pull-high/low control
- Universal Serial Bus (USB) 2.0 high/full speed compliance device and USB 2.0 UHCI/EHCI host controller
- UART/IrDA interface with baud rate up to 1.8432Mbps and 115.2Kbps
- 2 sets of SPI (master) interface
- I2C interface
- I2S master
- 2 sets of SD/SDHC/SDIO/MMC card interface, which also support eMMC interface
- MS/MS pro card interface
- Power manager control
- Low voltage reset
- 48KSPS 24-bit Stereo Sigma/Delta DAC for audio playback
- 48KSPS 16-bit Stereo line-in and Microphone line-in for audio recorder
- 125KSPS 10-bit SAR ADC with 5 line-in channels (also including Touch panel application), a core VDD channel and a system power source channel
- PLL
- 2-channel DC-DC Boost control circuit for LED Backlight and VGH/VGL voltage generator of TFT-LCD drivers

2.3 Block Diagram



2.4 Applications

- TV game
- Advanced educational toy
- Kid storybook, E-book
- Hand-held, multimedia LCD game
- Educational Learning Assistant
- Multi-Media Dictionary
- Portable Media Player
- Tablet PC DPF
- Netbook
- PDA

3 System Control

3.1 Introduction

This chapter describes four “System Control Units (SCU)” in GPL32900A, which are SCU_A, SCU_B, SCU_C, and SCU_D.

3.2 Features

- Two built-in SPLLL circuits.
- The clock of each module can be turned on/off individually for saving power.

3.3 Register Summary

SCUA

Name	Address	Description
P_SCUA_PERI_RST	0x93007000	SCUA Peripheral Reset
P_SCUA_PERI_CLKEN	0x93007004	SCUA Peripheral Clock Enable
P_SCUA_PERI_DGCLKEN	0x9300700C	SCUA Dynamic Clock Gating Enable
P_SCUA_DISP_TYPE	0x93007010	TFT0 Display Type Select
P_SCUA_PERI_CLKEN2	0x93007018	SCUA Peripheral Clock Enable 2
P_SCUA_USBPHY_CFG	0x9300703C	USB Phy Configure
P_SCUA_VDAC_CFG	0x93007040	Video DAC Configure
P_SCUA_APOLL_CFG	0x93007044	Audio PLL Configure
P_SCUA_LCD_CLK_CFG	0x93007080	TFT0 Clock Configure
P_SCUA_CSI_CLK_CFG	0x93007084	CSI Clock Configure
P_SCUA_I2S_BCK_CFG	0x93007090	I2S BCK Configure
P_SCUA_UART_CFG	0x93007094	UART Clock Configure
P_SCUA_SYS_SEL	0x930070E0	System Select
P_SCUA_CSI2_CLK_CFG	0x930070E8	CSI Clock Configure 2
P_SCUA_CDSP_PCLK	0x930070EC	CDSP Clock Configure
P_SCUA_MIPI_2CH_PCLK	0x930070F0	MIPI 2 channel Clock Configure

SCUB

Name	Address	Description
P_SCUB_PERI_RST	0x90005000	SCUB Peripheral Reset
P_SCUB_SPLL_CFG0	0x90005004	SPLL Configure 0
P_SCUB_INTR_STATUS	0x90005008	Timer Interrupt Status
P_SCUB_TIMER_ICE_EN	0x9000500C	Timer Stop Enable When ICE Connect

Name	Address	Description
P_SCUB_TIMER_EXT_CTRL	0x90005010	Timer External Trigger Up/Down
P_SCUB_REV	0x90005018	IC Version
P_SCUB_RAND0	0x9000501C	Random Seed
P_SCUB_PERI_CLKEN	0x90005020	SCUB Clock Enable
P_SCUB_PERI_DGCLKEN	0x90005024	SCUB Dynamic Gated Clock Enable
P_SCUB_UPDATE_RATIO	0x90005028	ARM Clock Ratio Update Control
P_SCUB_PWRC_CFG	0x90005040	PWRC Configure
P_SCUB_RAND1	0x90005048	Random Seed
P_SCUB_DBGRQ_CTRL	0x90005058	ARM Dynamic Gated Clock Control
P_SCUB_SPLL_CFG1	0x9000505C	SPLL Configure 1
P_SCUB_SPLL_CFG2	0x90005060	SPLL Configure 2
P_SCUB_PGS0	0x90005080	PAD Group Selection 0
P_SCUB_PGS1	0x90005084	PAD Group Selection 1
P_SCUB_PGS2	0x90005088	PAD Group Selection 2
P_SCUB_PGS3	0x9000508C	PAD Group Selection 3
P_SCUB_ARM_RATIO	0x900050D0	ARM Clock Ratio
P_SCUB_ARM_AHB_RATIO	0x900050D4	ARM AHB Clock Ratio
P_SCUB_ARM_APB_RATIO	0x900050D8	ARM APB Clock Ratio
P_SCUB_SYS_CNT_EN	0x900050DC	System Counter Enable
P_SCUB_PIN_MUX	0x90005144	IO Pad Mux Configure
P_SCUB_CP15SDISABLE	0x90005150	ARM CP15 Access Disable
P_SCUB_ARM_JTAG	0x90005154	AMR JTAG Information
P_SCUB_TZ_CFG	0x90005158	TrusZone Secure and Non-Secure Region Configure

SCUC

Name	Address	Description
P_SCUC_PERI_RST	0x92005000	SCUC Peripheral Reset
P_SCUC_PERI_CLKEN	0x92005004	SCUC Clock Enable
P_SCUC_PERI_DGCLKEN	0x92005008	SCUC Dynamic Gated Clock Control
P_SCUC_GC_CFG0	0x92005010	DDRPHY Configure 0
P_SCUC_SYS_RATIO_UPDATE	0x92005028	System Clock Ration Updata Control
P_SCUC_DDRPHY_CTRL2	0x92005040	DDRPHY Control 2
P_SCUC_DDRPHY_STATUS	0x92005044	DDRPHY Status
P_SCUC_DDRPHY_CTRL0	0x92005048	DDRPHY Control 0
P_SCUC_DDRPHY_CTRL1	0x9200504C	DDRPHY Control 1
P_SCUC_SYS_RATIO	0x92005100	System Clock Ratio

Name	Address	Description
P_SCUC_SYS_RT_RATIO	0x92005104	System RT Clock Ratio
P_SCUC_SYS_AHB_RATIO	0x92005108	System AHB Clock Ratio
P_SCUC_SYS_APB_RATIO	0x9200510C	System APB Clock Ratio
P_SCUC_CEVA_RATIO	0x92005110	CEVA Clock Ration
P_SCUC_CEVA_AHB_RATIO	0x92005114	CEVA AHB Clock Ration
P_SCUC_CEVA_APB_RATIO	0x92005118	CEVA APB Clock Ration
P_SCUC_CEVA_CNT_EN	0x9200511C	CEVA Counter Enable

SCUD

Name	Address	Description
P_SCUD_DRAM_MAP	0x90532800	DRAM Map Base
P_SCUD_SB0_RGN	0x90532804	Spare Space Control 0
P_SCUD_SB1_RGN	0x90532808	Spare Space Control 1
P_SCUD_PMIM_ST	0x9053280C	PMIM Status
P_SCUD_DMIM_ST	0x90532810	DMIM Status

3.4 Register Definition

3.4.1 SCUA

Peripheral reset, please reference to table 1 for the peripheral device of each source. Please set 1 to reset each device.

P_SCUA_PERI_RST 0x93007000 SCUA Peripheral Reset							
Bit	31	30	29	28	27	26	25
Function	<i>Peripheral[31:0]</i>						
Default	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
<i>Peripheral[31:0]</i>							

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
<i>Peripheral[31:0]</i>							

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
<i>Peripheral[31:0]</i>							

0 0 0 0 0 0 0 0

Peripheral clock enable, please reference to table 1 for the peripheral device of each source. Please set 1 to enable each device clock. If set 0, the clock of that peripheral module is gated.

P_SCUA_PERI_CLKEN 0x93007004 SCUA Peripheral Clock Enable								
Bit	31	30	29	28	27	26	25	
Function	Peripheral[31:0]							
Default	0	0	0	0	0	0	0	
Peripheral[31:0]								
23	22	21	20	19	18	17	16	
1	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	

Peripheral dynamic clock gating enable, please reference to table 1 for the peripheral device of each source. The dynamic gating can save the power consumption when the module is working. Please set 1 to enable dynamic clock gating and set 0 to disable dynamic clock gating.

P_SCUA_PERI_DGCLKEN 0x9300700C SCUA Dynamic Clock Gating Enable								
Bit	31	30	29	28	27	26	25	
Function	Peripheral[31:0]							
Default	1	1	1	1	1	1	1	
Peripheral[31:0]								
23	22	21	20	19	18	17	16	
1	1	1	1	1	1	1	1	
15	14	13	12	11	10	9	8	
1	1	1	1	1	1	1	1	
7	6	5	4	3	2	1	0	
1	1	1	1	1	1	1	1	

					R : Reset	G : Clock Gating			D : Dynamic Clock Gating			B : Debug ID				
System A	R	G	D	B	System B	R	G	D	B	System C	R	G	D	B		
	[0]				TCM_BIST	[0]	[0]			Fabric_C	[0]					
LCD CTRL	[1]	[1]	[1]	1	TCM CTRL	[1]	[1]	[1]		DMAC0	[1]	[1]	[1]	40,41		
					AHB2AHB	[2]	[2]	[2]	22	DMAC1	[2]	[2]	[2]			
USB0	[3]	[3]	[3]	3	AHB SW	[3]					[3]					
USB1	[4]	[4]	[4]	4	VIC0	[4]	[4]	[4]	24	DRAM CTRL	[4]	[4]	[4]	44		
LINEBUFFER	[5]	[5]			VIC1	[5]	[5]	[5]	25	SCU_C	[5]					
SCU_A	[6]				DPM (xxx)	[6]				I2C_CFG	[6]			[6]		
					APB Bridge	[7]				APBDMA_C	[7]	[7]	[7]	47		
					ARM1176JZF-S	[8]										
APBDMA_A	[9]	[9]	[9]	9	Timer0	[9]	[9]	[9]	29							
CMOS CTRL	[10]	[10]	[10]	A	Timer1	[10]	[10]	[10]	2A							
NAND0	[11]	[11]	[11]	B	UART	[11]				MS	[11]	[11]	[11]	4B		
					ARM_I2C	[12]	[12]	[12]	2C	INT_MEM	[12]	[12]	[12]	4C		
BCH	[13]	[13]	[13]	D	RAND	[13]	[13]	[13]	2D	UART_C0	[13]	[13]	[13]	4D		
APLL	[14]				GPIO	[14]	[14]	[14]								
UART_CON	[15]				RTC	[15]	[15]	[15]		UART_C2	[15]	[15]	[15]	4F		
AAHBM212	[16]	[16]	[16]	10						SSP0	[16]	[16]	[16]	50		
I2S	[17]	[17]	[17]	11	System D	R	G	D	B	SSP1	[17]	[17]	[17]	51		
I2SRX	[18]	[18]	[18]	12	CXS_SL	[0]			60	SD0	[18]	[18]	[18]	52		
SAACC	[19]	[19]	[19]	13						SD1	[19]	[19]	[19]	53		
nand_abt	[20]	[20]	[20]	14						SYS_I2C	[20]	[20]	[20]	54		
realtime_abt	[21]	[21]	[21]	15						Scaling	[21]	[21]	[21]	55		
rtabt212	[22]	[22]	[22]	16						2dscaleabt	[22]	[22]	[22]	56		
cahbm212	[23]		[23]	17						TI2C	[23]	[23]	[23]			
										SYS_A	[24]	[24]				
SPU	[25]	[25]	[25]							CXMP212		[25]	[25]	59		
SCA	[26]	[26]	[26]							CXMD212		[26]	[26]	5A		
OVG	[27]	[27]	[27]							CIR	[27]	[27]	[27]	5B		
MIPI	[28]	[28]	[28]													
CDSP	[29]	[29]	[29]							EFUSE	[29]	[29]	[29]			
AES	[30]	[30]	[30]													
System A New	R	G	D	B												
PPU_SPR	[0]	[0]	[0]													



Table 1 the control bit of the module in SCU

P_SCUA_DISP_TYPE 0x93007010
TFT0 Display Type Select

Bit	31	30	29	28	27	26	25	24
Function					-			
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
				-				DISP_TYPE
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
			-					BUSY_STATUS
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
								LINE_BUFFER_MODE
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:18]	Reserved	-						
[17:16]	DISP_TYPE	R/W	TFT0 display type select: [17:16] = 2'b00 : RGB/RGB565 [17:16] = 2'b01 : RGB666 [17:16] = 2'b10 : LCM [17:16] = 2'b11 : YpbPr					
[15:12]	Reserved	-						
[11:8]	BUSY_STATUS	R/W	Busy status [DSGR]					
[7:0]	LINE_BUFFER_MODE	R/W	TFT0 Line buffer mode select					
			0	D	D	S	S	
			1	D	D	S	G	
			2	D	D	G	G	
			3	D	D	V	V	
			4	D	D	S	R	
			5	D	D	G	R	
			6	D	S	S	R	
			7	D	S	G	R	
			8	D	G	G	R	
			9	D	V	V	R	
			A	D	V	V	G	
			B	S	S	G	G	
			C	S	S	G	R	
			D	S	G	G	R	
			E	V	V	G	G	
			F	V	V	G	R	
			(D)isplay (S)caler/(V)ertical Scaler (G)raphic2D (R)otator					

Peripheral clock enable, please reference to table 1 for the peripheral device of each source. Please set 1 to enable each device clock. If set 0, the clock of that peripheral module is gated

P_SCUA_PERI_CLKEN2 0x93007018 SCUA Peripheral Clock Enable 2								
Bit	31	30	29	28	27	26	25	
Function	Peripheral[31:0]							
Default	0	0	0	0	0	0	0	
23	22	21	20	19	18	17	16	
Peripheral[31:0]								
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
Peripheral[31:0]								
0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	
Peripheral[31:0]								
0	0	0	0	0	0	0	0	

P_SCUA_USBPHY_CFG 0x9300703C USB Phy Configure								
Bit	31	30	29	28	27	26	25	
Function	I2C_DIS	I2C_ID						
Default	0	0	0	0	0	1	0	
23	22	21	20	19	18	17	16	
PTG								
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
PHY1_TEST	PHY1_NODRIVE	P1	PHY0_TEST			P0		
0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	
XTAL_EN						D	H	
0	0	0	0	0	1	0	0	

There are two USB Ports in this chip. And there are 1 “USB Host Controller” and 1 “USB Device Controller” in this chip. USBPHY_CFG selects the connection of the USB Host/Device and the PHY Port0/Port1.

Bit	Function	Type	Description					Condition															
31	I2C_DIS	R/W	USBPHY I2C disable 1: I2C disable																				
[30:24]	I2C_ID	R/W	I2C ID of the two USBPHYS USBPHY0 ID = I2CC[6:0] USBPHY1 ID = I2CC[6:0] + 1																				
[23:20]	Reserved	-	-																				
[20:16]	PTG	R/W	For USBPHY test mode																				
15	PHY1_TEST	R/W	USBPHY0_CTRL_DPDMPAD in testmode																				
14	PHY1_NODRIVE	R/W	USBPHY1 Nodrive																				
[13:12]	P1	R/W	USBP1 Power Control x0: Control By USB Port1 Owner (Host/Device Controller) 01: Force Suspend 11: Force Wakeup																				
11	PHY0_TEST	R/W	USBPHY0_CTRL_DPDMPAD in testmode																				
[9:8]	P0	R/W	USBP0 Power Control x0: Control By USB Port0 Owner (Host Controller) 01: Force Susbspend 11: Force Wakeup																				
[7:3]	Reserved	-																					
2	XTAL_EN	R/W	Usb phy xtal enable																				
1	D	R/W	Select the host/device mode of USBPORT1 When D=0 or H=0, USBPHY1 is Device. When D=1 and H=1, USBPHY1 is Host																				
			<table border="1"> <tr> <td>{D,H}</td><td>2'b00</td><td>2'b01</td><td>2'b10</td><td>2'b11</td></tr> <tr> <td>USB Port0</td><td>Host</td><td>Disable</td><td>Host</td><td>Disable</td></tr> <tr> <td>USB Port1</td><td>Device</td><td>Device</td><td>Device</td><td>Host</td></tr> </table>					{D,H}	2'b00	2'b01	2'b10	2'b11	USB Port0	Host	Disable	Host	Disable	USB Port1	Device	Device	Device	Host	
{D,H}	2'b00	2'b01	2'b10	2'b11																			
USB Port0	Host	Disable	Host	Disable																			
USB Port1	Device	Device	Device	Host																			
0	H	R/W	Select the USB Port0 or USB Port1 as the USB host. When H=0, Host Select USB Port0. When H=1, Host Select USB Port1																				

P_SCUA_VDAC_CFG
0x93007040
Video DAC Configur

Bit	31	30	29	28	27	26	25	24
Function	-							

Default 0 0 0 0 0 0 0 0

23	22	21	20	19	18	17	16
-							

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
-							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
-	CLK_SEL	74_25M_EN	YPBPR_EN	E	U	T	P

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:4]	Reserved	-	-					
6	CLK_SEL		0: Not invert VDAC clock 1:Invert VDAC clock					
5	74_25M_EN		0: Disable 74.25MHz clock 1: Enable 74.25M clock					
4	YPBPR_EN		0: Disable YPbPr 1: Enable YPbPr					
3	E	R/W	When E=1, the clock to VDAC is on. When E=0, the clock to VDAC is off					
2	U	R/W	Test only, up or down counter					
1	T	R/W	Test only					
0	P	R/W	When P=1, VDAC power down. When P=0, VDAC power up					

P_SCUA_APPLL_CFG
0x93007044
Audio PLL Configure

Bit	31	30	29	28	27	26	25	24
<i>DA_RATIO</i>								
Function	0	0	0	0	0	1	0	1

23	22	21	20	19	18	17	16
<i>AD_RATIO</i>							

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
-		A_PATH	R	C	-		

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
	APLL_AS[5:0]		E	F	S	P	

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description	Condition
[31:24]	DA_RATIO	R/W	Set the MCLK for I2S. $f_{i2s_mclk} = \frac{f_{APLL}}{DA_RATIO}$	
[23:16]	AD_RATIO	R/W	Set the MCLK for I2SRX. $f_{i2srx_mclk} = \frac{f_{APLL}}{AD_RATIO}$	
[15:13]	Reserved			
12	A_PATH	R/W	Audio controller fifo path select APLL divider Reset -> 0x93007000[14]	
11	R	R/W	Set R=1 to reset APLL. Set R=0 to release reset APLL	
10	C	R/W	Test Only	
[9:8]	Reserved	-		
[7:4]	APLL_AS	R/W	The value should must set to 010010B only	
3	E	R/W	When E=1, the clock to APLL is on. When E=0, the clock to APLL is off	
2	F	R/W	When F=1, SCLK for I2S is sourced from APLL. When F=0, SCLK for I2S is sourced from I/O PAD	
1	S	R/W	S=0 for 48KHz Audio, and S=1 for 44.1KHz Audio If S=0, the f_{APLL} is 73.728MHz. If S=1, the f_{APLL} is 67.7376MHz.	
0	P	R/W	When P=0, APLL power down. When P=1, APLL power up	

The P_SCUA_LCD_CLK_CFG is a control register for TFT0.

P_SCUA_LCD_CLK_CFG 0x93007080 TFT0 Clock Configure								
Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0
LCD_CLK_SEL								
23	22	21	20	19	18	17	16	
0	0	0	0	0	0	0	0	
LCD_CLK_EN								
15	14	13	12	11	10	9	8	
0	0	0	0	0	0	0	0	
LCD_CLK_RATIO								
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	1	0	

$$f_{clk_lcd} = \frac{f_{clk_ref_ceva}}{(R[7:0]+1)}$$

When changing the LCD_CLK_RATIO [7:0], LCD_CLK_EN should be 0. The correct procedure to the enable clk_lcd is :

- Step1: Set LCD_CLK_EN =0 to disable clk_lcd
 - Step2: Set LCD_CLK_RATIO [7:0] to change the divide ratio
 - Step3: Set LCD_CLK_EN =1 to enable clk_lcd
- The procedure to disable clk_lcd is :

- Step1: Set LCD_CLK_EN =0 to disable clk_lcd.

Bit	Function	Type	Description					Condition
[31:19]	Reserved	-	-					
[18:16]	LCD_CLK_SEL	R/W	Select LCD clock source , xx1:LCD clock source is from XTAL 27MHz 000:LCD clock source is from SPLL 110: LCD clock source is from USBPHY 96MHz(USB must enable)					
[15:9]	Reserved	-	-					
8	LCD_CLK_EN	R/W	0: clk_lcd is off. 1: clk_lcd is on					
[7:0]	LCD_CLK_RATIO	R/W	Set the ratio for clk_lcd					

P_SCUA_CSI_CLK_CFG 0x93007084 CSI Clock Configure								
Bit	31	30	29	28	27	26	25	24
Function	CSI_PCLK_DELAY							
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	CSI_CLK_SEL							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	CSI_CLK_EN							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	CSI_CLK_RATIO							
	0	0	0	0	0	0	1	0

The clk_csi is the clock to the CMOS Sensor Interface (CSI). It's a clock source could be one of clk_ref_ceva or 96Mhz.

$$f_{clk_csi} = \frac{f_{clk_ref_ceva}}{(R[7:0]+1)}$$

When changing the CSI_CLK_RATIO [7:0], CSI_CLK_EN should be 0. The correct procedure to the enable clk_lcd is :

Step1: Set CSI_CLK_EN =0 to disable clk_csi

Step2: Set CSI_CLK_RATIO [7:0] to change the divide ratio

Step3: Set CSI_CLK_EN =1 to enable clk_csi

The procedure to disable clk_csi is :

Step1: Set CSI_CLK_EN =0 to disable clk_csi.

Bit	Function	Type	Description	Condition
[31:29]	Reserved	-	-	
[28:24]	CSI_PCLK_DELAY	R/W	PCLK delay	
16	CSI_CLK_SEL	R/W	Select CSI clock source , 0:CSI clock source is from SPLL 1: CSI clock source is from USBPHY 96MHz(USB must enable)	
[15:9]	Reserved	-	-	
8	CSI_CLK_EN	R/W	0: clk_csi is off. 1: clk_csi is on	
[7:0]	CSI_CLK_RATIO	R/W	Set the ratio for clk_csi	

P_SCUA_I2S_BCK_CFG 0x93007090 I2S BCK Configure							
Bit	31	30	29	28	27	26	25
Function					-		
Default	0	0	0	0	0	0	0
	23	22	21	20	19	18	17
				-			
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
				-			I2S_BCK_CNT_EN
	0	0	0	0	0	0	0
	7	6	5	4	3	2	1
							I2S_BCK_RATIO
	0	0	0	0	0	0	0

The i2s_mclk, which is generated from APLL output clock, is the clock to the I2S controller. The control register locates at P_SCUA_APLL_CFG named APLL_CFG. P_SCUA_APLL_CFG [31:24] defines the clock divide ratio of ADC (i2s_rx). P_SCUA_APLL_CFG [23:16] defines the clock divide ratio of DAC (i2s_tx).

The frequency formula is:

$$\left\{ \begin{array}{l} f_{APLL} = \begin{cases} 73.728MHz & \text{when } APLL_CFG[1] = 0 \\ 67.7376MHz & \text{when } APLL_CFG[1] = 1 \end{cases} \\ DA_RATIO = APLL_CFG[23:16] \\ AD_RATIO = APLL_CFG[31:24] \\ f_{i2s_mclk} = \frac{f_{APLL}}{DA_RATIO} \\ f_{i2s_bck} = \frac{f_{i2s_mclk}}{(I2S_BCK_CFG[7:0]+1)} \end{array} \right.$$

The procedure to set the ratio and turn on the i2s_bck is :

Step1. Disable the i2s_bck. Set I2S_BCK_CNT_EN = 0.

Step2. Set the divide ratio to I2S_BCK_RATIO.

Step3. Enable the i2s_bck. Set I2S_BCK_CNT_EN = 1.

The new divide ratio only update to the i2s_bck when I2S_BCK_CNT_EN = 0 to 1. The procedure to turn off the i2s_bck is just set I2S_BCK_CNT_EN = 0.

Bit	Function	Type	Description					Condition
[31:9]	resvered	-						
8	I2S_BCK_CNT_EN	R/W	0: disable i2s_bck 1: enable i2s_bck					
[7:0]	I2S_BCK_RATIO	R/W	I2S_BCK clock ratio					

P_SCUA_UART_CFG		0x93007094								UART Clock Configure											
Bit	Function	31	30	29	28	27	26	25	24												
Function		-																			
Default	0	0	0	0	0	0	0	0	0	23	22	21	20	19	18	17	16				
		0	0	0	0	0	1	0	1	UART2_CLK				UART0_CLK							
		15	14	13	12	11	10	9	8												
		0	0	0	0	0	0	0	0	UART_CLK_EN											

7	6	5	4	3	2	1	0
UART_CLK_RATIO							
0	0	0	0	0	0	1	0

The uclk, which is generated from clk_ref_ceva or bypass to XTAL, is the clock to the UART baud rate controller (please refer to UART section). If the P_SCUA_UART_CFG [8] is 0, the clk_uart is off. P_SCUA_UART_CFG [7:0] defines the clock divide ratio to the clk_uart. P_SCUA_UART_CFG [16] or P_SCUA_UART_CFG [18] is used to select which uclk will bypass to XTAL clock source.

The frequency formula is :

$$f_{uclk} = \frac{f_{clk_ref_ceva}}{(UART_CLK_RATIO[7:0]+1)}$$

The procedure to set the ratio and turn on the uclk is :

Step1. Select which uclk will be bypassed to XTAL

Or

Step1. Disable the uclk. Set P_SCUA_UART_CFG [8] = 0.

Step2. Set the divide ratio to P_SCUA_UART_CFG [7:0].

Step3. Enable the uclk. Set P_SCUA_UART_CFG [8] = 1.

The new divide ratio only update to the uclk when P_SCUA_UART_CFG [8] = 0 to 1. The procedure to turn off the uclk is just set P_SCUA_UART_CFG [8] = 0.

Bit	Function	Type	Description	Condition
[31:18]	Reserved	-	-	
18	UART2_CLK		Select UART2 clock source , 0:SPLL 1:XTAL 27MHz and bypass clock divider	
17	Reserved	-		
16	UART0_CLK	R/W	Select UART0 clock source , 0:SPLL 1:XTAL 27MHz and bypass clock divider	
[15:9]	Reserved	-	-	
8	UART_CLK_EN	R/W	0: clk_uart is off. 1: clk_uart is on	
[7:0]	UART_CLK_RATIO	R/W	Set the ratio for clk_uart,only valid whe clock select from SPLL.	

P_SCUA_SYS_SEL	0x930070E0								System Select
Bit	31	30	29	28	27	26	25	24	
Function	-								SPU_CLK_SEL
Default	0	0	0	0	0	0	0	0	

23	22	21	20	19	18	17	16
-							
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
SEN0_CLK SEL							
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
SPU_CLK	SEN1_EN	PPU_CLK27	PPU_CLK74	PPU_DSUB	YPBPR_SEL	LCD_SEL	TV_SEL
0	0	0	0	0	0	1	1

Bit	Function	Type	Description	Condition
[31:18]	Reserved	-	-	
[25:24]	SPU_CLK_SEL	R/W	SPU clock select: SPUMCLK = sys_clk/ (SPU_CLK_SEL[1:0] +1)	
[23:16]	Reserved	-		
15	SEN0_CLK	R/W	Sensor 0 clock enable 1: Enable 0: Disable	
14	SEN0_CLK_SEL	R/W	Sensor 0 clock select when using 2th interface 0: B_CMOS[1] 1: B_UART_MISC[1]	
13	SEN1_MIPI	R/W	Sensor 1 MIPI enable 1: Enable 0: Disable	
12	CDSP_MIPI	R/W	CDSP MIPI enable 1: Enable 0: Disable	
11	CDSP_CSI2_SEL	R/W	CDSP clock select when using 2th interface 0: B_CMOS[1] 1: B_UART_MISC[1]	
10	CDSP_POST	R/W	CDSP post processing mode enable 1: Enable 0: Disable	
9	CDSP_YUV	R/W	CDSP YUV data format mode enable 1: Enable 0: Disable	

Bit	Function	Type	Description	Condition
8	CDSP_CLK	R/W	CDSP clock enable 1: Enable 0: Disable	
7	SPU_CLK	R/W	SPU 27MHZ clock enable 1: Enable 0: Disable	
6	SEN1_EN	R/W	Sensor 1 clock enable 1: Enable 0: Disable	
5	PPU_CLK27	R/W	PPU 27MHz clock enable,for TV1 or YPBPR 480i 1: Enable 0: Disable	
4	PPU_CLK74	R/W	PPU 74.25MHz clock enable,for YPBPR 720P 1: Enable 0: Disable	
3	PPU_DSUB	R/W	DSUB enable,only for TV1 1: Enable 0: Disable	
2	YPBPR_SEL	R/W	VDAC data path from TV0 or TV1 1: TV1 0: TV0	
1	LCD_SEL	R/W	LCD data path from TFT0 or TFT1 1: TFT1 0: TFT0	
0	TV_SEL	R/W	VDAC data path from TV0 or TV1 1: TV1 0: TV0	

P_SCUA_CSI2_CLK_CFG 0x930070E8 CSI2 Clock Configure							
Bit	31	30	29	28	27	26	25
Function	<i>CSI2_PCLK_DELAY</i>						
Default	0	0	0	0	0	0	0
	23	22	21	20	19	18	17
	0	0	0	0	0	0	0
	23	22	21	20	19	18	17
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	0	0	0	0	0	0	0
	-						

7	6	5	4	3	2	1	0
CSI2_CLK_RATIO							
0	0	0	0	0	0	1	0

The clk_csi2 is the clock of the 2th sensor. It should be equal or small than clk_sys

$$f_{clk_csi2} = \frac{f_{clk_ref_ceva}}{(R[7:0]+1)}$$

Bit	Function	Type	Description	Condition
[31:29]	Reserved	-	-	
[28:24]	CSI2_PCLK_DELAY	R/W	PCLK input delay	
16	CSI2_CLK_SEL	R/W	Select CSI2 clock source , 0:CSI clock source is from SPLL 1: CSI clock source is from USBPHY 96MHz(USB must enable)	
[15:9]	Reserved	-	-	
8	CSI2_CLK_EN	R/W	0: CSI2 clock is off. 1: CSI2 clock is on	
[7:0]	CSI2_CLK_RATIO	R/W	Set the ratio for CSI2 clock	

The P_SCUA_CDSP_PCLK is only available when CDSP POST is enabled.

P_SCUA_CDSP_PCLK 0x930070EC CDSP Clock Configure							
Bit	31	30	29	28	27	26	25
Function	-						
Default	0	0	0	0	0	0	0
CDSP_CLK_SEL							
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
-							CDSP_CLK_EN
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
CDSP_RATIO							
0	0	0	0	0	0	1	0



The `clk_cdsp` is the clock of the CDSP. It should be equal or small than `clk_sys`.

$$f_{clk_cdsp} = \frac{f_{clk_ref_ceva}}{(R[7:0]+1)}$$

Bit	Function	Type	Description	Condition
[31:17]	Reserved	-	-	
16	CDSP_CLK_SEL	R/W	Select CDSP clock source , 0:CDSP clock source is from SPLL 1: CDSP clock source is from USBPHY 96MHz(USB must enable)	
[15:9]	Reserved	-	-	
8	CDSP_CLK_EN	R/W	0: CDSP clock is off. 1: CDSP clock is on	
[7:0]	CDSP_CLK_RATIO	R/W	Set the ratio for CDSP clock	

P_SCUA_MIPI_2CH_PCLK		0x930070F0		MIPI 2 channel Clock Configure							
Bit	31 30 29 28 27 26 25 24										
Function	-										
Default	0	0	0	0	0	0	0	0			
	23 22 21 20 19 18 17 16								MIPI_CLK_SEL		
	0	0	0	0	0	0	0	0			
	15 14 13 12 11 10 9 8								MIPI_CLK_EN		
	0	0	0	0	0	0	0	0			
	7 6 5 4 3 2 1 0								MIPI_RATIO		
	0	0	0	0	0	0	1	0			

When mipi 2-ch is enabled, configure this register to generate pclk. It should be equal or small than clk_sys.

$$f_{clk_mipi} = \frac{f_{clk_ref_ceva}}{(R[7:0]+1)}$$

Bit	Function	Type	Description					Condition
[31:17]	Reserved	-	-					
16	MIPI_CLK_SEL	R/W	Select MIPI clock source , 0:MIPI clock source is from SPLL 1: MIPI clock source is from USBPHY 96MHz(USB must enable)					
[15:9]	Reserved	-	-					
8	MIPI_CLK_EN	R/W	0: MIPI clock is off. 1: MIPI clock is on					
[7:0]	MIPI_CLK_RATIO	R/W	Set the ratio for MIPI clock					

3.4.2 SCUB

Peripheral reset, please reference to table 1 for the peripheral device of each source. Please set 1 to reset each device.

P_SCUB_PERI_RST 0x90005000 SCUB Peripheral Reset								
Bit	31	30	29	28	27	26	25	24
Function						-		
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
					-			
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
					Peripheral[15:0]			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
					Peripheral[15:0]			
	0	0	0	0	0	0	0	0

P_SCUB_SPLL_CFG0 is a control register for SPLL, please refer to section 1.6 SPLL for more information.

P_SCUB_SPLL_CFG0 0x90005004 SPLL Configure 0								
Bit	31	30	29	28	27	26	25	24
Function	-			SPLL_M		SPLL2_TEST	SPLL_TEST	
Default	0	1	1	0	1	1	0	0

23	22	21	20	19	18	17	16
XTAL_RESET	APR_REF_EN	X27ML_EN	SPLL_REF_EN	WEAK_27M	ARM_CLKSEL		
0	1	1	1	0	0	0	0
15	14	13	12	11	10	9	8
SPLL_PD	CEVA_CLKSEL		SPLL2_R				
0	0	0	0	1	1	0	0
7	6	5	4	3	2	1	0
		SPLL_N		SPLL_R			
1	1	0	1	1	0	1	1

Bit	Function	Type	Description	Condition
31	Reserved	-	-	
[30:26]	SPLL_M	R/W	SPLL clock input divider	
25	SPLL2_TEST	R/W	SPLL2 spread test	
24	SPLL_TEST	R/W	SPLL spread test	
23	XTAL_RESET	R/W	XTAL reset	
22	APR_REF_EN	R/W	apr clock enable apr clock is 27MHz, used for:PPU_27MCLK, SPU_27MCLK, IO_TRAP_CLK, one of Audio/Video/LCD/Sensor/Uart clock source, WDT,random generator,Hardmacro_PWM, mem_repair 1 : enable 0 : disable	
21	X27ML_EN	R/W	XTAL 27M enable 1 : enable 0 : disable	
20	SPLL_REF_EN	R/W	SPLL reference clock enable 1 : enable 0 : disable	
19	WEAK_27M	R/W	XTAL 27M weak mode enable 1 : weak mode 0 : normal mode	
[18:16]	ARM_CLKSEL	R/W	Clock selection to the ARM clock 0: 27, 1: RTC, 2: DIV2, 3: DIV3, 4: DIV1	
15	SPLL_PD	R/W	SPLL power down 1: power down 0: power up	

Bit	Function	Type	Description					Condition
[14:12]	CEVA_CLKSEL	R/W	0: 27, 1: RTC, 2: DIV2, 3: DIV3, 4: DIV1					
[11:10]	SPLL2_R	R/W	SPLL2 post divider					
[9:2]	SPLL_N	R/W	SPLL feedback divider					
[1:0]	SPLL_R	R/W	SPLL post divider					

P_SCUB_INTR_STATUS 0x90005008 Timer Interrupt Status

Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0
-								
Bit	23	22	21	20	19	18	17	16
Function	-							
Default	0	0	0	0	0	0	0	0
-								
Bit	15	14	13	12	11	10	9	8
Function	-				INT_TIMER1			
Default	0	0	0	0	0	0	0	0
-								
Bit	7	6	5	4	3	2	1	0
Function	-				INT_TIMER0			
Default	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:13]	Resvered							
[12:8]	INT_TIMER1	R	Timer 1 interrupt status					
[7:5]	Resvered							
[4:0]	INT_TIMER0	R	Timer 0 interrupt status					

P_SCUB_TIMER_ICE_EN 0x9000500C Timer Stop Enable When ICE Connect

Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0
-								
Bit	23	22	21	20	19	18	17	16
Function	-							
Default	0	0	0	0	0	0	0	0
-								

15	14	13	12	11	10	9	8	
-		TIMER1_ICE_STOP						

0	0	0	1	1	1	1	1	
-		TIMER0_ICE_STOP						

Bit	Function	Type	Description					Condition
[31:13]	Resvered							
[12:8]	TIMER1_ICE_STOP	R/W	1 : Timer will stop when ARM ICE connected 0 : Timer will not stop when ARM ICE connected					
[7:5]	Resvered							
[4:0]	TIMER0_ICE_STOP	R/W	1 : Timer will stop when ARM ICE connected 0 : Timer will not stop when ARM ICE connected					

P_SCUB_TIMER_EXT_CTRL 0x90005010
Timer External Trigger Down/Up

Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16	
-		T0_RESET						

15	14	13	12	11	10	9	8	
-								

7	6	5	4	3	2	1	0	
-		T0_EXT_TRI						

Bit	Function	Type	Description					Condition
[31:21]	Resvered	-	-					
[20:16]	T0_RESET	R/W	1 : Timer0 reset					
[15:5]	Resvered	-	-					

Bit	Function	Type	Description						Condition
[4:0]	T0_EXT_TRI	R/W	Timer0 external trigger up/down 0: Down Counting 1: Up Counting						

P_SCUB_REV **0x90005018** **IC Version**

Bit	31	30	29	28	27	26	25	24	
Function	REV								
Default	0	1	0	1	0	0	1	1	
23	22	21	20	19	18	17	16		
Function	REV								
0	1	0	1	0	0	0	0	0	
15	14	13	12	11	10	9	8		
Function	REV								
0	0	1	1	1	0	0	0	0	
7	6	5	4	3	2	1	0		
Function	REV								
0	0	1	1	0	0	0	0	0	

Bit	Function	Type	Description						Condition
[31:0]	REV	R	IC version						

P_SCUB_RAND0 **0x9000501C** **Random Seed**

Bit	31	30	29	28	27	26	25	24	
Function	RAND_SEED								
Default	1	1	0	1	1	1	1	0	
23	22	21	20	19	18	17	16		
Function	RAND_SEED								
1	0	1	0	0	1	0	1	0	
15	14	13	12	11	10	9	8		
Function	RAND_SEED								
1	1	0	1	0	0	0	0	0	

7	6	5	4	3	2	1	0
RAND_SEED							
0	0	0	0	1	0	1	1

Bit	Function	Type	Description					Condition
[31:0]	RAND_SEED	R/W	RAND Seed [31:0] Supervisor Mode Read Only Only Writeable for One Time, default = 0xDEADD00B					

Peripheral clock enable, please reference to table 1 for the peripheral device of each source. Please set 1 to enable each device clock. If set 0, the clock of that peripheral module is gated

P_SCUB_PERI_CLKEN 0x90005020 SCUB Clock Enable							
Bit	31	30	29	28	27	26	25
Function	-						
Default	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Peripheral[15:0]							
1	1	1	1	1	1	1	1

7	6	5	4	3	2	1	0
Peripheral[15:0]							
1	1	1	1	1	1	1	1

Peripheral dynamic clock gating enable, please reference to table 1 for the peripheral device of each source. The dynamic gating can save the power consumption when the module is working. Please set 1 to enable dynamic clock gating and set 0 to disable dynamic clock gating.

P_SCUB_PERI_DGCLKEN 0x90005024 SCUB Dynamic Gated Clock Enable							
Bit	31	30	29	28	27	26	25
Function	-						
Default	1	1	1	1	1	1	1

23	22	21	20	19	18	17	16
-							
1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8
Peripheral[15:0]							

1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
Peripheral[15:0]							

1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

P_SCUB_UPDATE_RATIO 0x90005028 ARM Clock Ratio Update Control

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
-							

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
-	-	-	-	-	APB_RATIO_U	AHB_RATIO_U	ARM_RATIO_U

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:3]	Resvered	-	-					
2	APB_RATIO_U	R/W	ARM APB Ratio Updated 1: update					
1	AHB_RATIO_U	R/W	ARM AHB Ratio Updated 1: update					
0	ARM_RATIO_U	R/W	ARM Ratio Updated 1: update					

P_SCUB_PWRC_CFG
0x90005040
PWRC Configure

Bit	31	30	29	28	27	26	25	24
Function	PWRC_LVRZ	-	PWRC_PWRON1	PWRC_PWRON0			-	
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
				-				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
				-				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
				-	SE	BT	RC	RE
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
31	PWRC_LVRZ	R	Test only					
30	Resvered	-	-					
29	PWRC_PWRON1	R	The status of external power on 1 from user push switch/putton					
28	PWRC_PWRON0	R	The status of external power on 0 from user push switch/putton					
[27:4]	Resvered	-	-					
3	SE	R/W	1 : Enable test, 0 : disable test					
2	BT	R/W	1 : Li-ion battery(3.5V~4.2V), 0 : Alkaline battery(1.8V~3.6V)					
1	RC	R/W	1 : Enter operation mode, 0 : suspend mode					
0	RE	R/W	1 : DCDC enable, 0 : DCDC suspend					

P_SCUB_RAND1
0x90005048
Random Seed1

Bit	31	30	29	28	27	26	25	24
Function					-			
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							
0	0	0	0	0	0	0	0
RAND_SEED1							
0	0	0	1	0	0	1	1
7	6	5	4	3	2	1	0
RAND_SEED1							
0	1	0	0	0	1	0	1

Bit	Function	Type	Description					Condition
[31:16]	Reversed	-	-					
[15:0]	RAND_SEED1	R/W	RAND Seed [47:32] Supervisor Mode Read Only Only Writeable for One Time, default = 0x1345					

P_SCUB_DBGRQ_CTRL 0x90005058 ARM Dynamic Gated Clock Control

Bit	31	30	29	28	27	26	25	24
-								
Function	0	0	0	0	0	0	0	0
Default								
23	22	21	20	19	18	17	16	
-								
0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	
-								
0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	
-								DBGRQ_CTRL
0	0	0	0	0	0	0	0	1

Bit	Function	Type	Description	Condition
[31:1]	Reversed	-	-	
0	DBGRQ_CTRL		Disable ARM1176JZF-S Dynamic Gated Clock 1 : No Dynamic Gated Clock 0 : With Dynamic Gated Clock	

P_SCUB_SPLL_CFG1 and P_SCUB_SPLL_CFG2 are control register for SPLL, please refer to section 1.6 SPLL for more information.

P_SCUB_SPLL_CFG1 0x9000505C SPLL Configure 1								
Bit	31	30	29	28	27	26	25	24
Function	-				SPLL2_PD	SPLL2_SS CGON	SPLL_SS CGON	CLKSEL
Default	0	0	0	0	1	0	0	0

23	22	21	20	19	18	17	16
SPLL2_MDS							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SPLL_MDS							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SPLL2_N							
0	0	1	1	0	1	1	0

Bit	Function	Type	Description	Condition
[31:28]	Reversed	-		
27	SPLL2_PD	R/W	SPLL2 power down 1: power down 0: power up	
26	SPLL2_SS CGON	R/W	1:SPLL2 spread disable 1:SPLL2 spread enable	
25	SPLL_SS CGON	R/W	0:SPLL2 spread disable 1:SPLL spread enable	
24	CLKSEL	R/W	0: ARM clock use SPLL 1: ARM clock use SPLL2	
[23:16]	SPLL2_MDS	R/W	SPLL2 spread parameter	

Bit	Function	Type	Description					Condition
[15:8]	SPLL_MDS	R/W	SPLL spread parameter					
[7:0]	SPLL2_N	R/W	SPLL2 feedback divider					

P_SCUB_SPLL_CFG2 0x90005060 SPLL Configure 2								
Bit	31	30	29	28	27	26	25	24
Function					-			
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
				-				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
				-				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	1	1	0	1	1
	SPLL2_M							

Bit	Function	Type	Description								Condition
[31:5]	Reversed	-	-								
[4:0]	SPLL2_M		SPLL2 clock input divider								

The pads are grouped based on the group ID (GID). Each pads group is selected by two bits of PGSx. ex. PGS0[1:0] for pads with GID = 0, and PGS0[3:2] for pads with GID= 1, and PGS3[31:0] for pad with GID=63~48.

	[31:30]	[29:28]	[27:26]	[25:24]	[23:22]	[21:20]	[19:18]	[17:16]	[15:14]	[13:12]	[11:10]	[9:8]	[7:6]	[5:4]	[3:2]	[1:0]
PGS0	GID15	GID14	GID13	GID12	GID11	GID10	GID9	GID8	GID7	GID6	GID5	GID4	GID3	GID2	GID1	GID0
PGS1	GID31	GID30	GID29	GID28	GID27	GID26	GID25	GID24	GID23	GID22	GID21	GID20	GID19	GID18	GID17	GID16
PGS2	GID47	GID46	GID45	GID44	GID43	GID42	GID41	GID40	GID39	GID38	GID37	GID36	GID35	GID34	GID33	GID32
PGS3	GID63	GID62	GID61	GID60	GID59	GID58	GID57	GID56	GID55	GID54	GID53	GID52	GID51	GID50	GID49	GID48

P_SCUB_PGS0
0x90005080
PAD Group Selection 0

Bit	31	30	29	28	27	26	25	24
Function	GID15		GID14		GID13		GID12	
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
GID11		GID10		GID9		GID8	

15	14	13	12	11	10	9	8
GID7		GID6		GID5		GID4	

7	6	5	4	3	2	1	0
GID3		GID2		GID1		GID0	

P_SCUB_PGS1
0x90005084
PAD Group Selection 1

Bit	31	30	29	28	27	26	25	24
Function	GID31		GID30		GID29		GID28	
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
GID27		GID26		GID25		GID24	

15	14	13	12	11	10	9	8
GID23		GID22		GID21		GID20	

7	6	5	4	3	2	1	0
GID19		GID18		GID17		GID16	

P_SCUB_PGS2
0x90005088
PAD Group Selection 2

Bit	31	30	29	28	27	26	25	24
Function	GID47		GID46		GID45		GID44	
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
GID43		GID42		GID41		GID40	

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
GID39		GID38		GID37		GID36	

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
GID35		GID34		GID33		GID32	

0 0 0 0 0 0 0 0

P_SCUB_PGS3
0x9000508C
PAD Group Selection 3

Bit	31	30	29	28	27	26	25	24
Function	GID63		GID62		GID61		GID60	
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
GID59		GID58		GID57		GID56	

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
GID55		GID54		GID53		GID52	

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
GID51		GID50		GID49		GID48	

0 0 0 0 0 0 0 0

P_SCUB_ARM_RATIO
0x900050D0
ARM Clock Ratio

Bit	31	30	29	28	27	26	25	24
Function					-			
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
-							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
ARM_RATIO							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:6]	Reserved							
[5:0]	ARM_RATIO	R/W	ARM clock ratio					

P_SCUB_ARM_AHB_RATIO 0x900050D4
ARM AHB Clock Ratio

Bit	31	30	29	28	27	26	25	24
-								

Default	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---

23	22	21	20	19	18	17	16
-							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
-							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
ARM_AHB_RATIO							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:6]	Reserved							
[5:0]	ARM_AHB_RATIO	R/W	ARM AHB Ratio					

P_SCUB_ARM_APB_RATIO 0x900050D8
ARM APB Clock Ratio

Bit	31	30	29	28	27	26	25	24
-								

Default	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---

23	22	21	20	19	18	17	16
-							
0	0	0	0	0	0	0	0
-							
15	14	13	12	11	10	9	8
-							
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
ARM_APB_RATIO							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:6]	Reserved							
[5:0]	ARM_APB_RATIO	R/W	ARM APB Ratio					

P_SCUB_SYS_CNT_EN controls the clock on/off of the clk_sys, clk_sys_ahb, and clk_sys_apb.

P_SCUB_SYS_CNT_EN 0x900050DC System Counter Enable

Bit	31	30	29	28	27	26	25	24
Function	CC							-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
				APB	AHB	RT	SYS
0	0	0	0	0	0	0	1

Bit	Function	Type	Description					Condition
31	CC	R/W	Command Check This bit will inverse automatically after writing this register, for the purpose of checking the command status. Because there are some cycles latency for the "writing command". For example, if this bit is 1, after writing SYS_CNT_EN, this bit will be 0 if the target clock is "Enabled or Disabled".					
[30:4]	Reserved	-						
3	APB	R/W	SYS APB Counter Enable					
2	AHB	R/W	SYS AHB Counter Enable					
1	RT	R/W	SYS RT Counter Enable					
0	SYS	R/W	SYS Counter Enable					

P_SCUB_PIN_MUX 0x90005144 IO Pad Mux Configure								
Bit	31	30	29	28	27	26	25	24
Function	MIPI_SPI	MIPI_SPICS0	MIPI_SPICS1	MIPI_TE	MIPI_CSIHD	MIPICLK_CSI	MIPICLK_LCD	MIPI_CSICTRL
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	-	-	SD1_CMD_GPIO	SD0_CMD_GPIO	NAND_CTRL_DEDI	KEY10_CSIHD	I2C_PWM2	KEY3_PWM1
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	UR_I2S_CS	KEY_NAND_CTRL	KEY_NAND_CS	ANO_CSI	ANO_SPI	TFT_TE	NAND_RDY	I80_READ
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	UARTMC_I2C_MODE	TCON_GPIO	CSI_FIELD_GPIO	NAND1_CS_GPIO	NAND0_CS_GPIO	LCDD_CSID	CEVA_JTAG	ARM_JTAG
	0	0	0	0	0	0	0	1

Bit	Function	Type	Description	Condition
31	MIPI_SPI	R/W	1: MIPI CLKN / DATAP/ DATAN is SPI signal	
30	MIPI_SPICS0	R/W	1: MIPI CLKP is SPICSI0	
29	MIPI_SPICS1	R/W	1: MIPI DATAN is SPICSI1	
28	MIPI_TE	R/W	1: MIPI DATAP is TFT_TE	
27	MIPI_CSIHD	R/W	1: MIPI DATAP1/DATAN1 for CDSP raw data bit [1:0]	
26	MIPICLK_CSI	R/W	1: B_CMOS[0] is miipi clock root	
25	MIPICLK_LCD	R/W	1: B_DISP[12] is miipi clock root	
24	MIPI_CSICTRL	R/W	1: MIPI channel0 4 signal is Sensor control signal	
[23:22]	-	-		
21	SD1_CMD_GPIO	R/W	1: Sdcard interface 1 working but B_KEYSCAN[3] (SD0_CMD) can be GPIO	
20	SD0_CMD_GPIO	R/W	1: Sdcard interface 0 working but B_SD_CARD[1] (SD0_CMD) can be GPIO	
19	NAND_CTRL_DEDI	R/W	1: B_NAND[11:8] NAND CTRL pin dedicated (used for partial nand share mode)	
18	KEY10_CSIHD	R/W	1: B_KEYSCAN[1:0] for CDSP raw data bit [1:0]	
17	I2C_PWM2	R/W	1: B_I2C_SDA0 is PWM2	
16	KEY3_PWM1	R/W	1: B_KEYSCAN[3] is PWM1	
15	UR_I2S	R/W	1: B_UART_BT / B_I2S is I2s mode	
14	KEY_NAND_CS	R/W	1: B_KEYSCAN[3] is nand CS	
13	KEY_NAND_CTRL	R/W	1: B_KEYSCAN[7:4] is nand ctrl	
12	ANO_CSI	R/W	1: B_UART_BT/B_UART_MISC/B_KEYSCAN[7:0] is Sensor signal (to Sensor 0 controller only)	
11	ANO_SPI	R/W	1: B_UART_BT and B_I2C_SDA0 is SPI signal	
10	TFT_TE	R/W	1: B_DISP[13] is TFT_TE	
9	NAND_RDY	R/W	1: only use one NAND rdy:B_NAND[13]	
8	I80_READ	R/W	1: B_DISP[19:4] is input	
7	UARTMC_I2C	R/W	1: B_UART_MISC is I2C interface	
6	TCON_MODE	R/W	1: B_DISP[24:22] is tcon signal	
5	CSI_FIELD_GPIO	R/W	1: CSI FIELD as GPIO	
4	NAND1_CS_GPIO	R/W	1: NAND1 CS as GPIO	
3	NAND0_CS_GPIO	R/W	1: NAND0 CS as GPIO	
2	LCDD_CSID	R/W	1: B_DISP[19:12] is cmos data	
1	CEVA_JTAG	R/W	1: CEVA jtag use keyscan	
0	ARM_JTAG	R/W	1: Use ARM JTAG	

P_SCUB_CP15SDISABLE 0x90005150 ARM CP15 Access Disable								
Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
-								
0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	
-								
0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	
-								
0	0	0	0	0	0	0	0	0
CP15_DIS								

Bit	Function	Type	Description						Condition
[31:7]	Reserved	-							
[6:0]	CP15_DIS	R/W	1: CP15 access disable 0: CP15 access enable						

P_SCUB_ARM_JTAG 0x90005154 AMR JTAG Information								
Bit	31	30	29	28	27	26	25	24
Function	DBGVERSION							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
-								
0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	
-								
0	0	0	0	0	1	1	1	1
7	6	5	4	3	2	1	0	
DBGMANID								
0	0	0	0	1	1	1	0	0

Bit	Function	Type	Description				Condition
[31:28]	DBGVERSION	R	Debug version				
[27:12]	Reserved	-					
[11:1]	DBGMANID	R	Debug main ID				
0	Reserved	-					

P_SCUB_TZ_CFG **0x90005158** **TrusZone Secure and Non-Secure Region Configure**

Bit	31	30	29	28	27	26	25	24
Function	-	-	AES	EFUSE	BCH	NAND0	USB_DEV_CTRL	USB_DEV_DATA
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
UHOST	MS	SD1	SD0	SPI1	SPI0	DSP	ARM
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SCUD	SCUC	SCUB	SCUA	CEVA_L1_DRAM	CEVA_L2_RAM	IRAM	IROM
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
M7	M6	M5	M4	M3	M2	M1	M0
0	0	0	0	0	0	0	0

The P_SCUB_TZ_CFG can setup memory areas as secure or no-secure. The programs running on non-secure world can't access the secure memory and secure peripheral.

Bit	Function	Type	Description				Condition
[31:30]	Reserved	-					
29	AES	R/W	1: Set AES to secure peripheral 0: Set AES to non-secure peripheral				
28	EFUSE	R/W	1: Set EFUSE to secure peripheral 0: Set EFUSE to non-secure peripheral				
27	BCH	R/W	1: Set BCH to secure peripheral 0: Set BCH to non-secure peripheral				

Bit	Function	Type	Description	Condition
26	NAND0	R/W	1: Set NAND0 to secure peripheral 0: Set NAND0 to non-secure peripheral	
25	USB_DEV_CTRL	R/W	1: Set USB device control to secure peripheral 0: Set USB device control to non-secure peripheral	
24	USB_DEV_DATA	R/W	1: Set USB device data to secure peripheral 0: Set USB device data to non-secure peripheral	
23	UHOST	R/W	1: Set UHOST to secure peripheral 0: Set UHOST to non-secure peripheral	
22	MS	R/W	1: Set MS to secure peripheral 0: Set MS to non-secure peripheral	
21	SD1	R/W	1: Set SD1 to secure peripheral 0: Set SD1 to non-secure peripheral	
20	SD0	R/W	1: Set SD0 to secure peripheral 0: Set SD0 to non-secure peripheral	
19	SPI1	R/W	1: Set SPI1 to secure peripheral 0: Set SPI1 to non-secure peripheral	
18	SPI0	R/W	1: Set SPI0 to secure peripheral 0: Set SPI0 to non-secure peripheral	
17	DSP	R/W	1: Set DSP to secure peripheral 0: Set DSP to non-secure peripheral	
16	ARM	R/W	1: Set ARM(0x9000_XXXX) to secure peripheral 0: Set ARM(0x9000_XXXX)to non-secure peripheral	
15	SCUD	R/W	1: Set SCUD to secure peripheral 0: Set SCUD to non-secure peripheral	
14	SCUC	R/W	1: Set SCUC to secure peripheral 0: Set SCUC to non-secure peripheral	
13	SCUB	R/W	1: Set SCUB to secure peripheral 0: Set SCUB to non-secure peripheral	
12	SCUA	R/W	1: Set SCUA to secure peripheral 0: Set SCUA to non-secure peripheral	
11	CEVA_L1_DRAM	R/W	1: Set CEVA L1 Data RAM to secure region 0: Set CEVA L1 Data RAM to non-secure region	
10	CEVA_L2_RAM	R/W	1: Set CEVA L2 RAM to secure region 0: Set CEVA L2 RAM to non-secure region	
9	IRAM	R/W	1: Set internal ram to secure region 0: Set internal ram to non-secure region	
8	IROM	R/W	1: Set internal rom to secure region 0: Set internal rom to non-secure region	
7	M7	R/W	1: Set memory to secure region 0: Set memory to non-secure region	
6	M6	R/W	1: Set memory to secure region 0: Set memory to non-secure region	

Bit	Function	Type	Description	Condition
5	M5	R/W	1: Set memory to secure region 0: Set memory to non-secure region	
4	M4	R/W	1: Set memory to secure region 0: Set memory to non-secure region	
3	M3	R/W	1: Set memory to secure region 0: Set memory to non-secure region	
2	M2	R/W	1: Set memory to secure region 0: Set memory to non-secure region	
1	M1	R/W	1: Set memory to secure region 0: Set memory to non-secure region	
0	M0	R/W	1: Set memory to secure region 0: Set memory to non-secure region	

3.4.3 SCUC

Peripheral reset, please reference to table 1 for the peripheral device of each source. Please set 1 to reset each device.

P_SCUC_PERI_RST 0x92005000 SCUC Peripheral Reset							
Bit	31	30	29	28	27	26	25
Function							
Default	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
Peripheral[31:0]							
0	0	0	0	0	0	0	0
Peripheral[31:0]							
15	14	13	12	11	10	9	8
Peripheral[31:0]							
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
Peripheral[31:0]							
0	0	0	0	0	0	0	0

Peripheral clock enable, please reference to table 1 for the peripheral device of each source. Please set 1 to enable each device clock. If set 0, the clock of that peripheral module is gated.

P_SCUC_PERI_CLKEN 0x92005004 SCUC Peripheral Clock Enable							
Bit	31	30	29	28	27	26	25
Function							
Default	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
Peripheral[31:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
Peripheral[31:0]							

0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
Peripheral[31:0]							

0	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---

Peripheral dynamic clock gating enable, please reference to table 1 for the peripheral device of each source. The dynamic gating can save the power consumption when the module is working. Please set 1 to enable dynamic clock gating and set 0 to disable dynamic clock gating.

P_SCUC_PERI_DGCLKEN 0x92005008 SCUC Dynamic Clock Gating Enable

Bit	31	30	29	28	27	26	25	24
Peripheral[31:0]								
Default	1	1	1	1	1	1	1	1

23	22	21	20	19	18	17	16
Peripheral[31:0]							

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
Peripheral[31:0]							

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
Peripheral[31:0]							

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

P_SCUC_GC_CFG0 0x92005010 DDRPHY Configure 0

Bit	31	30	29	28	27	26	25	24
-								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
			-				
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
		-		DS		-	
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
				ID			
0	0	0	0	0	0	1	1

Bit	Function	Type	Description	Condition
[31:11]	Reserved	-		
10	DS	R/W	When DS=1, DDRPHY is controlled by CPU When DS=0 (default), DDRPHY is controlled by IO PADs (test mode only) The default value of DS is 0. "test mode" is the defalut status for the DDRPHY. Please set this bit to 1 before configuring the DDRPHY	
[9:7]	Reserved	-		
[6:0]	ID	R/W	I2C ID for the DDRPHY	

P_SCUC_ System Clock Ration Updata Control 0x92005028

SYS_RATIO_UPDATE

Bit	31	30	29	28	27	26	25	24
Function	UCF				-			
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
			-				
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0



7	6	5	4	3	2	1	0
-	CEVA_APB	CEVA_AHB	CEVA	SYS_APB	SYS_AHB	SYS_RT	SYS
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31	UCF	R/W	<p>Updated Check Flag</p> <p>This bit will inverse automatically after writing this register, for the purpose of checking the command status. Because there are some cycles latency for updating the clock ratio. For example, if this bit is 1 (or 0), after writing SYS_RATIO_UPDATE, this bit will be 0 (or 1) if the target clock ratio is updated or not.</p>	
[30:7]	Reserved	-		
6	CEVA_APB	R/W	CEVA APB Ratio Updated	
5	CEVA_AHB	R/W	CEVA AHB Ratio Updated	
4	CEVA	R/W	CEVA Ratio Updated	
3	SYS_APB	R/W	SYS APB Ratio Updated	
2	SYS_AHB	R/W	SYS AHB Ratio Updated	
1	SYS_RT	R/W	SYS RT Ratio Updated	
0	SYS	R/W	SYS Ratio Updated	

P SCUC DDRPHY CTRL2 0x92005040

DDRPHY Control 2

15	14	13	12	11	10	9	8
RAM_A2	CPEQDN	ICP_SEL2	-	SN_EN	IUP_SEL_G1		
0	0	0	0	0	0	1	1

7	6	5	4	3	2	1	0
IUP_SEL_G0	DLL_PWDN	PLL_PWDN	CPEQUP	PHASE_INC	VC		
1	1	0	0	0	0	0	1

Bit	Function	Type	Description	Condition
[31:19]	Reserved	-		
18	IUP_SEL2_G1	R/W	IUP_SEL[2] group 1 Delay Line Delay Step Compensation Please refer to IUP_SEL_G1	
17	IUP_SEL2_G0	R/W	IUP_SEL[2] group 0 Delay Line Delay Step Compensation Please refer to IUP_SEL_G0	
16	RAM_A6	R/W	DDR ODT control	
15	RAM_A2	R/W	DDR ODT control	
[14:13]	CPEQDN	R/W	Charge Pump Down Current Decrease 00: Normal 01: Decrease 12.5% 10: Decrease 25% 11: Decrease 37.5%	
12	ICP_SEL2	R/W	Charge Pump Current Selection,use with P_SCUC_DDRPHY_CTRL1 bit[8:7] Please reference P_SCUC_DDRPHY_CTRL1 bit[8:7]	
11	Reserved	-		
10	SN_EN	R/W	Share pad B_RAM_ADDR[11:0] for nand interface 1: Enable 0: Disable	
[9:8]	IUP_SEL_G1	R/W	IUP_SEL[1:0] group 1, use with IUP_SEL2_G1 Delay Line Delay Step Compensation IUP_SEL[2:0] group 1 = { IUP_SEL2_G1, IUP_SEL_G1} 000: Normal 001: Reduce 6% delay time 010: Reduce 4% delay time 011: Reduce 10% delay time 100: Reduce 4% delay time 101: Reduce 10% delay time 110: Reduce 8% delay time 111: Reduce 14% delay time	
[7:6]	IUP_SEL_G0	R/W	IUP_SEL[2:0] group 0, use with IUP_SEL2_G0 Delay Line Delay Step Compensation IUP_SEL[2:0] group 0 = { IUP_SEL2_G0, IUP_SEL_G0} 000: Normal 001: Reduce 6% delay time 010: Reduce 4% delay time 011: Reduce 10% delay time 100: Reduce 4% delay time 101: Reduce 10% delay time 110: Reduce 8% delay time 111: Reduce 14% delay time	

Bit	Function	Type	Description	Condition
5	DLL_PWDN	R/W	Power Saving Mode Selection 0: Normal 1: Power Down	
4	PLL_PWDN	R/W	Power Saving Mode Selection 0: Normal 1: Power Down	
[3:2]	CPEQUP	R/W	Charge Pump Down Current Increase 00: Normal 01: Increase 12.5% 10: Increase 25% 11: Increase 37.5%	
1	PHASE_INC	R/W	1: PLL Output Phase (0, 270) Leading 100ps than '0' Setting.	
0	VC	R/W	VC set 0: Reduce KVCO V TO I Resistor *1.5 1: Normal	

P_SCUC_DDRPHY_STATUS 0x92005044
DDRPHY Status

Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	-							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	-							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	-							LD
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:1]	Reserved	-		
0	LD	R	DDRPHY LD	

P_SCUC_DDRPHY_CTRL0 0x92005048
DDRPHY Control 0

Bit	31	30	29	28	27	26	25	24
Function	-		DDRPHY_S1					

Default 0 0 0 0 0 0 0 0

23	22	21	20	19	18	17	16
-	DDRPHY_S0						

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
-	DDRPHY_DS						

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
-	DDRPHY_LV						

0 0 0 0 0 0 0 0

Bit	Function	Type	Description			Condition
[31:29]	Reserved	-				
[28:24]	DDRPHY_S1	R/W	DDRPHY_S1,please refer to table 2 {ck_ckb, cmd_addr, dm, dqs, dq }			
[23:21]	Reserved	-				
[20:16]	DDRPHY_S0	R/W	DDRPHY_S0,please refer to table 2 {ck_ckb, cmd_addr, dm, dqs, dq }			
[15:13]	Reserved	-				
[12:8]	DDRPHY_DS	R/W	DDRPHY_DS,please refer to table 2 {ck_ckb, cmd_addr, dm, dqs, dq }			
[7:5]	Reserved	-				
[4:0]	DDRPHY_LV	R/W	DDRPHY_LV,please refer to table 2 {ck_ckb, cmd_addr, dm, dqs, dq }			

	DDRPHY_CTRL0[31:0]	S1[28:24]	S0[20:16]	DS[12:8]	LV[4:0]
Pad Type	DDR2 SSTL18	0x1F	0x00	0x00	0x00
	DDR SSTL2	0x1F	0x1F	0x1F	0x00
	MDDR	0x1F	0x00	0x00	0x1F
	SDR LVTTL	0x1F	0x1F	0x00	0x1F

Table 2: DDRPHY pad type

P_SCUC_DDRPHY_CTRL1 0x9200504C DDRPHY Control 1								
Bit	31	30	29	28	27	26	25	
Function	-		DDRPHY_PD					
Default	0	0	0	0	0	0	0	
	23	22	21	20	19	18	17	
	-	BOOST_ENB	LOCK_MODE1	LOCK_MODE0	DECT_MODE1	DECT_MODE0	IEN	
	0	1	0	1	1	1	0	
	15	14	13	12	11	10	9	
	DL_SEL	HCUR	VCOBUF_EN	DDR_PHY_PDN	PLL_DLL_PWDN	PHCHK_SEL	ICP_SEL	
	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	
	ICP_SEL	CLK_SYS_SEL	PLL_DLL_TEST	DLL_CLKSEL	FREQ_RANGE_SEL			
	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:29]	Reserved	-	-	
[28:24]	DDRPHY_PD	R/W	DDRPHY_PD[4:0] = {ck_ckb, cmd_addr, dm, dqs, dq } 0: Enable 1: Disable	
23	Reserved	-	-	
22	BOOST_ENB	R/W	1: Enable Boost 0: Disable Boost	
21	LOCK_MODE1	R/W	1: Enable LOCK_MODE1 0: Disable LOCK_MODE1	
20	LOCK_MODE0	R/W	1: Enable LOCK_MODE0 0: Disable LOCK_MODE0	
19	DECT_MODE1	R/W	1: Enable DECT_MODE1 0: Disable DECT_MODE1	
18	DECT_MODE0	R/W	1: Enable DECT_MODE0 0: Disable DECT_MODE0	
[17:16]	IEN	R/W	00: 20uA*4 01: 30uA*4 10: 40uA*4 11: 50uA*4	
15	DL_SEL	R/W	0: Delay Line(8 phase) 1: VCO(4 phase)	

Bit	Function	Type	Description	Condition
14	HCUR	R/W	0: Normal 1: Half	
13	VCOBUF_EN	R/W	0: Disable 1: Enable	
12	DDR_PHY_PDN	R/W	0: Normal 1: DDR PHY Digital fully clock gated	
11	PLL_DLL_PWDN	R/W	0: Normal 1: DDR PLL/DLL off	
[10:9]	PHCHK_SEL	R/W	00: HP Test1 01: HP_Test2 10:FP_Test 11:Reserved	
[8:7]	ICP_SEL	R/W	Use with P_SCUC_DDRPHY_CTRL2[12] ICP_SEL[2:0]: 000: 8uA 001: 16uA 010: 24uA 011: 48uA 100: 12uA 101: 24uA 110: 36uA 111: 72uA	
6	CLK_SYS_SEL	R/W	0: DDR PLL 1: DDR DLL	
5	PLL_DLL_TEST	R/W	1:Enable test 0:Disable test	
[4:3]	DLL_CLKSEL	R/W	00: DDR_PLL 01: Reserved 10: CLK_SYS(apr leaf) 11: TEST MODE	
[2:0]	FREQ_RANGE_SEL	R/W	000=250MHz~150MHz 001=150MHz~100MHz 100=81MHz 101=54MHz 110=27MHz	

SYS_RATIO defines the divide ratio of clk_sys and the clk_ref_ceva.

SYS_RT_RATIO is not used.

SYS_AHB_RATIO defines the divide ratio of the clk_sys_ahb and the clk_ref_ceva.

SYS_APB_RATIO defines the divide ratio of the clk_sys_apb and the clk_ref_ceva.

P_SCUC_SYS_RATIO								0x92005100								System Clock Ratio								
Bit	31	30	29	28	27	26	25	24																
Function	-																							
Default	0	0	0	0	0	0	0	0																
23	22	21	20	19	18	17	16																	
0	0	0	0	0	0	0	0																	
15	14	13	12	11	10	9	8																	
0	0	0	0	0	0	0	0																	
7	6	5	4	3	2	1	0	SYS_RATIO																
0	0	0	0	0	0	0	0																	

Bit	Function	Type	Description																Condition
[31:6]	Reserved																		
[5:0]	SYS_RATIO	R/W	The system clock ratio																

P_SCUC_SYS_RT_RATIO								0x92005104								System RT Clock Ratio								
Bit	31	30	29	28	27	26	25	24																
Function	-																							
Default	0	0	0	0	0	0	0	0																
23	22	21	20	19	18	17	16																	
0	0	0	0	0	0	0	0																	
15	14	13	12	11	10	9	8																	
0	0	0	0	0	0	0	0																	
7	6	5	4	3	2	1	0	SYS_RT_RATIO																
0	0	0	0	0	0	0	0																	

Bit	Function	Type	Description						Condition
[31:6]	Reserved								
[5:0]	SYS_RT_RATIO	R/W	The rt clock ratio for fabric register						

P_SCUC_SYS_AHB_RATIO 0x92005108 System AHB Clock Ratio									
Bit	31	30	29	28	27	26	25	24	
Function	-								
Default	0	0	0	0	0	0	0	0	
	23	22	21	20	19	18	17	16	
	-								
	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	
	-								
	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
					SYS_AHB_RATIO				
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description						Condition
[31:6]	Reserved								
[5:0]	SYS_AHB_RATIO	R/W	The system AHB clock ratio						

P_SCUC_SYS_APB_RATIO 0x9200510C System APB Clock Ratio									
Bit	31	30	29	28	27	26	25	24	
Function	-								
Default	0	0	0	0	0	0	0	0	
	23	22	21	20	19	18	17	16	
	-								
	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	
	-								
	0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
SYS_APB_RATIO							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:6]	Reserved			
[5:0]	SYS_APB_RATIO	R/W	The system APB clock ratio	

CEVA_RATIO defines the divide ratio of clk_ceva and the clk_ref_ceva.

CEVA_RT_RATIO is not used.

CEVA_AHB_RATIO defines the divide ratio of the clk_ceva_ahb and the clk_ref_ceva.

CEVA_APB_RATIO defines the divide ratio of the clk_ceva_apb and the clk_ref_ceva

P_SCUC_CEVA_RATIO 0x92005110 CEVA Clock Ration							
Bit	31	30	29	28	27	26	25
Function	-						
Default	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
CEVA_RATIO							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:6]	Reserved			
[5:0]	CEVA_RATIO	R/W	The cevaX1620 clock ratio	

P_SCUC_CEVA_AHB_RATIO 0x92005114 CEVA AHB Clock Ration							
Bit	31	30	29	28	27	26	25
Function	-						
Default	0	0	0	0	0	0	0



Bit	Function	Type	Description	Condition
[31:6]	Reserved			
[5:0]	CEVA_AHB_RATIO	R/W	The ceva ahb clock ratio	

P_SCUC_CEVA_APB_RATIO 0x92005118

CEVA APB Clock Ration

Bit	Function	Type	Description	Condition
[31:6]	Reserved			
[5:0]	CEVA_APB_RATIO	R/W	The CEVA APB clock ratio	

P_SCUC_CEVA_CNT_EN		0x9200511C								CEVA Counter Enable							
Bit	31	30	29	28	27	26	25	24									
Function	CHK_CNT_EN	-															
Default	0	0	0	0	0	0	0	0									
23	22	21	20	19	18	17	16										
0	0	0	0	0	0	0	0	-									
15	14	13	12	11	10	9	8										
0	0	0	0	0	0	0	0	-									
7	6	5	4	3	2	1	0	CEVA_APB_CNT_EN		CEVA_AHB_CNT_EN		CEVA_CNT_EN					
0	0	0	0	0	0	0	0										

Bit	Function	Type	Description								Condition
31	CHK_CNT_EN	R/W	Check Counter Enable								
[30:3]	Reserved										
2	CEVA_APB_CNT_EN	R/W	CEVA APB Counter Enable, enable ceva apb clock 1: Enable 0: Disable								
1	CEVA_AHB_CNT_EN	R/W	CEVA AHB Counter Enable, enable ceva ahb clock 1: Enable 0: Disable								
0	CEVA_CNT_EN	R/W	CEVA Counter Enable, enable cevaX1620 clock 1: Enable 0: Disable								

3.4.4 SCUD

SCUD is used for setting CEVA subsystem, for detail description, please refer to DSP section.

P_SCUD_DRAM_MAP		0x90532800								DRAM Map Base							
Bit	31	30	29	28	27	26	25	24									
Function	-	DRAM_BASE															
Default	0	0	0	0	0	0	0	0									

23	22	21	20	19	18	17	16
-							-

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
-							-

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
MAP_SIZE							-

0 0 0 1 1 1 0 0

Bit	Function	Type	Description					Condition
[31:30]	Reserved							
[29:22]	DRAM_BASE	R/W	DRAM map base					
[21:5]	Reserved							
[4:2]	MAP_SIZE	R/W	DRAM map size					
[1:0]	Reserved							

P_SCUD_SB0_RGN
0x90532804
Spare Space Control 0

Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
SB_RGN_BASE							-

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
-							-

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
SB_RGN_SIZE							SB_RGN_EN

0 0 0 0 0 0 0 0

Bit	Function	Type	Description					Condition
[31:15]	SB_RGN_BASE	R/W	SB RGN Base					

Bit	Function	Type	Description	Condition
[14:5]	Reserved			
[4:2]	SB_RGN_SIZE	R/W	SB RGN size 000 : 32KB 001 : 32KB 010 : 64KB 011 : 128KB 100 : 256KB others : 512KB	
1	Reserved			
0	SB_RGN_EN	R/W	SB RGN Enable 1: Enable 0: Disable	

P_SCUD_SB1_RGN									0x90532808									Spare Space Control 1								
Bit	31	30	29	28	27	26	25	24																		
Function																										
Default	0	0	0	0	0	0	0	0																		
SB_RGN_BASE	23	22	21	20	19	18	17	16																		
	0	0	0	0	0	0	0	0																		
SB_RGN_SIZE	15	14	13	12	11	10	9	8																		
	0	0	0	0	0	0	0	0																		
SB_RGN_EN	7	6	5	4	3	2	1	0																		
	0	0	0	0	0	0	0	0																		

Bit	Function	Type	Description												Condition
[31:15]	SB_RGN_BASE	R/W	SB RGN Base												
[14:5]	Reserved														
[4:2]	SB_RGN_SIZE	R/W	SB RGN size 000 : 32KB 001 : 32KB 010 : 64KB 011 : 128KB 100 : 256KB others : 512KB												

Bit	Function	Type	Description					Condition
1	Reserved							
0	SB_RGN_EN	R/W	SB RGN Enable 1: Enable 0: Disable					

P_SCUD_PMIM_ST
0x9053280C
PMIM Status

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	P_MAP_ERR	P_RESP_ERR
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:2]	Reserved							
1	P_MAP_ERR	R/W	PMIM map error					
0	P_RESP_ERR	R/W	PMIM resp error					

P_SCUD_DMIM_ST
0x90532810
DMIM Status

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
-						D_MAP_ERR	D_RESP_ERR

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description	Condition
[31:2]	Reserved			
1	D_MAP_ERR	R/W	DMIM map error	
0	D_RESP_ERR	R/W	DMIM resp error	

3.5 CLOCK

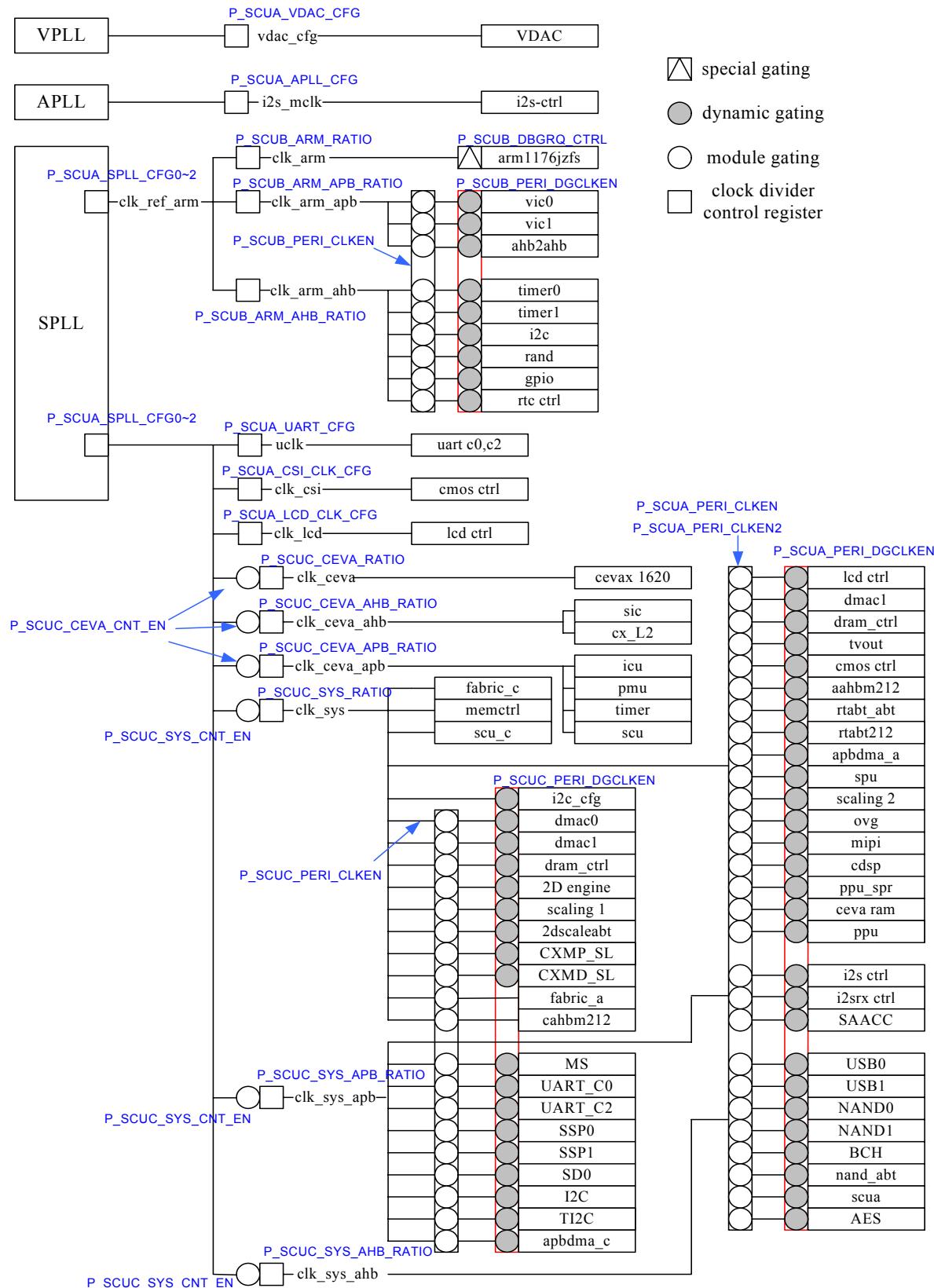


Fig. 1 Clocks and Control Registers in GPL32900A

3.6 SPLL

SPLL is the clock source for almost the whole system. It generates the reference clock, clk_ref_arm and clk_ref_ceva, to the ARM subsystem, CEVA subsystem, and the SYS subsystem. There are 2 SPLL modules in GPL32900A and it's named by SPLL and SPLL2. The SPLL is the clock source for ARM and CEVA subsystem and the SPLL2 is just for ARM subsystem. In other word, ARM subsystem can choose SPLL or SPLL2 either from a clock source. The output frequency F_{spll} of the SPLL is programmable, configured by P_SCUB_SPLL_CFG0, P_SCUB_SPLL_CFG1 and P_SCUB_SPLL_CFG2.

3.6.1 SPLL Clock Output

The clock output formula is

$$f_{spll} = \frac{CLKIN}{M} * N * R * [(1 - MR\%) \sim (1 + MR\%)]$$

- (1) CLKIN: 27MHz
- (2) M[4:0]: CLKIN divider
M= 1~31, default = 27
- (3) N[7:0]: PLL feedback divider
N = 32~212
- (4) R [1:0]: PLL post-divider
R [1:0] = (0, x) \rightarrow R = 8
R [1:0] = (1, x) \rightarrow R = 4 (default)
- (5) MR: modulation rate, only valid in spread spectrum mode

Example:

R [1:0]=(1, 1) \rightarrow R = 4
M [8:0] = (0, 0, 0, 0, 1, 1, 0, 1) \rightarrow M = 27
CLKIN=27MHz,
SPLL CLKOUT=512MHz, if N [8:0] = (0, 1, 0, 0, 0, 0, 0, 0)
SPLL CLKOUT=768MHz, if N [8:0] = (0, 1, 1, 0, 0, 0, 0, 0)
SPLL CLKOUT=848MHz, if N [8:0] = (0, 1, 1, 0, 1, 0, 1, 0)

3.6.2 SPLL Spread Spectrum Mode

Only useful in P_SCUB_SPLL_CFG1.SPLL_SSCGON = 1 or P_SCUB_SPLL_CFG1.SPLL2_SSCGON = 1, otherwise they are useless.

- (1) MDS [6:4]: Modulation Clock Control, two modulate clock will appear in spread spectrum mode.

FPFD = CLKIN/M

MODCLK=FPFD/(MDS[6:4]*2+4)/2

MODCLK=FPFD/((MDS[6:4]*2-1)+4)/2

If FPFD=27MHz/27=1MHz

(0, 0, 0)=FPFD/8=125KHz or FPFD/6=166.67kHz

(2) MDS [3:0] :Modulation Rate Control

Modulation Rate, MR= $\delta N/N * 100\%$

If N=100

(0, 0, 0, 0), $\delta N=1$, MR=1%, UP Spread
(0, 0, 0, 1), $\delta N=2$, MR=2%, Center Spread
(0, 0, 1, 0), $\delta N=4$, MR=4%, Center Spread
(0, 0, 1, 1), $\delta N=6$, MR=6%, Center Spread
(0, 1, 0, 0), $\delta N=8$, MR=8%, Center Spread
(0, 1, 0, 1), $\delta N=10$, MR=10%, Center Spread
(0, 1, 1, 0), $\delta N=12$, MR=12%, Center Spread
(0, 1, 1, 1), $\delta N=14$, MR=14%, Center Spread
(1, 0, 0, 0), $\delta N=16$, MR=16%, Center Spread
(1, 0, 0, 1), $\delta N=18$, MR=18%, Center Spread
(1, 0, 1, 0), $\delta N=20$, MR=20%, Center Spread
(1, 0, 1, 1), $\delta N=22$, MR=22%, Center Spread
(1, 1, 0, 0), $\delta N=24$, MR=24%, Center Spread
(1, 1, 0, 1), $\delta N=26$, MR=26%, Center Spread
(1, 1, 1, 0), $\delta N=28$, MR=28%, Center Spread
(1, 1, 1, 1), $\delta N=30$, MR=30%, Center Spread

Example:

M=27, N=212, R [1:0] = (1, 1), MDS [7:0] = (0, 0, 1, 1, 0, 0, 1, 0)

MR=1.89%, MODCLK = 50KHz

CLKOUT=832~864MHz

3.6.3 SPLL Modification Procedure

The procedure to modify the SPLL is :

Step1. Ensure that all the code executed by the spll modification procedure is in internal memory, including system ram or ceva ram.

Step2.Switch the clock to 27MHz by setting ARM_CLKSEL = 0 and CEVA_CLKSEL = 0.

Step3.Set SPLL_PD=1 to power down the PLL.

Step4.Set the valid setttings

Step5.Set SPLL_PD=0 to power up the PLL

Step6. At least wait 500us.

Step7.Switch back to the target frequency by settting ARM_CLKSEL and CEVA_CLKSEL.

3.7 ARM Subsystem

There are 3 clock dormains in the ARM subsystem, clk_arm, clk_arm_ahb, and clk_arm_apb, which are generated based on the clk_ref_arm. clk_ref_arm is generated from SPLL, and the configuration register locates at P_SCUB_SPLL_CFG0. The frequency formula of clk_ref_arm is :

$$\text{if } \text{ARM_CLKSEL}[2:0] = \begin{cases} 0: f_{\text{clk_ref_arm}} = 27MHz \\ 1: f_{\text{clk_ref_arm}} = 32.768KHz \\ 2 \text{ or } 3: f_{\text{clk_ref_arm}} = \frac{f_{\text{spl}}}{(\text{ARM_CLKSEL}[2:0])} \\ 4 \sim 7: f_{\text{clk_ref_arm}} = f_{\text{spl}} \end{cases}$$

- ARM1176JZFS :

The ARM1176JZFS is controlled by clk_arm, and the configuration register located at P_SCUB_ARM_RATIO. The frequency formula is:

$$f_{\text{clk_arm}} = \frac{f_{\text{clk_ref_arm}}}{(\text{ARM_RATIO}[5:0]+1)}$$

When ARM enter WFI(wait for interrupt) mode, the clk_arm is gated. After ARM leaving WFI mode, the clock is enabled automatically. The detail WFI function of ARM1176JZFS could be refered in ARM related manuals.

There are three methods to change the frequency of clk_arm. The first is changing ARM_CLKSEL, the second is changing ARM_RATIO, and the last is changing fspll. Only the last one need to change the setting of SPLL, which effects to the all subsystems related to SPLL. ARM_CLKSEL can be modified directly.

The procedure to modify the ARM_RATIO is :

Step1. Set the ARM_RATIO in P_SCUB_ARM_RATIO

Step2. trigger the update flags of the modified ratio in P_SCUB_UPDATE_RATIO.

- ARM Subsystem (ARM AHB/APB)

There are two clock domain in ARM subsystem, clk_arm_ahb, and clk_arm_apb. Two kinds of clock gating are used in each modules, “module gating” and “dynamic gating”. “module gating” are enable/disable by P_SCUB_PERI_CLKEN, and “dynamic gating” are enable/disable by P_SCUB_PERI_DGCLKEN. If the related bit of the dedicated module in P_SCUB_PERI_CLKEN is 0, the clock of that module is gated, and the module enters the sleep mode. The register value of the module will not volatilize during sleep mode. If the bit P_SCUB_PERI_CLKEN is 1, and P_SCUB_PERI_DGCLKEN is 1, the dynamic gating is enabled. The dynamic gating can save the power consumption when the module is working. If P_SCUB_PERI_DGCLKEN is 0, the dynamic gating is disable, and the power consumption is higher. The related control bit of the modules defined in Table 1. At Table 1 “G” means “clock gating”, “D” means “dynamic clock gating”.

The frequency formula is:

$$\begin{cases} f_{clk_arm_ahb} = \frac{f_{clk_arm}}{(ARM_AHB_RATIO[5:0]+1)} \\ f_{clk_arm_apb} = \frac{f_{clk_arm_ahb}}{(ARM_APB_RATIO[5:0]+1)} \end{cases}$$

There are three methods to change the frequency of clk_arm_ahb or clk_arm_apb. The first is changing ARM_CLKSEL, the second is changing ARM_AHB_RATIO or ARM_APB_RATIO, and the last is changing fspll. Only the last one need to change the setting of SPLL, which effects to the all subsystems related to SPLL. ARM_CLKSEL can be modified directly.

The procedure to modify the ARM_AHB_RATIO or ARM_APB_RATIO is :

Step1. Set the ARM_AHB_RATIO in P_SCUB_ARM_AHB_RATIO, or ARM_APB_RATIO in P_SCUB_ARM_APB_RATIO

Step2. Trigger the update flags of the modified ratio in P_SCUB_UPDATE_RATIO.

3.8 CEVA Subsystem

There are 3 clock domains in the CEVA subsystem, clk_ceva, clk_ceva_ahb, and clk_ceva_apb, which are generated based on the clk_ref_ceva. clk_ref_ceva is generated from SPLL, and the configuration register locates at P_SCUB_SPLL_CFG0. The frequency formula of clk_ref_ceva is :

$$\text{if } \text{CEVA_CLKSEL}[2:0] = \begin{cases} 0: f_{clk_ref_ceva} = 27MHz \\ 1: f_{clk_ref_ceva} = 32.768KHz \\ 2 \text{ or } 3: f_{clk_ref_ceva} = \frac{f_{spll}}{(\text{CEVA_CLKSEL}[2:0])} \\ 4 \sim 7: f_{clk_ref_ceva} = f_{spll} \end{cases}$$

Unlike ARM subsystem, there is no “clock gating” or “dynamic gating” per module in CEVA subsystem. There is a gloable “clock gating” for each clock domain. The global clock gating is controlled in P_SCUC_CEVA_CNT_EN. If the assigned bit in P_SCUC_CEVA_CNT_EN is 0, the clock domain is disabled. For example, if the bit[1] in P_SCUC_CEVA_CNT_EN, clk_ceva_ahb is disabled, and the related modules are not work.

The frequency formula is:

$$\begin{cases} f_{clk_ceva} = \frac{f_{clk_ref_ceva}}{(CEVA_RATIO[5:0]+1)} \\ f_{clk_ceva_ahb} = \frac{f_{clk_ceva}}{(CEVA_AHB_RATIO[5:0]+1)} \\ f_{clk_ceva_apb} = \frac{f_{clk_ceva_ahb}}{(CEVA_APB_RATIO[5:0]+1)} \end{cases}$$

There are three methods to change the frequency of CEVA subsystem. The first is changing CEVA_CLKSEL, the second is changing the RATIO of CEVA, CEVA_AHB or CEVA_APB, and the last is changing fspll. Only the last one need to change the setting of SPLL, which effects to the all subsystems related to SPLL. CEVA_CLKSEL can be modified directly.

The procedure to modify the RATIO is :

- Step1. Set the CEVA_RATIO in P_SCUC_CEVA_RATIO, CEVA_AHB_RATIO in P_SCUC_CEVA_AHB_RATIO, or CEVA_APB_RATIO in P_SCUC_CEVA_APB_RATIO.
- Step2. Read the “Updated Check Flag (UCF)” in P_SCUC_SYS_RATIO_UPDATE
- Step3. Trigger the update flags of the modified ratio in P_SCUC_SYS_RATIO_UPDATE.
- Step4. Pooling the UCF until the value is change

The clock RATIOS for ceva subsystem can be modified and updated anytime; even through the related ceva clocks is disabled. The clock for SCUC (clk_sys) should be turned on, in order to set and update the RATIOS.

3.9 SYS Subsystem

There are 3 clock dormains in the System subsystem, clk_sys, clk_sys_ahb, and clk_sys_apb, which are generated based on the clk_ref_ceva. clk_ref_ceva is generated from SPLL, which could be referred in the previous section. The related modules of these clocks were shown in Figure 1. Three kinds of clock gating are used in each modules, “domain gating”, “module gating”, and “dynamic gating”.

“domain gating” are enable/disable by P_SCUB_SYS_CNT_EN. If the assigned bit in P_SCUB_SYS_CNT_EN is 0, the clock domain is disabled. For example, if the bit[2] in P_SCUB_SYS_CNT_EN, clk_sys_ahb is disabled, and the related modules are not work. (ps. SYS RT domain is not implement.)

“module gating” are enable/disable by P_SCUC_PERI_CLKEN and P_SCUA_PERI_CLKEN. If the related bit of the dedicated module in P_SCUC_PERI_CLKEN or P_SCUA_PERI_CLKEN is 0, the clock of that module is gated, and the module enters the sleep mode. The register value of the module will not volatilize during sleep mode. The related control bit of the modules defined in Table 1, “G” means “clock gating”.

“dynamic gating” are enable/disable by P_SCUC_PERI_DGCLKEN and P_SCUA_PERI_DGCLKEN. If the related bit of the dedicated module in P_SCUC_PERI_DGCLKEN or P_SCUA_PERI_DGCLKEN is 1, the dynamic gating of the assigned module is enabled. The dynamic gating can save the power consumption when the module is working. If control bit is 0, the dynamic gating is disable, and the power consumption is higher. The related control bit of the modules defined in Table 1. At Table 1 “D” means “dynamic clock gating”.

The frequency formula is:

$$\begin{cases} f_{clk_sys} = \frac{f_{clk_ref_ceva}}{(SYS_RATIO[5:0]+1)} \\ f_{clk_sys_ahb} = \frac{f_{clk_sys}}{(SYS_AHB_RATIO[5:0]+1)} \\ f_{clk_sys_apb} = \frac{f_{clk_sys_ahb}}{(SYS_APB_RATIO[5:0]+1)} \end{cases}$$

There are three methods to change the frequency of SYS subsystem. The first is changing CEVA_CLKSEL, the second is changing the RATIO of SYS, SYS_AHB or SYS_APB, and the last is changing fspll. Only the last one need to change the setting of SPLL, which effects to the all subsystems related to SPLL. CEVA_CLKSEL can be modified directly.

The procedure to modify the RATIO is :

- Step1. Set the SYS_RATIO in P_SCUC_SYS_RATIO, SYS_AHB_RATIO in P_SCUC_SYS_AHB_RATIO, or SYS_APB_RATIO in P_SCUC_SYS_APB_RATIO.
- Step2. Read the “Updated Check Flag (UCF)” in P_SCUC_SYS_RATIO_UPDATE
- Step3. Trigger the update flags of the modified ratio in P_SCUC_SYS_RATIO_UPDATE.
- Step4. Pooling the UCF until the value is change

The clock RATIOS for SYS subsystem can be modified and updated anytime, but the clk_sys should be turned on first. “clk_sys” can be turned on by P_SCUB_SYS_CNT_EN.

3.10 Clock Configuration

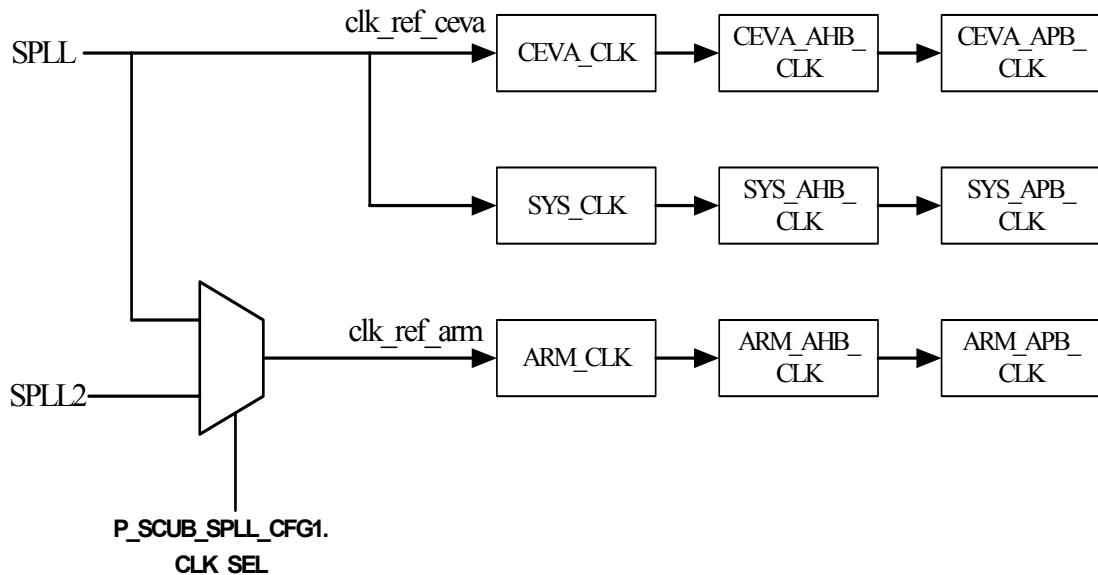


Fig. 2 Clock tree in GPL32900A

Please refer following ratio to set clocks:

(1) ceva_clk:ceva_ahb_clk:ceva_apb_clk = 4:2:1 or 2:2:1, for example:

ceva_clk = 160MHz

ceva_ahb_clk = 160MHz

ceva_apb_clk = 80MHz

(2) sys_clk:sys_ahb_clk:sys_apb_clk = 2:2:1, for example:

sys_clk = 160MHz

sys_ahb_clk = 160MHz

sys_apb_clk = 80MHz

(3) arm_clk:arm_ahb_clk:arm_apb_clk = 4:2:1, for example:

arm_clk = 400MHz

arm_ahb_clk = 200MHz

arm_apb_clk = 100MHz

How to set control registers about above clock, please refer to following settings:

(1) P_SCUB_ARM_RATIO = 0, P_SCUB_ARM_AHB_RATIO = 1, P_SCUB_ARM_APB_RATIO = 1

(2) P_SCUC_CEVA_RATIO = 1, P_SCUC_CEVA_AHB_RATIO = 0, P_SCUC_CEVA_APB_RATIO = 1

(3) P_SCUC_SYS_RATIO = 1, P_SCUC_SYS_AHB_RATIO = 0, P_SCUC_SYS_APB_RATIO = 1

(4) P_SCUB_SPLL_CFG0[1:0] = 3, P_SCUB_SPLL_CFG0[9:2] = 160, P_SCUB_SPLL_CFG0[11:10] = 3

(5) P_SCUB_SPLL_CFG1[7:0] = 200, P_SCUB_SPLL_CFG1[24] = 1.

(6) P_SCUB_SPLL_CFG0[14:12] = 2, P_SCUB_SPLL_CFG0[18:16] = 2

4 Digital Signal Processor (DSP)

4.1 Introduction

CEVA-XS is a fully integrated, general-purpos DSP platform that serves a variety of DSP applications. It is equipped with a selected cluster of advanced peripheral and glueless interfaces. The CEVA-XS includes the powerful, fixed-point CEVA-X1620 DSP, which provides high performance and supports low-power modes.

The backbone of the CEVA-XS data traffic is a multi-layered AHB-Lite interconnect matrix. The main masters over this interconnect matrix are the CEVA-X1620 data ports, a powerful DMA module and a slave bridge that allows external potential masters to access the CEVA-XS peripheral. CEVA-XS master may access an external AHB system bus via a dedicated master AHB bridge.

4.2 Features

The system is equipped with a selected cluster of advanced peripherals and glueless interfaces. The following is a list of CEVA modules:

- High performance Direct Memory Access (XDMA) controller
- XAHB Interconnect
- A configurable number of Buffered Time Division Multiplexing Ports (BTDMRx/Tx)
- Power Management Unit (PMU) with automatic turn-off of peripherals when inactive
- Glueless interface with two external memory arrays
- Interrupt Control Unit (ICU) with vector interrupt support
- Asynchronous AHB master bridge providing access to an external AHB bus
- Asynchronous AHB slave bridge providing an external AHB bus with access to CEVA-XS memory space
- A hook for attaching additional AHB accelerators
- XAPB bus bridge (APBB)
- Code Replacement Unit (CRU)
- Two timer units (TIMER0 and TIMER1)
- Processor Interface Unit (PIU)
- General-purpose I/Os (GPIOs)

5 Vector Interrupt Controller (VIC)

5.1 Introduction

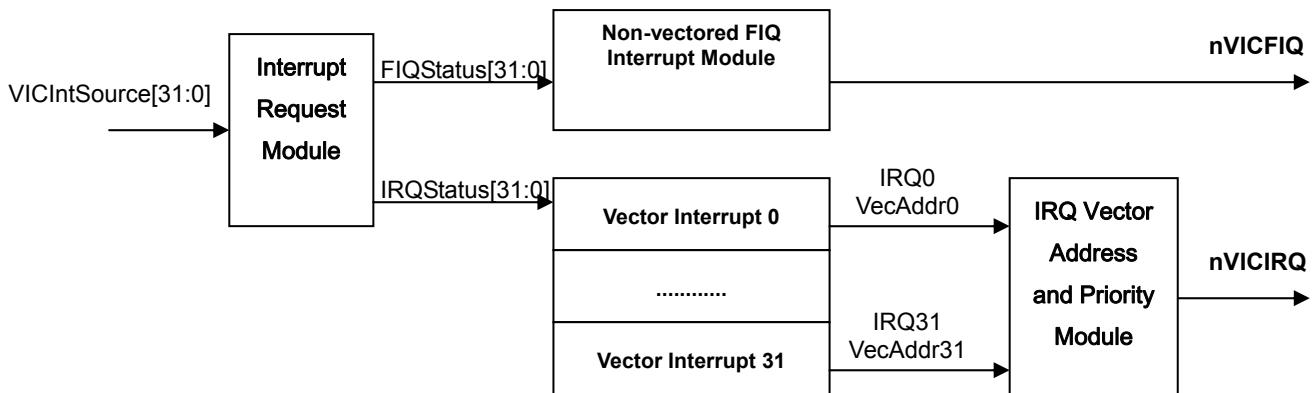
In an ARM processor based system, two levels of interrupt are available: the FIQ for fast, low latency interrupt handling and the IRQ for general interrupts. The VIC provides a software interface to the interrupt system. It supports the interrupt number of the service routine corresponding to the priority requesting interrupt source by level sensitive input. The interrupt source active HIGH and held asserted until the interrupt is cleared in the peripheral by the interrupt service routine.

5.2 Features

- There are 32 interrupts available for both controller (VIC0, VIC1).
- Both of programmed priority level and the fixed hardware priority levels are provide.
- VIC0 has the highest hardware priority level.
- The software can control each interrupt request line to generate a software interrupt.
- Provides interrupt priority masking

5.3 Block Diagram

The following diagram is a functional block diagram of the VIC module.



5.4 Interrupt Source of VIC0

Source Number	Source	Module	Default Priority
0	sdmac_int0[0]	SDMA	15
1	sdmac_int0[1]	SDMA	15
2	xdmac_int1[0]	XDMA CH0	15
3	xdmac_int1[1]	XDMA CH1	15
4	xdmac_int1[2]	XDMA CH2	15

Source Number	Source	Module	Default Priority
5	xdmac_int1[3]	XDMA CH3	15
6	Reserved	Reserved	-
7	t0_intr[0]	TIMER_B	15
8	t0_intr[1]	TIMER_B	15
9	t0_intr[4:2]	TIMER_B	15
10	t1_intr[0]	WDT	15
11	t1_intr[1]	WDT	15
12	t1_intr[4:2]	WDT	15
13	i2s_tx	I2S TX	15
14	i2s_rx	I2S RX	15
15	cdsp	CDSP	15
16	usb1	USB1 (Device)	15
17	ehci	USB0 (HOST)	15
18	ohci	USB0 (HOST)	15
19	csi	SENSOR	15
20	ppu	PPU	15
21	tvout	TV	15
22	scaler	SCALER	15
23	lcd	LCD	15
24	a_apbdma[0]	APBDMA	15
25	a_apbdma[1]	APBDMA	15
26	a_apbdma[2]	APBDMA	15
27	a_apbdma[3]	APBDMA	15
28	cx_sem_hint[0]	CEVA PIU	15
29	cx_sem_hint[1]	CEVA PIU	15
30	cx_sem_hint[2]	CEVA PIU	15
31	cx_cr_hint	CEVA PIU	15

5.5 Interrupt Source of VIC1

Source Number	Source	Module	Default Priority
0	gpio0	GPIO0	15
1	gpio1	GPIO1	15
2	pwrc_lvrz	PWRC	15
3	i2c_b0	I2C_B	15
4	i2c_c0	I2C_C	15
5	rtc	RTC	15
6	poweron	PWRC	15
7	int_xdma_bp_r	XDMA	15

Source Number	Source	Module	Default Priority
8	sd0	SD0	15
9	sd1	SD1	15
10	mipi	MIPI	15
11	bch	BCH	15
12	ovg	OVG	15
13	gpio2	GPIO2	15
14	gpio3	GPIO3	15
15	Reserved	Reserved	-
16	cir	REMOTE CONTROLLER	15
17	uart_b0	UART_B0	15
18	nand0	NAND Flash	15
19	spu	SPU	15
20	ssp0	SSP/SPI0	15
21	ssp1	SSP/SPI1	15
22	ms	MEMORY STICK CONTROLLER	15
23	i2c_c1	Turbo I2C	15
24	uart_c0	UARTC0	15
25	aes	AES	15
26	uart_c2	UARTC2	15
27	saacc	SAR ADC CTRL	15
28	c_apbdma[0]	APBDMAC	15
29	c_apbdma[1]	APBDMAC	15
30	c_apbdma[2]	APBDMAC	15
31	c_apbdma[3]	APBDMAC	15

5.6 Register Summary

Name	Address	Description
P_VICx_IRQSTATUS	0x90010000 / 0x90020000	VIC0/1 IRQ Status Register
P_VICx_FIQSTATUS	0x90010004 / 0x90020004	VIC0/1 FIQ Status Register
P_VICx_IRQRRAWSTATUS	0x90010008 / 0x90020008	VIC0/1 Raw Interrupt Status Register
P_VICx_INTSELECT	0x9001000C / 0x9002000C	VIC0/1 Interrupt Select Register
P_VICx_INTENABLE	0x90010010 / 0x90020010	VIC0/1 Interrupt Enable Register
P_VICx_INTENCLEAR	0x90010014 / 0x90020014	VIC0/1 Interrupt Enable Clear Register
P_VICx_SOFTINT	0x90010018 / 0x90020018	VIC0/1 Software Interrupt Register
P_VICx_SOFTINTCLEAR	0x9001001C / 0x9002001C	VIC0/1 Software Interrupt Clear Register
P_VICx_PROTECTION	0x90010020 / 0x90020020	VIC0/1 Protection Enable Register
P_VICx_PRIORITYMASK	0x90010024 / 0x90020024	VIC0/1 Software Priority Mask Register

Name	Address	Description
P_VICx_VECTADDR0	0x90010100 / 0x90020100	VIC0/1 Interrupt Number 0
P_VICx_VECTADDR1	0x90010104 / 0x90020104	VIC0/1 Interrupt Number 1
P_VICx_VECTADDR2	0x90010108 / 0x90020108	VIC0/1 Interrupt Number 2
P_VICx_VECTADDR3	0x9001010C / 0x9002010C	VIC0/1 Interrupt Number 3
P_VICx_VECTADDR4	0x90010110 / 0x90020110	VIC0/1 Interrupt Number 4
P_VICx_VECTADDR5	0x90010114 / 0x90020114	VIC0/1 Interrupt Number 5
P_VICx_VECTADDR6	0x90010118 / 0x90020118	VIC0/1 Interrupt Number 6
P_VICx_VECTADDR7	0x9001011C / 0x9002011C	VIC0/1 Interrupt Number 7
P_VICx_VECTADDR8	0x90010120 / 0x90020120	VIC0/1 Interrupt Number 8
P_VICx_VECTADDR9	0x90010124 / 0x90020124	VIC0/1 Interrupt Number 9
P_VICx_VECTADDR10	0x90010128 / 0x90020128	VIC0/1 Interrupt Number 10
P_VICx_VECTADDR11	0x9001012C / 0x9002012C	VIC0/1 Interrupt Number 11
P_VICx_VECTADDR12	0x90010130 / 0x90020130	VIC0/1 Interrupt Number 12
P_VICx_VECTADDR13	0x90010134 / 0x90020134	VIC0/1 Interrupt Number 13
P_VICx_VECTADDR14	0x90010138 / 0x90020138	VIC0/1 Interrupt Number 14
P_VICx_VECTADDR15	0x9001013C / 0x9002013C	VIC0/1 Interrupt Number 15
P_VICx_VECTADDR16	0x90010140 / 0x90020140	VIC0/1 Interrupt Number 16
P_VICx_VECTADDR17	0x90010144 / 0x90020144	VIC0/1 Interrupt Number 17
P_VICx_VECTADDR18	0x90010148 / 0x90020148	VIC0/1 Interrupt Number 18
P_VICx_VECTADDR19	0x9001014C / 0x9002014C	VIC0/1 Interrupt Number 19
P_VICx_VECTADDR20	0x90010150 / 0x90020150	VIC0/1 Interrupt Number 20
P_VICx_VECTADDR21	0x90010154 / 0x90020154	VIC0/1 Interrupt Number 21
P_VICx_VECTADDR22	0x90010158 / 0x90020158	VIC0/1 Interrupt Number 22
P_VICx_VECTADDR23	0x9001015C / 0x9002015C	VIC0/1 Interrupt Number 23
P_VICx_VECTADDR24	0x90010160 / 0x90020160	VIC0/1 Interrupt Number 24
P_VICx_VECTADDR25	0x90010164 / 0x90020164	VIC0/1 Interrupt Number 25
P_VICx_VECTADDR26	0x90010168 / 0x90020168	VIC0/1 Interrupt Number 26
P_VICx_VECTADDR27	0x9001016C / 0x9002016C	VIC0/1 Interrupt Number 27
P_VICx_VECTADDR28	0x90010170 / 0x90020170	VIC0/1 Interrupt Number 28
P_VICx_VECTADDR29	0x90010174 / 0x90020174	VIC0/1 Interrupt Number 29
P_VICx_VECTADDR30	0x90010178 / 0x90020178	VIC0/1 Interrupt Number 30
P_VICx_VECTADDR31	0x9001017C / 0x9002017C	VIC0/1 Interrupt Number 31
P_VICx_PRIORITY0	0x90010200 / 0x90020200	VIC0/1 Priority 0 Register
P_VICx_PRIORITY1	0x90010204 / 0x90020204	VIC0/1 Priority 1 Register
P_VICx_PRIORITY2	0x90010208 / 0x90020208	VIC0/1 Priority 2 Register
P_VICx_PRIORITY3	0x9001020C / 0x9002020C	VIC0/1 Priority 3 Register
P_VICx_PRIORITY4	0x90010210 / 0x90020210	VIC0/1 Priority 4 Register
P_VICx_PRIORITY5	0x90010214 / 0x90020214	VIC0/1 Priority 5 Register
P_VICx_PRIORITY6	0x90010218 / 0x90020218	VIC0/1 Priority 6 Register
P_VICx_PRIORITY7	0x9001021C / 0x9002021C	VIC0/1 Priority 7 Register
P_VICx_PRIORITY8	0x90010220 / 0x90020220	VIC0/1 Priority 8 Register

Name	Address	Description
P_VICx_PRIORITY9	0x90010224 / 0x90020224	VIC0/1 Priority 9 Register
P_VICx_PRIORITY10	0x90010228 / 0x90020228	VIC0/1 Priority 10 Register
P_VICx_PRIORITY11	0x9001022C / 0x9002022C	VIC0/1 Priority 11 Register
P_VICx_PRIORITY12	0x90010230 / 0x90020230	VIC0/1 Priority 12 Register
P_VICx_PRIORITY13	0x90010234 / 0x90020234	VIC0/1 Priority 13 Register
P_VICx_PRIORITY14	0x90010238 / 0x90020238	VIC0/1 Priority 14 Register
P_VICx_PRIORITY15	0x9001023C / 0x9002023C	VIC0/1 Priority 15 Register
P_VICx_PRIORITY16	0x90010240 / 0x90020240	VIC0/1 Priority 16 Register
P_VICx_PRIORITY17	0x90010244 / 0x90020244	VIC0/1 Priority 17 Register
P_VICx_PRIORITY18	0x90010248 / 0x90020248	VIC0/1 Priority 18 Register
P_VICx_PRIORITY19	0x9001024C / 0x9002024C	VIC0/1 Priority 19 Register
P_VICx_PRIORITY20	0x90010250 / 0x90020250	VIC0/1 Priority 20 Register
P_VICx_PRIORITY21	0x90010254 / 0x90020254	VIC0/1 Priority 21 Register
P_VICx_PRIORITY22	0x90010258 / 0x90020258	VIC0/1 Priority 22 Register
P_VICx_PRIORITY23	0x9001025C / 0x9002025C	VIC0/1 Priority 23 Register
P_VICx_PRIORITY24	0x90010260 / 0x90020260	VIC0/1 Priority 24 Register
P_VICx_PRIORITY25	0x90010264 / 0x90020264	VIC0/1 Priority 25 Register
P_VICx_PRIORITY26	0x90010268 / 0x90020268	VIC0/1 Priority 26 Register
P_VICx_PRIORITY27	0x9001026C / 0x9002026C	VIC0/1 Priority 27 Register
P_VICx_PRIORITY28	0x90010270 / 0x90020270	VIC0/1 Priority 28 Register
P_VICx_PRIORITY29	0x90010274 / 0x90020274	VIC0/1 Priority 29 Register
P_VICx_PRIORITY30	0x90010278 / 0x90020278	VIC0/1 Priority 30 Register
P_VICx_PRIORITY31	0x9001027C / 0x9002027C	VIC0/1 Priority 31 Register
P_VICx_ADDRESS	0x90010F00 / 0x90020F00	VIC0/1 Address Register

5.7 Register Definition

P_VICx_IRQSTATUS 0x90010000 / 0x90020000 VIC0/1 IRQ Status Register								
Bit	31	30	29	28	27	26	25	24
Function								
Default	0	0	0	0	0	0	0	0
IRQFLAG								
23	22	21	20	19	18	17	16	
IRQFLAG								
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
IRQFLAG								
0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
IRQFLAG							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	IRQFLAG	R	The P_VICx_IRQSTATUS register shows the status of the IRQ after masking by the P_VICx_INTENABLE and P_VICx_INTSELECT registers. Each IRQ flag corresponds to an interrupt source individually.					0 = interrupt is inactive (reset) 1 = interrupt is active

P_VICx_FIQSTATUS 0x90010004 / 0x90020004 VIC0/1 FIQ Status Register

Bit	31	30	29	28	27	26	25	24
FIQFLAG								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
FIQFLAG							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
FIQFLAG							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
FIQFLAG							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	FIQFLAG	R	The P_VICx_FIQSTATUS register shows the status of the FIQ after masking by the P_VICx_INTENABLE and P_VICx_INTSELECT registers. Each FIQ flag corresponds to an interrupt source individually.					0 = interrupt is inactive (reset) 1 = interrupt is active

P_VICx_IRQRAWSTATUS 0x90010008 / 0x90020008 VIC0/1 Raw Interrupt Status Register

Bit	31	30	29	28	27	26	25	24
RAWIRQFLAG								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
RAWIRQFLAG							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
RAWIRQFLAG							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
RAWIRQFLAG							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:0]	RAWIRQFLAG	R	The P_VICx_IRQRAWSTATUS register shows the status of the interrupt before masking by the P_VICx_INTENABLE register. Each RAWIRQFLAG flag corresponds to an interrupt source individually.					0 = interrupt is inactive before masking 1 = interrupt is active before masking

P_VICx_INTSELECT 0x9001000C / 0x9002000C VIC0/1 Interrupt Select Register

Bit	31	30	29	28	27	26	25	24
VICTYPE								

Default	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---

23	22	21	20	19	18	17	16
VICTYPE							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
VICTYPE							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
VICTYPE							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:0]	VICTYPE	R/W	The P_VICx_INTSELECT register selects type of interrupt for request. Each VICTYPE corresponds to an interrupt source individually.					0 = IRQ interrupt (default) 1 = FIQ interrupt

P_VICx_INTENABLE 0x90010010 / 0x90020010 VIC0/1 Interrupt Enable Register

Bit	31	30	29	28	27	26	25	24
Function	INTEN							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:0]	INTEN	R/W	The P_VICx_INTENABLE register enables interrupt request lines, which allow the interrupts to reach the processor. Each INTEN corresponds to an interrupt source individually.	Read : 0 = disable interrupt (default) 1 = enable interrupt Write : 0 = no effect 1 = enable interrupt

P_VICx_INTENCLEAR 0x90010014 / 0x90020014 VIC0/1 Interrupt Enable Clear Register

Bit	31	30	29	28	27	26	25	24
Function	INTCLR							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
INTCLR							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	INTCLR	W	The P_VICx_INTENCLEAR register clears corresponding bits in the P_VICx_INTENABLE register. Each INTCLR corresponds to an interrupt source individually.	0 = no effect 1 = disable interrupt in P_VICx_INTENABLE register

P_VICx_SOFTINT **0x90010018 / 0x90020018** **VIC0/1** **Software** **Interrupt**
Register

Bit	31	30	29	28	27	26	25	24
SOFTINT								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
SOFTINT							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SOFTINT							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SOFTINT							
0	0	0	0	0	0	0	0



Bit	Function	Type	Description	Condition
[31:0]	SOFTINT	R/W	The P_VICx_SOFTINT register generates a software interrupt for the selected source before interrupt masking. Each SOFTINT corresponds to an interrupt source individually.	Read: 0 = disable software interrupt (default) 1 = enable software interrupt Write: 0 = no effect 1 = enable software interrupt

P_VICx_SOFTINTCLEAR 0x9001001C / 0x9002001C VIC0/1 Software Interrupt Clear Register

23	22	21	20	19	18	17	16
SOFTINTCLR							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SOFTINTCLR							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SOFTINTCLR							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	SOFTINTCLR	W	The P_VICx_SOFTINTCLEAR register clears corresponding bits in the P_VICx_SOFTINT register. Each SOFTINTCLR corresponds to an interrupt source individually.	0 = no effect 1 = disable interrupt in P_VICx_SOFTINT register

P_VICx_PROTECTION	0x90010020 / 0x90020020	VIC0/1	Protection	Enable
			Register	

23	22	21	20	19	18	17	16
PROTEC							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
PROTEC							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PROTEC							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	PROTEC	R/W	The P_VICx_PROTECTION register enables or disables protected register access. When disabled, both user mode and privileged mode can access the registers. This register can only be accessed in privileged mode, even when protection mode is disabled. Each PROTEC corresponds to an interrupt source individually.	0 = disable protection mode (reset) 1 = enable protection mode

P_VICx_PRIORITYMASK 0x90010024 / 0x90020024 VIC0/1 Software Priority Mask Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
PRIMASK							
1	1	1	1	1	1	1	1

7	6	5	4	3	2	1	0
PRIMASK							
1	1	1	1	1	1	1	1

Bit	Function	Type	Description					Condition
[15:0]	PRIMASK	R/W	The P_VICx_PRIORITYMASK register controls software masking of the 16 interrupt priority levels.					0 = interrupt priority level is masked 1 = interrupt priority level is not masked

P_VICx_VECTADDR0 **0x90010100 / 0x90020100** **VIC0/1 Interrupt Number 0**

~ ~ ~

P_VICx_VECTADDR31 **0x9001017C / 0x9002017C** **VIC0/1 Interrupt Number 31**

Bit	31	30	29	28	27	26	25	24
INTNUM								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16	
INTNUM								
0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	
INTNUM								
0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0	
INTNUM								
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	INTNUM	R	The P_VICx_VECTADDR0 ~ P_VICx_VECTADDR31 contain ISR numbers					P_VIC0_VECTADDR0 ~ P_VIC0_VECTADDR31 : 0 ~ 31 P_VIC1_VECTADDR0 ~ P_VIC1_VECTADDR31 : 31 ~ 63

P_VICx_PRIORITY0 **0x90010200 / 0x90020200** **VIC0/1 Priority 0 Register**

~ ~ ~

P_VICx_PRIORITY31 **0x9001027C / 0x9002027C** **VIC0/1 Priority 31 Register**

Bit	31	30	29	28	27	26	25	24
-								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
-				PRIORITY			
0	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[3:0]	PRIORITY	R/W	The P_VICx_PRIORITY0 ~ P_VICx_PRIORITY31 registers select vectored interrupt priority level (form 0-15) for each interrupt source. The defaults are on the same priority level, 15, which is the lowest. Hardware priority levels only take effect when multiple interrupts have the same priority level. The highest priority level is 0.	0000 = 1 st 0001 = 2 nd 0010 = 3 rd 0011 = 4 th 0100 = 5 th 0101 = 6 th 0110 = 7 th 0111 = 8 th 1000 = 9 th 1001 = 10 th 1010 = 11 th 1011 = 12 th 1100 = 13 th 1101 = 14 th 1110 = 15 th 1111 = 16 th

P_VICx_ADDRESS 0x90010F00 / 0x90020F00 VIC0/1 Address Register							
Bit	31	30	29	28	27	26	25
Function							
Default	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
CURIRQ							
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
CURIRQ							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
CURIRQ							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	CURIRQ	R/W	<p>The P_VICx_ADDRESS register contains the number of the currently active ISR.</p> <p>A read must only be performed while there is an active interrupt.</p> <p>A write of any value to this register clears the current interrupt</p>	-

6 Nand Flash & BCH Controller

6.1 Introduction

Nand flash controller of GPL329XX can support nand flash with standard Read/Write command with BCH correcting technology. Depend on the redundant size (spare size) of Nand flash, GPL329XXA bch can support 512B4/1K16/1K24/1K40/1K60 BCH type to make sure the correctness of user data. The Nand controller support 512Bytes Page (small Page) and 2K/4K/8K page size and flash, and all kind of nand type (SLC/MLC/TLC/NandRom).

As a huge storage device, NAND gate flash becomes more and more popular nowadays because of the characteristics of large capacity and relatively low price. To bring GPL329XXA series to more application fields, a NAND flash interface is incorporated. GPL329XXA provides easy-to-use control registers to generate read/write signals to access NAND flash. In addition, the data movement also accelerates because of the DMA channels provided to the NAND Flash. GPL329XXA supports 8 bits NAND type flash memories, with hardware ECC (Error Correction Code) and checksum or BCH 4/8/16/24/40/60 bit ECC supported.

- Supports hardware Solomon ECC
- Supports hardware BCH 4/8/16/24/40/60 ECC
- Programmable setup/hold timing for accessing.

6.2 Features

- Suitable to either SLC, MLC, TLC and NandRom module
- Compatible to any page size(512B/2K/4K/8K) of nand-flash modules
- Supports randomize to protect data for TLC nand flash type.
- Adaptable AC timing parameters for kinds of Nand-Flash modules
- Descriptor Base SW I/F for efficiency
- HW flow control automatic adjustment ability (Only for Standard Nand Command)
- Interleave facility for high performance.
- Supports Multi Plane (multi Chip Select) Nand flash.
- Supports hardware Solomon ECC
- Supports hardware BCH 4/8/16/24/40/60 ECC
- Programmable setup/hold timing for accessing.

6.3 Block Diagram

The following diagram is a functional block diagram of Nand Flash Controller of GPL329XX.

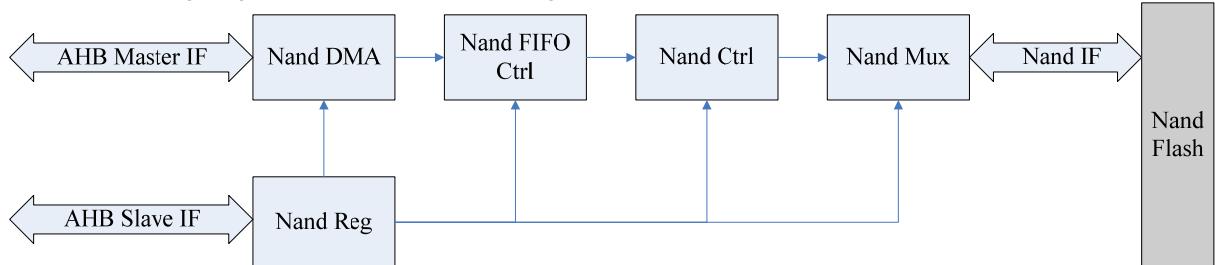


Fig 1.3 Nand flash controller diagram.

- **Nand DMA:** AHB-Lite Master, responsible for descriptor handling / data fetching in descriptor mode.
- **Nand FIFO Ctrl:** Embedded 16x32 bits FIFO, responsible for adjusting data flow between nand_dma and nand_ctrl. Write / read commands due to half-duplex process once.
- **Nand Ctrl:** Nand-Flash control signal generator, also responsible for managing control sequence.
- **Nand MUX:** PIO / Descriptor mode multiplexer selection.
- **Nand Reg:** I/O mapping registers.

6.4 Descriptor Defines

We should prepare the sequence memory(8x4 Bytes/Align16Addr) buffer to config the descriptor define.

Before Nand controller active, we need to register this Descriptor into P_NF_DESC_BASE_ADDR.

0	1	2	3
CMD1	CMD0	ReadStatusData	(CS ID + CMDType<<4)
4	5	6	7
SpareSize		DataSize	
8	9	10	11
INT Status		INT Mask	
12	13	14	15
AddrInfo0			
16	17	18	19
NextDesBP		AddrInfo1	
20	21	22	23
Data Addr			
24	25	26	27
Spare Addr			
28	29	30	31
BCH Parity Size		BCH Parity Alignment Size	

CMD0 : The Nand first command

CMD1: Nand Second (End) Command

Read Status Data: When send Read status command, the status response will record in this memory address

CS ID: Support Multi chip select, this can assign the Active CS to choice internal nand chip.

CMD Type:

- 0x1: READ
- 0x2: WRITE
- 0x3: Erase
- 0x4: Read Status
- 0x7: Manual Mode CMD Phase
- 0x8: Manual Mode Addr Phase
- 0x9: Manual mode Data Write
- 0xA: Manual Mode Dara Read
- 0xB: Manual Mode Spare/Redut Write
- 0xC: Manual Mode Spare/Redunt Read
- 0xD: SmallPage Read
- 0xF: Manual Mode Status Read

Spare Size: The Nand Spare area size

Data Size: The Nand Page User Data Area Size (512/2048/4096/8192 Bytes)

INT STATUS:

Bit[0]: INT Descriptor Done

Bit[1]: INT Descriptor Error

Bit[3]: INT Descriptor INVALID (0: well done)

Bit[12]: Nand Ready Response0 (for CS0)

Bit[13]: Nand Ready Response1 (for CS1)

Bit[14]: Nand Ready Response2 (for CS2)

Bit[15]: Nand Ready Response3 (for CS3)

Spare Size: Nand Spare Area Size, depend on nand spec.

Data Size: Nand User Data Size, depend on nand spec.

INT Status: the same as P_NF_INTR_STS.

INT Mask: the same as P_NF_INTR_MASK.

AddrInfo0[7:0]: NandAddr2

AddrInfo0[15:8]: NandAddr3

AddrInfo0[23:16]: NandAddr4

AddrInfo0[27:24]: Address Cycle

AddrInfo0[28]: if Nand has Spare/Redunt Area set 1, if no spare area set0 (Nand manager in chip or NandROM)

AddrInfo0[29]: fix set 1, for internal use.

AddrInfo0[30]: Descriptor save to HW message EN, If this bit is set to 1, HW will return to descriptor base address (DESC_BA) automatically after serving this descriptor for next time use.

AddrInfo0[31]: Descriptor Owner, 1:HW, 0:SW (usually, set 1)

AddrInfo1[15:0]: Next Descriptor BP

AddrInfo1[23:16]: NandAddr0

AddrInfo1[31:24]: NandAddr1

Data Addr: Data Buffer Address

Spare Addr: Spare/Redunt Buffer Address

BCH Parity Size: each kind of BCH algorithm has it's need parity data size, we assign in it.

BCH Parity Alignment Size: GPL329XX BCH decode request the BCH parity must alignment 16Bytes, so we should give the Alignment Size for Nand Controller to re-permutation the Parity data for BCH decoder.

6.5 Register Summary

Name	Address	Description
P_NF_CSR	0x93008000	Nand Control Register
P_NF_DESC_BASE_ADDR	0x93008004	Nand Description Base Address
P_NF_AC_TIMING	0x93008008	Nand AC Timing
P_NF_RDYBSY	0x9300800C	RDY pin read enable
P_NF_INTR_MASK	0x93008040	Nand Flash Work Status Mask
P_NF_INTR_STS	0x93008044	Nand Flash Work Status Register
P_NF_FAST_STATUS_CTRL	0x930080C0	Get Status Cmd Ctrl Reg
P_NF_FAST_STATUS	0x930080C4	Get Status Cmd response value
P_NF_FAST_STATUS_FLAG	0x930080C8	Get Status Cmd response OK flag
P_NF_RANDOMIZE_CTRL	0x93008090	RANDOMIZE Control Register
P_NF_SEED_GEN	0x93008098	Select Seed Engine Generator
P_NF_RANDOMIZE_PAGE_ID	0x9300809C	Register Randomize Ref Page ID
P_NF_PRE_NUM0	0x930080A0	SeedPara of RandomBit[0]
P_NF_PRE_NUM1	0x930080A4	SeedPara of RandomBit[1]
P_NF_PRE_NUM2	0x930080A8	SeedPara of RandomBit[2]
P_NF_PRE_NUM3	0x930080AC	SeedPara of RandomBit[3]
P_NF_PRE_NUM4	0x930080B0	SeedPara of RandomBit[4]
P_NF_PRE_NUM5	0x930080B4	SeedPara of RandomBit[5]
P_NF_PRE_NUM6	0x930080B8	SeedPara of RandomBit[6]
P_NF_PRE_NUM7	0x930080BC	SeedPara of RandomBit[7]
P_BCH_CFG	0x9300A000	BCH Config Register
P_BCH_DATA_PTR	0x9300A004	BCH data source start Ptr
P_BCH_PARITY_PTR	0x9300A008	BCH parity start Ptr
P_BCH_INT_STATUS	0x9300A00C	BCH working status register
P_BCH_SOFT_RST	0x9300A010	BCH reset register
P_BCH_INT_MASK	0x9300A014	BCH Status Mask
P_BCH_REPORT_STATUS	0x9300A018	BCH detail status report
P_BCH_SEC_ERR_REPORT	0x9300A01C	BCH error report
P_BCH_SEC_FAIL_REPORT	0x9300A020	BCH Fail decode report
P_BCH_MAX_ERR_BITS	0x9300A024	Return Max Error Bit Nums

6.6 Register Definition

P_NAND_CSR (0x93008000)

P_NAND_CSR			0x93008000			Nand Control Register		
Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

7	6	5	4	3	2	1	0
-	-	EDO_TYPE	Fetch_desc_en	AHB_ACC	Nand_PIO_SEL	Nand_PIO_EN	Nand_EN
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

Bit	Function	Type	Description	Condition
0	Nand_EN	W	Nand-flash function enable 0x0	0: Nand Disable, 1:Nand Enable
1	Nand_PIO_EN	W	Nand-flash PIO enable.	0: for descriptor base ; 1 for PIO base
2	Nand_PIO_SEL	W	Nand-flash PIO mode select. This bit is valid only when nand_pio_en is set to PIO base.	0 : PIO_MD1. All FM signals are fully controlled by SW (PIO_CTRL1 /CTRL2 / CTRL3), 1 : PIO_MD2. SW controls FM flows by PIO_CTRL4 ~ 8
3	AHB_ACC	W	AHB burst access number.	0 for burst 4, 1 for burst 8
4	Fetch_desc_en	W	Fetch descriptor enable. This bit is valid only when nand_pio_en is set to descriptor base. HW will stops servicing and sets this bit to 0 while either : No more valid descriptor Severity error happens SW should set this bit to 1 to trigger HW. HW will service SW's descriptor queue from DESC_BA.	1: HW fetch Descriptor; 0:SW descriptor
5	EDO_TYPE	W	EDO type nand-flash. SW should set this bit before HW acts.	1: HW, 0: SW

DESC BA			0x93008004			Nand Descriptor Base Address		
Bit	15	14	13	12	11	10	9	8
Function	Descriptor Memory Address (H)							
Default	0x0							

7	6	5	4	3	2	1	0
Descriptor Memory Address (L)							
0x0							

Bit	Function	Type	Description	Condition
[31:0]	Desc Base Addr	R/W	Nand Controller Descriptor Base Address	

P_NAND_RDYBSY (0x9300800C)

P_NAND_RDYBSY			0x9300800C			Nand Read Pin Read Enable		
Bit	15	14	13	12	11	10	9	8
Function	RDY3 En						RDY2 En	
Default	0x00						0x00	

7	6	5	4	3	2	1	0
RDY1 En							
0x00							

Bit	Function	Type	Description	Condition
[3:0]	RDYBSY0	W	RDYBSY0 Enable	RDYBSY0 Enable. 0001 : refer to 1st chip select; 0010 : refer to 2st chip select; 0100 : refer to 3st chip select; 1000 : refer to 4st chip select
[7:4]	RDYBSY1	W	RDYBSY1 Enable	RDYBSY0 Enable. 0001 : refer to 1st chip select; 0010 : refer to 2st chip select; 0100 : refer to 3st chip select; 1000 : refer to 4st chip select
[11:8]	RDYBSY2	W	RDYBSY2 Enable	RDYBSY0 Enable. 0001 : refer to 1st chip select; 0010 : refer to 2st chip select; 0100 : refer to 3st chip select; 1000 : refer to 4st chip select
[15:12]	RDYBSY3	W	RDYBSY3 Enable	RDYBSY0 Enable. 0001 : refer to 1st chip select; 0010 : refer to 2st chip select; 0100 : refer to 3st chip select; 1000 : refer to 4st chip select

P_NAND_INTR_STS (0x93008044)

P_NAND_INTR_STS			0x93008044			Nand Stats Flag		
Bit	15	14	13	12	11	10	9	8
Function	-	-	RB2_intr	RB1_intr	Rdata_intr	Wdata_intr	Addr_int	Cmd_intr
Default	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

7	6	5	4	3	2	1	0
-	read status error	ahb bus busy	invalid cmd_type	desc invalid	desc error	desc end	desc complete
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

Bit	Function	Type	Description	Condition
0	desc complete	RW	Descriptor Completion Interrupt. This bit is set to 1 regardless of any error or not. SW writes 1 to clear.	1: Clear
1	1 desc end	RW	Descriptor End Interrupt. Last of descriptor indicates HW to stop servicing further descriptor for SW handling. SW writes 1 to clear. Note that only 1st RDYBSY function is enabled by default, other RDYBSY	1: Clear
2	desc error	RW	Descriptor Error Interrupt. This bit is set to 1 either :Illegal payload / redund / spare BP AHB bus paralysis due to certain unexpected errors HW will stop servicing further descriptor for SW handling. SW writes 1 to clear.	1: Clear
3	desc invalid	RW	Descriptor Invalid Interrupt. This bit is set to 1 while no more valid descriptor is available. SW writes 1 to clear.	1: Clear
4	invalid cmd_type	RW	Invalid command type field indicated by SW. SW writes 1 to clear.	1: Clear
5	ahb bus busy	RW	DMA internal FIFO traffic is not facile. This may be caused either : AHB bus is too busy to serve nand-flash DMA. Each AHB burst number is not enough. SW writes 1 to clear.	1: Clear
6	read status error	RW	Read Status Error Interrupt. This bit is set to 1 while there is something wrong for read status. SW writes 1 to clear.	1: Clear
7	(reserved)	RW		

Bit	Function	Type	Description	Condition
8	Cmd_intr	RW	For PIO mode 2 / manual command, nand-flash command port finish interrupt. SW writes 1 to clear	1: Clear
9	Addr_int	RW	For PIO mode 2 / manual address, nand-flash address port finish interrupt. Write 1 to clear	1: Clear
10	Wdata_intr	RW	For PIO mode 2 / manual wdata, nand-flash write data port finish interrupt. SW writes 1 to clear	1: Clear
11	Rdata_intr	RW	For PIO mode 2 / manual rdata, nand-flash read data port finish interrupt. SW writes 1 to clear	1: Clear
12	RB1_intr	RW	Nand-flash R/B1 signal finish interrupt. SW writes 1 to clear. It is usually used for interleave.	1: Clear
13	RB2_intr	RW	Nand-flash R/B2 signal finish interrupt. SW writes 1 to clear. It is usually used for interleave.	1: Clear
[31:16]	Error_descriptor_addr	RW	This field is valid when HW returns error descriptor. SW might handle the consequent procedures according this pointer.	1: Clear

P_NF_AC_TIMING (0x93008008)

P_NF_AC_TIMING			0x93008008			AC Timing Adjust		
Bit	23	22	21	20	19	18	17	16
Function	Ready Status Pulse width			WAIT				
Default	0x1			0xf				

15	14	13	12	11	10	9	8
REC pulse width				ACT pulse width			
0x0				0x0			

7	6	5	4	3	2	1	0
ALE pulse width				CLE pulse width			
0x0				0x0			

Bit	Function	Type	Description	Condition
[3:0]	CLE pulse width	RW	CLE pulse width. Unit : clock cycle	NAND cycle time is referenced with Sys AHB Clock
[7:4]	ALE pulse width	RW	ALE pulse width. Unit : clock cycle	NAND cycle time is referenced with Sys AHB Clock
[11:8]	ACT pulse width	RW	WRNN / RDNN active pulse width. Unit : clock cycle	NAND cycle time is referenced with Sys AHB Clock
[15:12]	REC pulse width	RW	WRNN / RDNN clock cycle	NAND cycle time is referenced with Sys AHB Clock
[19:16]	WAIT	RW		NAND cycle time is referenced with Sys AHB Clock
[25:20]	Ready Status Pulse width	RW	Read Status pulse. Unit : clock cycle	NAND cycle time is referenced with Sys AHB Clock

P_BCH_CFG (0x9300A000)

P_BCH_CFG			0x9300A000			BCH Config Register		
Bit	31	30	29	28	27	26	25	24
Function	reserve	sbyte mode			reserve	reserve	Correct Mask	
Default	0	0			0	0	0	

23	22	21	20	19	18	17	16	
reserve	reserve	reserve	Sector Number					
0x0				0x0				

15	14	13	12	11	10	9	8
reserve				DataMode	Correct Mode		
0x0				0x0			

7	6	5	4	3	2	1	0
reserve	reserve	reserve	Encode/Decode	reserve	reserve	reserve	Start Trigger
0x0				0	0	0	0

BIT	Name	Type	Function	Default Value
[0]	start trigger	WP	write 1 to trigger BCH encoder or decoder	0
[4]	encode/decode	R/W	value 0 : select encoder ; value 1 : select decoder	0
[10:8]	correct mode	R/W	correct mode : 1'b0=>1KB/16-bit; 1'b1=>1KB/24-bit	0
[11]	data mode	R/W	Only valid for 24-bit correct mode 0: parity sector size=64bytes with 32bytes alignment 1: parity sector size=48bytes with 16bytes alignment For 16-bit correct mode, parity sector size fixed to 32 bytes with 32bytes alignment	
[20:16]	sector number	R/W	Set the number N of block (1block=1K Byte) for BCH N=0~31(1~32 block)	0
[25:24]	corr mask	R/W		
[30:28]	sbyte mode	R/W		
[31]	no small block auto shift	R/W		

P_BCH_DATA_PTR(0x9300A004)

BIT	Name	Type	Function	Default Value
[31:0]	data pointer	R/W	bch data pointer address, align to word(4 bytes) bit[1:0] will be set to 2'b00 automatically	0

P_BCH_PARITY_PTR(0x9300A008)

BIT	Name	Type	Function	Default Value
[31:0]	parity pointer	R/W	bch parity pointer address align to word(4 bytes) bit[1:0] will be set to 2'b00 automatically	0

P_BCH_INT_STATUS (0x9300A00C)

P_BCH_INT_STATUS			0x9300A00C			BCH Interrupt Status Flags		
Bit	23	22	21	20	19	18	17	16
Function	reserve	reserve	reserve	correct number				
Default	0	0	0	0				

15	14	13	12	11	10	9	8
reserve	reserve	reserve	current number				
0	0	0	0				

P_BCH_INT_STATUS			0x9300A00C				BCH Interrupt Status Flags		
	7	6	5	4	3	2	1	0	
	reserve	reserve	reserve	Busy	reserve	reserve	reserve	BCH INT	
	0	0	0	0	0	0	0	0	

BIT	Name	Type	Function	Default Value
[0]	bch interrupt	R/W	bch interrupt register, write 1 clear	0
[4]	busy	R	Busy flag, pull high when start trigger and pull low when work finish	0
[12:8]	current number	R	data block number counter	0
[20:16]	correct number	R	corrected block number counter	0

P_BCH_SOFT_RST (0x9300A010)

BIT	Name	Type	Function	DefaultValue
[0]	clr_flag	R/W	User writes 1 to reset bch, it will return to 0 when reset finished.	0

P_BCH_INT_MASK (0x9300A014)

BIT	Name	Type	Function	DefaultValue
[0]	interrupt mask	R/W	1/0: when bch work finish, interrupt asserts or not	0

P_BCH_REPORT_STATUS (0x9300A018)

P_BCH_REPORT_STATUS			0x9300A018			BCH Config Register		
Bit	31	30	29	28	27	26	25	24
Function	reserve	reserve	reserve	reserve	reserve	reserve	OO Flag	FF Flag
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
reserve	reserve	reserve	reserve	reserve	reserve	Error Nums[9:8]	
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Error Nums[7:0]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
reserve	reserve	reserve	decode_fail	reserve	reserve	reserve	finish
0	0	0	0	0	0	0	0

BIT	Name	Type	Function	Default Value
[0]	finish	R	1/0: bch encoder /decoder finish	0
[4]	decode_fail	R	1/0: bch decoder fail or not	0
[17:8]	error_num	R	count total error bit of this work, maximum= 24*32=768 bits	0
[24]	FF_flag	R	Indicate that all block data are 0xFF	0
[28]	OO_flag	R	Indicate that all block data are 0x00	0

P_BCH_SEC_ERR REPORT (0x9300A01C)

BIT	Name	Type	Function	Default Value
[31:0]	sector error	R	It represents the error block of this work. Note the error block still can be corrected by bch, when can't corrected, "decoder fail" will assert. EX: bit 0 : 1 _{st} block has error or not bit 1 : 2 _{nd} block has error or not 32'hffff_0000: blcok 1~16 has no errors. block 17~32 has errors	0

P_BCH_SEC_FAIL_REPORT (0x9300A020)

BIT	Name	Type	Function	Default Value
[31:0]	sector decode fail	R	<p>It represents decoded fail block of this work. EX: bit 0 : 1_{st} block decode fail or not bit 1 : 2_{nd} block decode fail or not 32'hffff_0000: blcok 1~16 decode success. block 17~32 decode fail</p>	0

P_BCH_MAX_ERR_BITS (0x9300A024)

0x9300A024					BCH MAX Err Bit Nums return			
Bit	15	14	13	12	11	10	9	8
Function	reserve	reserve	reserve	reserve	reserve	Max Error Num[10:8]		
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Max Error Num[10:8]							
0x0							

BIT	Name	Type	Function	Default Value
[10:0]	max error num	R	It represents the max err number occur in a single block	0

7 Watch-dog timer (WDT)

7.1 Introduction

Watch-dog timer is a 16 bits timer used to resume the controller operation when it is disturbed by mal-functions such as noise and system errors. It uses a free running 27MHz clock as a clock source. The timer behavior is the same as normal timer; expect the capability to reset whole system. The count interval of timer is determined by pre-scale register. WDT generates the interrupt and reset signal when the internal counter reaches at the end value. A compare register is used to generate PWM output (WDT_OUT).

7.2 Features

- Support 3 WDTs.
- 16 bits timer.
- Generate global chip reset when time out.

7.3 Register Summary

Name	Address	Description
P_WDT0_CTRL	0x90001000	WDT control register.
P_WDT1_CTRL	0x90001020	
P_WDT2_CTRL	0x90001040	
P_WDT0_PSR	0x90001004	WDT pre-scale register.
P_WDT1_PSR	0x90001024	
P_WDT2_PSR	0x90001044	
P_WDT0_LDR	0x90001008	WDT load value register.
P_WDT1_LDR	0x90001028	
P_WDT2_LDR	0x90001048	
P_WDT0_VLR	0x9000100C	WDT current counter register.
P_WDT1_VLR	0x9000102C	
P_WDT2_VLR	0x9000104C	
P_WDT0_CMP	0x90001010	WDT compare register.
P_WDT1_CMP	0x90001030	
P_WDT2_CMP	0x90001050	

7.4 Register Definition

P_WDT0_CTRL	0x90001000	WDT control register.
P_WDT1_CTRL	0x90001020	
P_WDT2_CTRL	0x90001040	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	

Function	- - - - - - - - - - - - - - - -
Default	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit	Function	Type	Description	Condition
[31:5]			Reserved	
4	PWM	R/W	PWM mode enable bit.	Write 0 = PWM mode. Write 1 = WDT mode.
3	RE	R/W	Reset enable bit.	Write 0 = Disable reset generation. Write 1 = Enable reset generation.
2	OE	R/W	Output enable bit.	Write 0 = Disable. Write 1 = Enable.
1	IE	R/W	Interrupt enable bit.	Write 0 = Disable. Write 1 = Enable.
0	TE	R/W	Timer enable bit.	Write 0 = Disable. Write 1 = Enable.

P_WDT0_PSR	0x90001004	WDT pre-scale register.
P_WDT1_PSR	0x90001024	
P_WDT2_PSR	0x90001044	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	

Function	- - - - - - - - - - - - - - - -
Default	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit	Function	Type	Description	Condition
[31:16]			Reserved.	
[15:0]	PSR	R/W	Pre-scale value.	

P_WDT0_LDR **0x90001008** WDT load value register.

P_WDT1_LDR **0x90001028**

P_WDT2_LDR **0x90001048**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LDR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]			Reserved.	
[15:0]	LDR	R/W	Load value. When the timer reaches the end, counter value will reload this value to count automatically.	

P_WDT0_VLR **0x9000100C** WDT current counter register.

P_WDT1_VLR **0x9000102C**

P_WDT2_VLR **0x9000104C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	VLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]			Reserved.	
[15:0]	VLR	R/W	Counter value. Read operation on this register returns the current value of internal counter. Writing 0x1225 will reset the value of internal counter. Writing other value will generate interrupt and reset signals in WDT mode.	

P_WDT0_CMP
0x90001010
WDT compare register.
P_WDT1_CMP
0x90001030
P_WDT2_CMP
0x90001050

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CMP															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]			Reserved.	
[15:0]	CMP	R/W	Compare value. This values is ude only in PWM output mode.	

8 APBDMA0

8.1 Introduction

The APBDMA controller built in this chip has 4-channel. Each APBDMA channel is capable of doing APBDMA transfer from MIU to peripheral or from peripheral to MIU.

8.2 Features

- 4 independent DMA Channel
- MIU to peripheral, peripheral to MIU
- 8-bit single, 16-bit single, 32-bit single, and 32-bit burst transfer

8.3 Operation Mode

DMA controller can offer four Channels to R/W memory for APB peripheral modules at the same time, each channel can be established into four kinds of transmission means:

8 bits single transfer

16 bits single transfer

32 bits single transfer

32 bits burst transfer

DMA has two kinds of starting mode:

Case 1: When channel enable is set to 1, DMA controller begins to read or write the data successively and terminates when it is done.

Case 2: Channel enable is set to 1, DMA controller carries out Read or Write movements once when APB peripheral modules send out REQ and terminate when all REQ finish sending to DMA.

The space planning of Memory has two ways when DMA R/W MIU:

Case 1: Single buffering. By specifying the start and end of DMA address, DMA controller only reads or writes this memory space, and is terminated by sending out IRQ when read or write operation completes.

Case 2: Double buffering. By specifying the start and end address of BUFFER A and BUFFER B at the same time, DMA will read or write BUFFERA and BUFFERB alternately. It will generate an IRQ signal when either transfer completes. Set channel enable back to '0' can generate a DMA operation cease request, and DMA controller will terminate its operation when the current block transfer completes.

There are two kinds of addressing for DMA to that APB peripheral modules R/W making location ways of PORT

Case 1: Regular Address

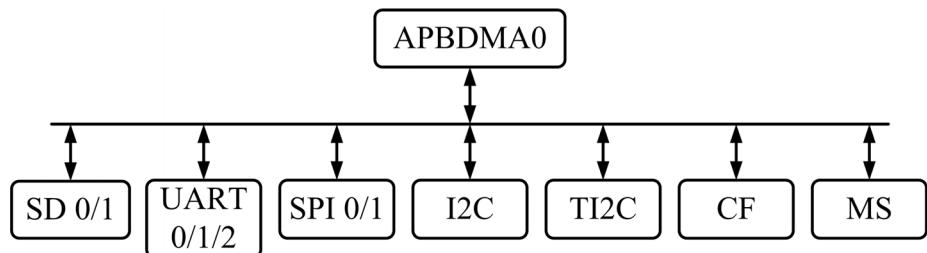
Case 2: Continuous Address

8.4 DMA End

When DMA finishes, users need to write '1' to the corresponding IRQ STATUS bit to clear DMA interrupt and write '0' to the corresponding channel enable

8.5 Peripheral

The following figure shows the peripherals that can access data via APBDMA. Before using these peripherals, user should enable the clock of APBDMA even though you don't need to use APBDMA to access data.



8.6 Register Summary

Name	Address	Description
P_SCUC_SYS_RATIO_UPDATE	0x92005028	System clock ration updata control.
P_SCUC_SYS_APB_RATIO	0x9200510C	System APB clock ratio.
P_APBDMA0_STATUS	0x92B00000	APBDMA0 channel status register.
P_APBDMA0_INT	0x92B00004	APBDMA0 Interrupt Register.
P_APBDMA0_SA0A	0x92B00008	APBDMA0 buffer A start address register.
P_APBDMA0_SA1A	0x92B0000C	
P_APBDMA0_SA2A	0x92B00010	
P_APBDMA0_SA3A	0x92B00014	
P_APBDMA0_EA0A	0x92B00018	APBDMA0 buffer A end address register.
P_APBDMA0_EA1A	0x92B0001C	
P_APBDMA0_EA2A	0x92B00020	
P_APBDMA0_EA3A	0x92B00024	
P_APBDMA0_SA0	0x92B00028	APBDMA0 module port access start address register.
P_APBDMA0_SA1	0x92B0002C	
P_APBDMA0_SA2	0x92B00030	
P_APBDMA0_SA3	0x92B00034	
P_APBDMA0_SA0B	0x92B0004C	APBDMA0 buffer B start address register.
P_APBDMA0_SA1B	0x92B00050	
P_APBDMA0_SA2B	0x92B00054	
P_APBDMA0_SA3B	0x92B00058	

Name	Address	Description
P_APBDMA0_EA0B	0x92B0005C	APBDMA0 buffer B end address register.
P_APBDMA0_EA1B	0x92B00060	
P_APBDMA0_EA2B	0x92B00064	
P_APBDMA0_EA3B	0x92B00068	
P_APBDMA0_CTRL0	0x92B0006C	APBDMA0 channel control register.
P_APBDMA0_CTRL1	0x92B00070	
P_APBDMA0_CTRL2	0x92B00074	
P_APBDMA0_CTRL3	0x92B00078	
P_APBDMA0_RESET	0x92B0007C	APBDMA0 software reset register.

8.7 Register Definition

P_APBDMA0_STATUS 0x92B00000 APBDMA0 channel status register.																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	CH3_B	CH2_B	CH1_B	CH0_B	-	-	-	-	CH3_BUSY	CH2_BUSY	CH1_BUSY	CH0_BUSY
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description													Condition
[31:4]			Reserved.													
[11:8]	CHX_B	R	DMA channel buffer status. This bit indicates which buffer APBDMA channel is accessing.													0 = Buffer A. 1 = Buffer B.
[7:4]			Reserved.													
[3:0]	CHX_BUSY	R	DMA channelX busy status. This bit indicates whether APBDMA channel is busy or not.													0 = Not busy. 1 = Busy.

P_APBDMA0_INT 0x92B00004 APBDMA0 Interrupt Register.																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	CH3_IRQ	CH2_IRQ	CH1_IRQ	CH0_IRQ
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:4]			Reserved.	
[3:0]	CHX_IRQ	R/W	DMA channelX IRQ status. Write "1" to clear the flag. This bit is set to "1" by hardware if APBDMA finishes data accessing.	Read 0 = DMA IRQ not occurred. Read 1 = DMA IRQ occurred. Write 0 = No effect. Write 1 = Clear DMA IRQ flag.

P_APBDMA0_SA0A **0x92B00008** APBDMA0 buffer A start address register.

P_APBDMA0_SA1A **0x92B0000C**

P_APBDMA0_SA2A **0x92B00010**

P_APBDMA0_SA3A **0x92B00014**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function																
Default																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default																

Bit	Function	Type	Description	Condition
[31:0]	BUFA_SA	R/W	Buffer A start address for channelX.	

P_APBDMA0_EA0A **0x92B00018** APBDMA0 buffer A end address register.

P_APBDMA0_EA1A **0x92B0001C**

P_APBDMA0_EA2A **0x92B00020**

P_APBDMA0_EA3A **0x92B00024**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function																
Default																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default																

Bit	Function	Type	Description	Condition
[31:0]	BUFA_EA	R/W	Buffer A end address for channelX.	

P_APBDMA0_SA0 0x92B00028 APBDMA0 module port access start address register.

P_APBDMA0_SA1 0x92B0002C

P_APBDMA0_SA2 0x92B00030

P_APBDMA0_SA3 0x92B00034

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	APB_PORT															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	APB_PORT															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[31:0]	APB_PORT	R/W	APB module access port start address for channelX.	

P_APBDMA0_SA0B 0x92B0004C APBDMA0 buffer B start address register.

P_APBDMA0_SA1B 0x92B00050

P_APBDMA0_SA2B 0x92B00054

P_APBDMA0_SA3B 0x92B00058

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	BUFA_SB															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BUFA_SB															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[31:0]	BUFA_SB	R/W	Buffer B start address for channelX.	

P_APBDMA0_EA0B 0x92B0005C APBDMA0 buffer B end address register.

P_APBDMA0_EA1B 0x92B00060

P_APBDMA0_EA2B 0x92B00064

P_APBDMA0_EA3B 0x92B00068

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	BUFA_EB															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BUFA_EB															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[31:0]	BUFA_EB	R/W	Buffer B end address for channelX.	

P_APBDMA0_CTRL0 **0x92B0006C** APBDMA0 channel control register.

P_APBDMA0_CTRL1 **0x92B00070**

P_APBDMA0_CTRL2 **0x92B00074**

P_APBDMA0_CTRL3 **0x92B00078**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	CHEN	IRQEN	TXMODE	MIUMODE	APBMODE	DMAMODE	DIR			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]			Reserved.	
7	CHEN	R/W	DMA channel enable.	0 = Disable DMA channel. 1 = Enable DMA channel.
6	IRQEN	R/W	DMA channel IRQ enable.	0 = Disable DMA IRQ. 1 = Enable DMA IRQ.
[5:4]	TXMODE	R/W	DMA channel transfer mode.	00 = 8 bits single transfer. 01 = 16 bits single transfer. 10 = 32 bits single transfer. 11 = 32 bits burst transfer.
3	MIUMODE	R/W	MIU memory mode.	0 = Single buffer. 1 = Double buffer.
2	APBMODE	R/W	ChannelX DMA's APB peripheral modules location mode.	1 = Regular mode.
1	DMAMODE	R/W	ChannelX DMA mode selection.	1 = Polling mode.
0	DIR	R/W	ChannelX direction for R/W.	0 = MIU TO APB. 1 = APB TO MIU.

P_APBDMA0_RESET **0x92B0007C** APBDMA0 software reset register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	CH3_RESET	CH2_RESET	CH1_RESET	CH0_RESET
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:4]			Reserved.	
[3:0]	CHX_RESET	R/W	ChannelX reset. Write “1” to this bit will reset APBDMA Channel. This bit will be cleared to “0” after reset complete by hardware.	0 = Disable. 1 = Enable.

9 APBDMA1

9.1 Introduction

The APBDMA controller built in this chip has 2-channel. Each APBDMA channel is capable of doing APBDMA transfer from MIU to peripheral or from peripheral to MIU.

9.2 Features

- 2 independent DMA Channel
- MIU to peripheral, peripheral to MIU
- 8-bit single, 16-bit single, 32-bit single, and 32-bit burst transfer

9.3 Operation Mode

DMA controller can offer four Channels to R/W memory for APB peripheral modules at the same time, each channel can be established into four kinds of transmission means:

8 bits single transfer

16 bits single transfer

32 bits single transfer

32 bits burst transfer

DMA has two kinds of starting mode:

Case 1: When channel enable is set to 1, DMA controller begins to read or write the data successively and terminates when it is done.

Case 2: Channel enable is set to 1, DMA controller carries out Read or Write movements once when APB peripheral modules send out REQ and terminate when all REQ finish sending to DMA.

The space planning of Memory has two ways when DMA R/W MIU:

Case 1: Single buffering. By specifying the start and end of DMA address, DMA controller only reads or writes this memory space, and is terminated by sending out IRQ when read or write operation completes.

Case 2: Double buffering. By specifying the start and end address of BUFFER A and BUFFER B at the same time, DMA will read or write BUFFERA and BUFFERB alternately. It will generate an IRQ signal when either transfer completes. Set channel enable back to '0' can generate a DMA operation cease request, and DMA controller will terminate its operation when the current block transfer completes.

There are two kinds of addressing for DMA to that APB peripheral modules R/W making location ways of PORT

Case 1: Regular Address

Case 2: Continuous Address

9.4 DMA End

When DMA finishes, users need to write '1' to the corresponding IRQ STATUS bit to clear DMA interrupt and write '0' to the corresponding channel enable

9.5 Control Register

Name	Address	Description
P_APBDMA1_STATUS	0x93010000	APBDMA1 channel status register.
P_APBDMA1_INT	0x93010004	APBDMA1 Interrupt Register.
P_APBDMA1_SA0A P_APBDMA1_SA1A	0x93010008 0x9301000C	APBDMA1 buffer A start address register.
P_APBDMA1_EA0A P_APBDMA1_EA1A	0x93010018 0x9301001C	APBDMA1 buffer A end address register.
P_APBDMA1_SA0 P_APBDMA1_SA1	0x93010028 0x9301002C	APBDMA1 module port access start address register.
P_APBDMA1_SA0B P_APBDMA1_SA1B	0x9301004C 0x93010050	APBDMA1 buffer B start address register.
P_APBDMA1_EA0B P_APBDMA1_EA1B	0x9301005C 0x93010060	APBDMA1 buffer B end address register.
P_APBDMA1_CTRL0 P_APBDMA1_CTRL1	0x9301006C 0x93010070	APBDMA1 channel control register.
P_APBDMA1_RESET	0x9301007C	APBDMA1 software reset register.

P_APBDMA1_STATUS 0x93010000 APBDMA1 channel status register.

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	Ch1_DM A_Status	Ch0_DMA Status

Default 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
-	-	-	-	-	-	Ch1_Busy	Ch0_Busy
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]	-	-	Reserved	-
[9:8]	Ch_Busy	R/W	Channel0 and channel1 Busy Status	0: Not Busy 1: Busy
[7:2]	-	-	Reserved	-
[1:0]	Ch_DMA Status	R/W	Channel0 and channel1 DMA Status	Read 0:Access Buffer A Read 1: Access Buffer B

P_APBDMA1_INT

0x93010004

APBDMA1 Interrupt Register.

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	1	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	-	-	-	-	-	Ch1_IRQ Status	Ch0_IRQ Status
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:2]	-	-	Reserved					-
[1:0]	CHX_IRQ	R/W	DMA channelX IRQ status. Write "1" to clear the flag. This bit is set to "1" by hardware if APBDMA finishes data accessing.					Read 0 = DMA IRQ not occurred. Read 1 = DMA IRQ occurred. Write 0 = No effect. Write 1 = Clear DMA IRQ flag.

P_APBDMA0_SA0A

0x93010008

0x9301000C

APBDMA0 buffer A start address register

P_APBDMA0_SA1A

Bit	31	30	29	28	27	26	25	24
Function	BUFA_SA							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
BUFA_SA							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
BUFA_SA							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
BUFA_SA							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	BUFA_SA	R/W	Buffer A Start Address For ChannelX					-

P_APBDMA0_EA0A	0x93010018	APBDMA1 buffer A end address register.						
P_APBDMA0_EA1A	0x9301001C							
Bit	31	30	29	28	27	26	25	24
Function	BUFA_EA							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
BUFA_EA								
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
BUFA_EA								
0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	
BUFA_EA								
0	0	0	0	0	0	0	0	
Bit	Function	Type	Description					Condition
[31:0]	BUFA_EA	R/W	Buffer A End Address For ChannelX					-

P_APBDMA0_SA0	0x93010028	APBDMA1 module port access start address						
P_APBDMA0_SA1	0x9301002C	register.						
Bit	31	30	29	28	27	26	25	24
Function	APB_PORT							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
APB_PORT								
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
APB_PORT								
0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	
APB_PORT								
0	0	0	0	0	0	0	0	

Bit	Function	Type	Description					Condition
[31:0]	APB_PORT	R/W	APB Module Access Port Start Address For ChannelX					-

P_APBDMA0_SA0B
0x9301004C
APBDMA1 buffer B start address register
P_APBDMA0_SA1B
0x93010050

Bit	31	30	29	28	27	26	25	24
-----	----	----	----	----	----	----	----	----

Function	BUFB_SA							
----------	---------	--	--	--	--	--	--	--

Default 0 0 0 0 0 0 0 0

23	22	21	20	19	18	17	16	
----	----	----	----	----	----	----	----	--

BUFB_SA

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8	
----	----	----	----	----	----	---	---	--

BUFB_SA

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0	
---	---	---	---	---	---	---	---	--

BUFB_SA

0 0 0 0 0 0 0 0

Bit	Function	Type	Description					Condition
[31:0]	BUFB_SA	R/W	Buffer B Start Address For ChannelX					-

P_APBDMA0_EA0B
0x9301005C
APBDMA1 buffer B end address register.
P_APBDMA0_EA1B
0x93010060

Bit	31	30	29	28	27	26	25	24
-----	----	----	----	----	----	----	----	----

Function	BUFB_EA							
----------	---------	--	--	--	--	--	--	--

Default 0 0 0 0 0 0 0 0

23	22	21	20	19	18	17	16	
----	----	----	----	----	----	----	----	--

BUFB_EA

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8	
----	----	----	----	----	----	---	---	--

BUFB_EA

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
BUFB_EA							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:0]	BUFB_EA	R/W	BufferB End Address For ChannelX				-

P_APBDMA0_CTRL0
0x9301006C
P_APBDMA0_CTRL1
0x93010070
APBDMA1 channel control register

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChEn	IRQEN	TXMODE	MIUMODE	APBMODE	DMAMODE	DIR	
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:8]	-	-	Reserved				-
7	ChEn	R/W	Channel Enable				0:Disable DMA Channel 1:Enable DMA Channel
6	IRQEN	R/W	DMA Channel IRQ Enable				0:Disable DMA IRQ 1:Enable DMA IRQ
[5:4]	TXMODE	R/W	Channel transfer mode				00: 8 bits single transfer 01: 16 bits single transfer 10: 32 bits single transfer 11: 32 bits burst transfer
3	MIUMODE	R/W	MIU Memory mode				0: Single Buffer 1: Double Buffer
2	APBMODE	R/W	ChannelX DMA's APB peripheral Modules location mode				0: Continuous mode 1: Regular mode
1	DMAMODE	R/W	ChannelX DMA Mode Selection				0:auto mode 1: Polling mode
0	DIR	R/W	Channel direction for R/W				0: MIU TO APB 1. APB TO MIU

P_DMAA_RST								0x9301007C								APBDMA1 Software Reset Register								
Bit	15	14	13	12	11	10	9	8																
Function	-																							
Default	0	0	0	0	0	0	0	0																
	7	6	5	4	3	2	1	0									CH1_RESET				CH0_RESET			
	0	0	0	0	0	0	0	0																

Bit	Function	Type	Description	Condition
[1:0]	CHX_RESET		ChannelX Reset Write “1” to this bit will reset APBDMA Channel. This bit will be cleared to “0” after reset complete by hardware.	0 = Disable 1 = Enable

10 SDMA

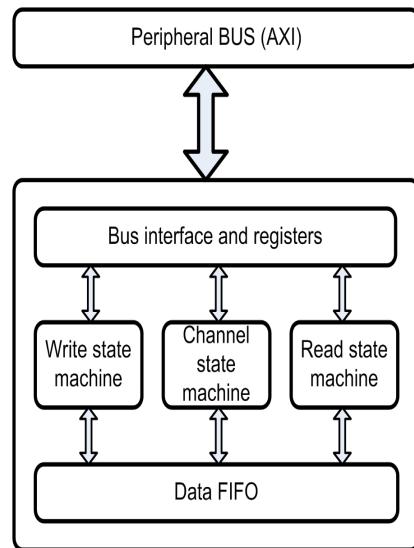
10.1 Introduction

System DMA (SDMA) is a powerful DMA engine in this chip. As everyone knows, DMA can offload processor loading and increase system performance. SDMA support not only normal memory moving but also index and link-list mode memory access which will introduce at the following subsection. It also provides various data width and burst mode for more flexible usage. User can implement some memory access functions in C library, like memset, memcpy and so on, by using SDMA function.

10.2 Features

- 2 bidirectional DMA channels.
- Pause and resume channel.
- Addressing mode: non-incrementing/Incrementing/Constant Fill/Index (IDX) for source/destination addressing.
- Link-list mode support
- Programmable DMA data bus width (8, 16, 32 bit width).
- Programmable DMA burst length (1, 4, 8, 16).
- Data FIFO: 16x32-bit per channel.
- Big-endian and little-endian switch support.
- TCP/IP Checksum Offload Engine (TOE).
- Column Copy mode support.
- Boundary data mask support.

10.3 Block Diagram



10.4 Transfer Operation Control

There are various transfer modes in SDMA function. User can configure SDMA registers for each transfer mode. Channel configuration and transfer modes will be introduced in the below subsections.

10.4.1 Channel Enable

When **CHEN** bit in **P_SDMAX_STATUS**¹ is set, it means the channel is ready to start. When the channel starts to operate, **CHEN** bit will be auto cleared, immediately.

NOTE 1 P_SDMAX_STATUS: X means 0 or 1 to indicate channel 0 or 1.

10.4.2 Pause Operation

SDMA transfer can be paused when it is still running by setting **PAR** bit in **P_SDMAX_STATUS**. Channel will be paused after the current transaction finished. And then, **PAR** bit and **PAU** bit will be cleared and set, respectively at the same time. User must check the **PAU** bit in **P_SDMAX_STATUS** to make sure that channel is paused.

10.4.3 Resume Operation

During the pause state, channel can be resumed by setting **PAR** bits to 1.

10.4.4 Stop Operation

Setting **STOP** bit in **P_SDMAX_STATUS** will force stop SDMA transaction. Setting **STOP** bit during channel is still operating is not suggested, because it may cause unpredictable result. Programmer must comply with the following procedure to stop the channel,

Step1: Pause the channel (Set **PAR** to 1)

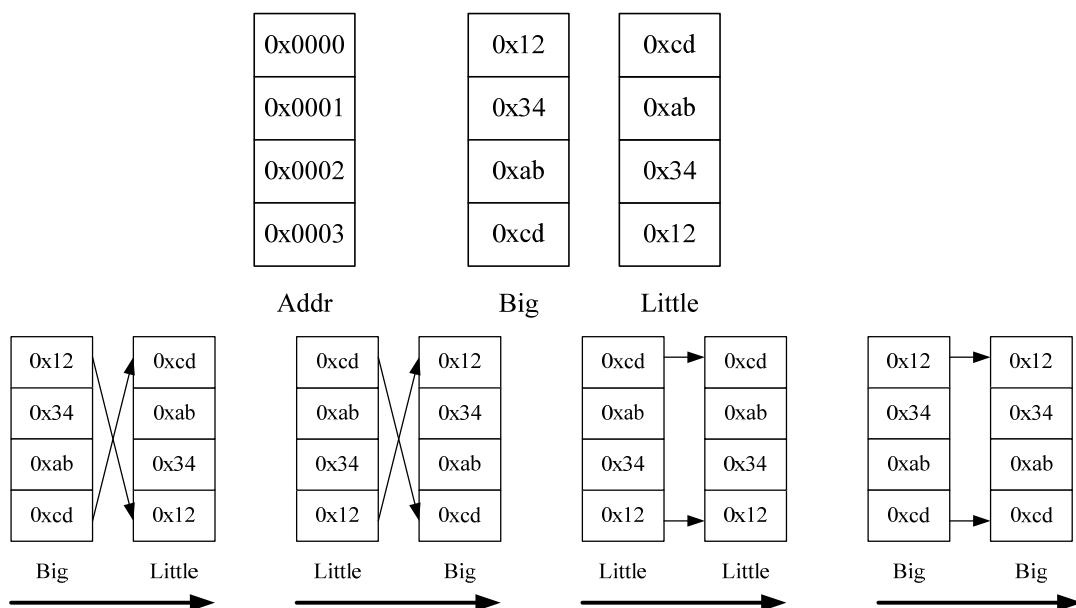
Step2: Check the channel status to paused (Keep polling **PAU** until 0)

Step3: Stop the channel (Set **STOP** to 1)

After step1~step3, the driver can resetting the configuration registers to restart a new transfer.

10.4.5 Endian Switch

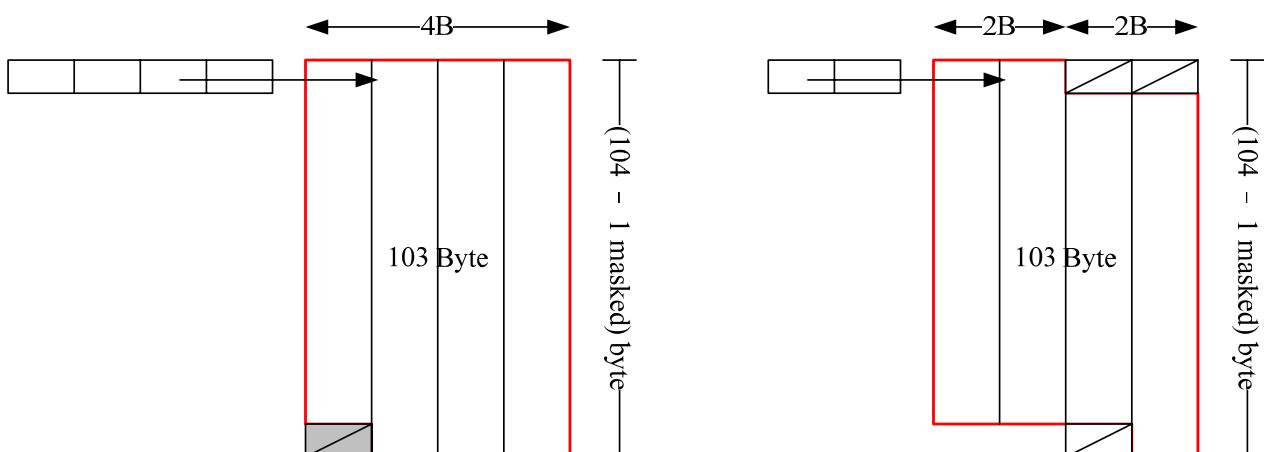
EDSW bit in **P_SDMAX_STATUS** can enable/disable the endian switch during the DMA transferring. If the endian type is different for the source and destination port, DMA will swap the data automatically. The definition of the big/little endian is shown at the below figure. For big endian, high byte data puts at the low byte address, and the little endian puts the low byte data at the low byte address. This function is only for source and destination data width as 4 bytes.



10.4.6 Data Mask

The **P_SDMAX_SBSIZE** (block size) should be **DSIZE** alignment. If not, **EMASK** in **P_SDMAX_ECFG** should be configured to mask the last write data. For example, if **DSIZE** is 4 byte, and the transferred data length is 103 bytes, then **P_SDMAX_SBSIZE** should be 104 and the **EMASK[3:0] = 0x8**.

There are two examples at following figure. The left one is the case **DSIZE = 2** (4 byte mode), and the right one is **DSIZE = 1** (2 byte mode). The data sizes are both 103 bytes. Because of the different starting address, the **EMASK** should be different. For the left case, **EMASK[3:0] = 0x8** to mask the last byte located at the left-bottom corner. For the right case, **DSIZE = 2** is illegal because of the **DSIZE** un-alignment. **DSIZE** could be 1 or 0. If **DSIZE = 1**, **EMASK[3:0] = 0x2** to mask the last byte. If **DSIZE = 0**, **EMASK[3:0]** should be **0x0**.



10.4.7 Addressing Mode

SDMA supports 4 types of addressing modes for source and destination address, non-increment, increment, constant-fill and index mode. User can combine each mode for source and destination more flexible. Please note that if data width is set as 4 bytes, start address must be 4 byte alignment. Similarly, when data width is set as 2 bytes, start address must be 2 byte alignment. These four modes will be shown as following.

10.4.7.1 Non-Increment Mode

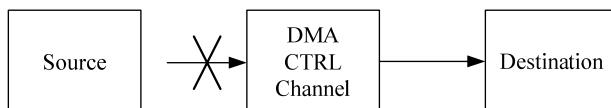
When setting **DAM** or **SAM** in **P_SDMAX_CFG** to non-increment mode, the transferring address for source port or destination port would keep constant during the transaction. For example, source port is non-increment mode and the **P_SDMAX_SADR** = 0x300C, then the reading address will be fixed as 0x300C, 0x300C ... until transfer end.

10.4.7.2 Increment Mode

When **DAM** or **SAM** is set as increment mode, the address would increase by data width every transaction. For example, set **P_SDMAX_SADR** = 0x300C and the **SSIZE** = 0x1 (2 bytes) in **P_SDMAX_CFG**, then the reading address will be 0x300C, 0X300E, 0x3010, etc.

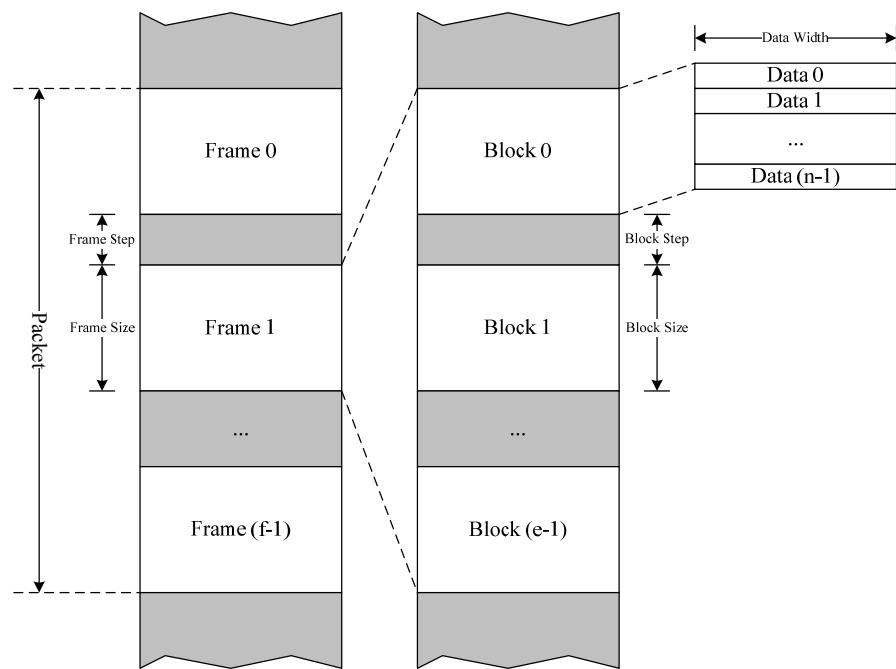
10.4.7.3 Constant Fill Mode

This mode is similar with non-increment mode. The difference of these two modes is non-increment will read source address each transaction, and constant fill would not do this. It will read **P_SDMAX_SADR** as its data and write to the destination address.



10.4.7.4 Index Mode

As shown as following figure, user can configure three layers of parameter to index addressing mode. Blcok size (**P_SDMAX_S/DBSIZE**) and block step size (**P_SDMAX_S/DBSTEP**) are the parameters for first layer. Frame size (**P_SDMAX_S/DFSIZE**) and frame step size (**P_SDMAX_S/DFSTEP**) are the parameters for second layer. Packet size (**P_SDMAX_S/DPSIZE**) is the parameters for third layer.

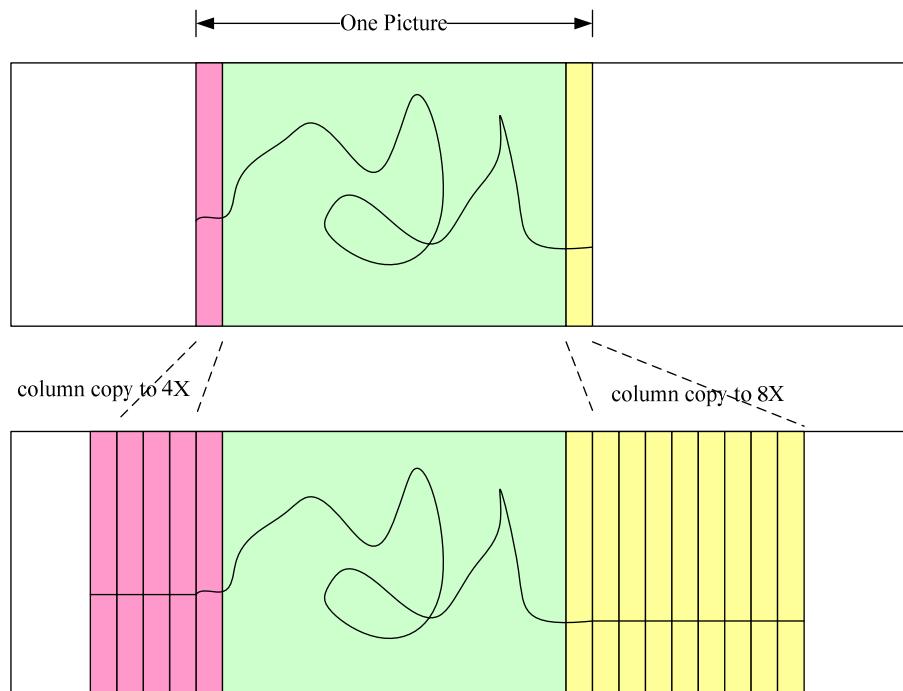


One block size	= P_SDMAX_S/DBSIZE
One frame size	= (P_SDMAX_S/DBSIZE + P_SDMAX_S/DBSTEP) * P_SDMAX_S/DFSIZE - P_SDMAX_S/DBSTEP
One packet size	= (one frame size+P_SDMAX_S/DFSTEP)*P_SDMAX_S/DPSIZE - (P_SDMAX_S/DFSTEP)

When address across a block data, the controller will skip block step size and read or write data. When address over a frame data, the next address value will be the current address add the frame step.

10.4.8 Column Copy

For some graphic usage, if there is one picture which is not full screen size, user may need to fill the empty area to fit the screen size. There is one way to stuff image edge data to the empty area called column copy here. This function is supported by SDMA controller illustrated as follow. Column copy mode is the extend function of the index addressing mode.



10.4.9 TCP/IP Checksum Offload (TOE)

The TOE engine used the big endian algorithm to calculate the checksum. It chips the all TCP packet into 2bytes, and sums all the chipped data by using 1's complement adder. If the DRAM data type is little endian, it should be swapped by turning on **EDSW** bit in **P_SDMAX_CTRL**.

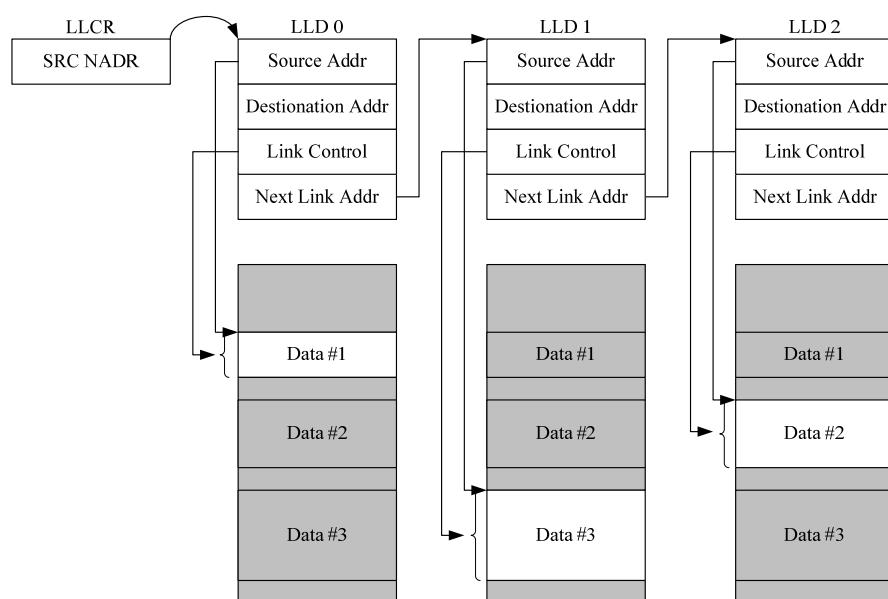
When the TCP payload length is not 2 bytes alignment, for example 103 bytes, the last one byte should be zero padded to be 2 bytes alignment. SDMA doesn't need actually setting the 104th byte to zero in DRAM. It could achieve the zero padding just by "Data Mask" in **SMASK** and **EMASK** of **P_SDMAX_ECFG**.

The running procedure of the TOE:

- step1: Set the base address of the packet (**P_SDMAX_SADR**). It should be 4-Byte alignment.
- step2: Set the pack length (**P_SDMAX_SBSIZE**). It should be 4-byte alignment.
- step3: Set the start byte-mask for the checksum data (**SMASK**).
- step4: Set the end byte-mask for the checksum data (**EMASK**).
- step5: Set the endian mode (**EDSW**) of the ethernet packet.
- step6: Set the addressing mode (**SAM**) to 3'b100.
- step7: Set the channel enable bit (**CHEN = 1**)

10.4.10 Linked List

SDMA supports “Linked List (LL)” type mode. Unlike the transferring mode described above, the transaction information, named linked list descriptor (LLD), stores in the memory, instead of the configure registers. There are four data fields per basic descriptor, source data address, destination data address, link control, and next link address, as shown as following. After the data transferring initialized by the first descriptor, channel controller would fetch the next descriptor from the memory address pointed by “Next Link Addr”, and operate the fetched descriptor immediately.



The address of the first LLD (LLD0) is pointed by the configuration register, P_SDMAX_LLDADDR. After initializing the configuration registers, and touching the LPI bit in P_SDMAX_STATUS, SDMA will fetch the first LLD. The next LLD (LLD1) will be fetched after the first one finish. The data fields format of the link control are shown as follows:

	link control										Data format for the “link control” field.												
Bit	31	30	29	28	27	26	25	24	23	22													
Function	-	-	DAM		SAM		SPEC			NE	L												
Default	0	0	0	0	0	0	0	0	0	0													
Bit	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function(SAM!=2)	-																						
Function(SAM=2)	COLCPY	EMASK	SMASK		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:30]			Reserved.	
[29:28]	DAM	R/W	Destination addressing mode.	0 = Non-increment. 1 = Increment. 2 = Index mode. 3 = Constant fill.
[27:26]	SAM	R/W	Source addressing mode.	0 = Non-increment. 1 = Increment. 2 = Index mode. 3 = Constant fill.
[25:24]	SPEC	R/W	Special addressing mode.	0 = Normal mode. 1 = Column copy.
23	NE	R/W	Next linker address field enable. When this bit is set, it means the “Next Link Addr” field in the LLD is valid.	0 = Disable. 1 = Enable.
22	L	R/W	Last bit. When this bit is set, it means this LLD is last one.	0 = Not Last one. 1 = Last one LLD.
[20:0]	BSIZE	R/W	Source block size. This field is invalid when SAM = 2 (IDX mode).	
[21:20]	COLCPY	R/W	Column copy. This field is valid when SAM = 2 (IDX mode).	1 = Duplicate the LSB. 2 = Duplicate the MSB.
[19:16]	EMASK	R/W	End byte write mask. This field is valid when SAM = 2 (IDX mode).	0001b = Byte 0 mask. 0010b = Byte 1 mask. 0100b = Byte 2 mask. 1000b = Byte 3 mask.
[15:12]	SMASK	R/W	Start byte write mask. This field is valid when SAM = 2 (IDX mode).	0001b = Byte 0 mask. 0010b = Byte 1 mask. 0100b = Byte 2 mask. 1000b = Byte 3 mask.

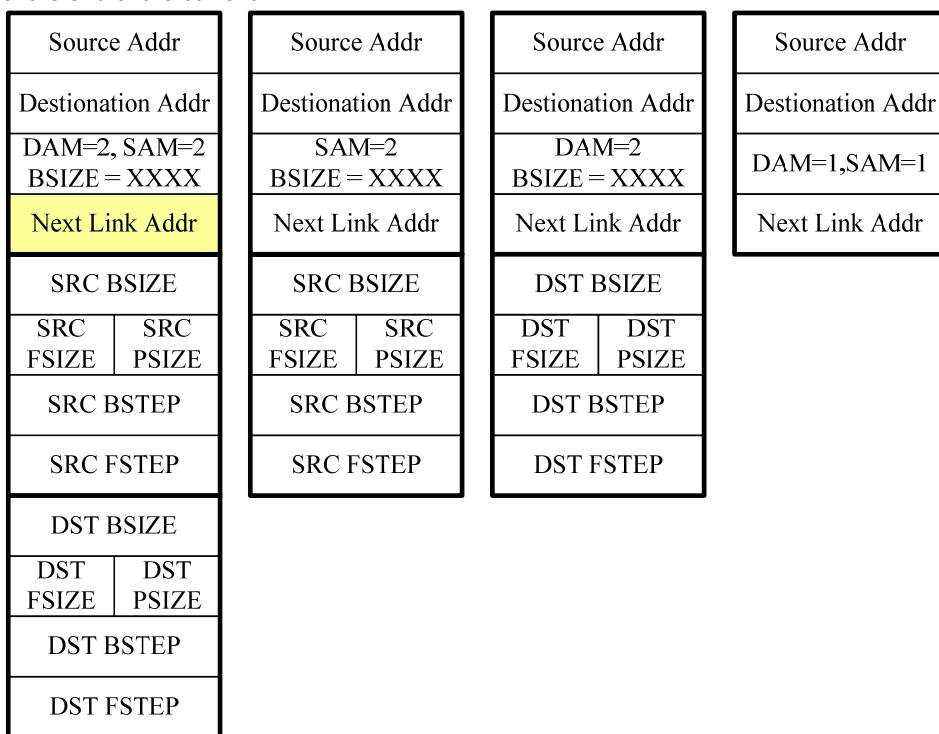
Next Link Addr
Data format for the “Next Link Addr” field.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	NLADR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

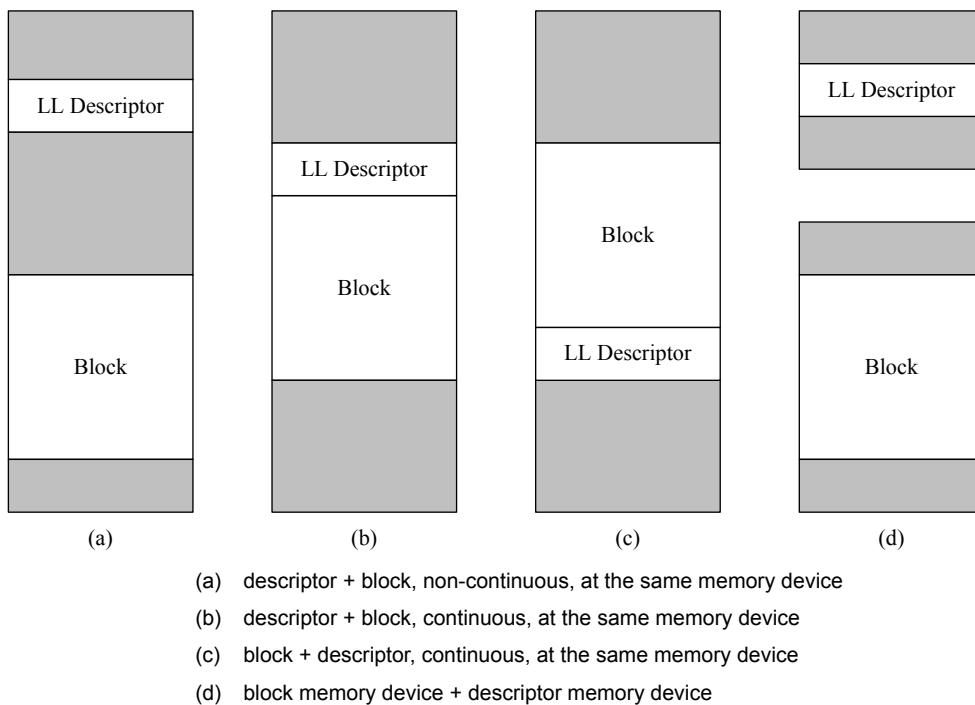
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NLADR								DID				SID			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]	NLADR	R/W	Next Link Address (256 byte alignment is required).	
[7:4]	DID	R/W	Destination ID for DMA request.	15 = Memory.
[3:0]	SID	R/W	Source ID for DMA request.	15 = Memory.

There are two types of methods to define the next LLDS's address. The following figure is illustrated the format of the link control when **SAM** = 2 or **DAM** = 2. If **NE** (next link address enable) is 1, the next LLD's address is the "Next Link Addr" field defined in the LLD. If the NE is 0, the next LLD's address is the next line of the end of the current LLD.



After initializing each LLD, LPI bit in P_SDMAX_STATUS should be set to increase the LLD counter.



The above figure shows four different types of memory allocate for saving descriptor and data block. It's possible saving the descriptor and data block at the same memory device, as shown at (a) ~ (c). Because of the inherent in the SDRAM, non-continuous access would cause worse performance. We suggest allocating descriptor and block side by side as shown at (b) and (c). Another solution is separating the memory as shown at (d), one for fast-small access (ex. descriptor allocates in internal SRAM), and the other for slow-large access (data block in external SDRAM).

10.5 Register Summary

There are 2 DMA channels on this module. The prefix symbol P_SDMA0 and P_SDMA1 are used to indicate channel 0 and channel 1, respectively.

Name	Address	Description
P_SDMA0_ISR	0x92000040	Interrupt status after masking.
P_SDMA1_ISR	0x920000A0	
P_SDMA0_ICR	0x92000044	Interrupt clear register.
P_SDMA1_ICR	0x920000A4	
P_SDMA0_IMR	0x92000048	Interrupt mask register.
P_SDMA1_IMR	0x920000A8	
P_SDMA0IRR	0x9200004C	Interrupt status before masking.
P_SDMA1IRR	0x920000AC	
P_SDMA0_STATUS	0x92000050	SDMA channel status.
P_SDMA1_STATUS	0x920000B0	
P_SDMA0_SUM	0x92000054	Check sum register.
P_SDMA1_SUM	0x920000B4	

Name	Address	Description
P_SDMA0_LLCNT	0x92000058	Link-list counter register.
P_SDMA1_LLCNT	0x920000B8	
P_SDMA0_ECFG	0x9200005C	Extended configuration register.
P_SDMA1_ECFG	0x920000BC	
P_SDMA0_CFG	0x92000060	Configuration register.
P_SDMA1_CFG	0x920000C0	
P_SDMA0_CTRL	0x92000064	SDMA control register.
P_SDMA1_CTRL	0x920000C4	
P_SDMA0_SADR	0x92000068	Source data address.
P_SDMA1_SADR	0x920000C8	
P_SDMA0_DADDR	0x9200006C	Destination data address.
P_SDMA1_DADDR	0x920000CC	
P_SDMA0_LLDADR	0x92000070	Link list descriptor address.
P_SDMA1_LLDADR	0x920000D0	
P_SDMA0_SBSIZE	0x92000078	Source block size.
P_SDMA1_SBSIZE	0x920000D8	
P_SDMA0_DBSIZE	0x9200007C	Destination block size.
P_SDMA1_DBSIZE	0x920000DC	
P_SDMA0_SFSIZE	0x92000080	Source frame size.
P_SDMA1_SFSIZE	0x920000E0	
P_SDMA0_DFSIZE	0x92000084	Destination frame size.
P_SDMA1_DFSIZE	0x920000E4	
P_SDMA0_SPSIZE	0x92000088	Source packet size.
P_SDMA1_SPSIZE	0x920000E8	
P_SDMA0_DPSIZE	0x9200008C	Destination packet size.
P_SDMA1_DPSIZE	0x920000EC	
P_SDMA0_SBSTEP	0x92000090	Source block step size.
P_SDMA1_SBSTEP	0x920000F0	
P_SDMA0_DBSTEP	0x92000094	Destination block step size.
P_SDMA1_DBSTEP	0x920000F4	
P_SDMA0_SFSTEP	0x92000098	Source frame step size.
P_SDMA1_SFSTEP	0x920000F8	
P_SDMA0_DFSTEP	0x9200009C	Destination frame step size.
P_SDMA1_DFSTEP	0x920000FC	

10.6 Register Definition

P_SDMA0_ISR															0x92000040	Interrupt status after masking.			
P_SDMA1_ISR															0x920000A0				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
Function	-	-	-	-	-	-	-	-	DERR	DIDX					-	-			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	-	FIN	SERR	SIDX					LL		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:24]			Reserved.	
23	DERR	R	Destination bus error. Actual error status is at P_SDMAX_STATUS [31:28].	Read 0 = Not occurred. Read 1 = Occurred.
[22:18]	DIDX	R	Destination Index mode status.	00001b = End of half-block. 00010b = End of block. 00100b = End of fame. 01000b = End of packet. 10000b = End of index mode.
[17:9]			Reserved.	
8	FIN	R	DMA finish.	Read 0 = Not occurred. Read 1 = Occurred.
7	SERR	R	Source bus error. Actual error status is at P_SDMAX_STATUS [31:28].	Read 0 = Not occurred. Read 1 = Occurred.
[6:2]	SIDX	R	Source Index mode status.	00001b = End of half-block. 00010b = End of block. 00100b = End of fame. 01000b = End of packet. 10000b = End of index mode.
[1:0]	LL	R	Link-list mode status.	01b = End of LL mode. 10b = LLD is triggered.

P_SDMA0_ICR															0x92000044	Interrupt clear register.			
P_SDMA1_ICR															0x920000A4				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
Function	-	-	-	-	-	-	-	-	CDERR	CDIDX					-	-			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function	-	-	-	-	-	-	-	CFIN	CSERR	CSIDX					CLL			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:24]			Reserved.	
23	CDERR	W	Clear destination bus error. Actual error status is at P_SDMAX_STATUS [31:28].	Write 0 = No effect. Write 1 = Clear.
[22:18]	CDIDX	W	Clear Destination Index mode status.	00001b = Clear end of half-block. 00010b = Clear end of block. 00100b = Clear end of fame. 01000b = Clear end of packet. 10000b = Clear end of index mode.
[17:9]			Reserved.	
8	CFIN	W	Clear DMA finish.	Write 0 = No effect Write 1 = Clear
7	CSERR	W	Clear Source bus error. Actual error status is at P_SDMAX_STATUS [31:28].	Write 0 = No effect Write 1 = Clear
[6:2]	CSIDX	W	Clear Source Index mode status.	00001b = Clear end of half-block. 00010b = Clear end of block. 00100b = Clear end of fame. 01000b = Clear end of packet. 10000b = Clear end of index mode.
[1:0]	CLL	W	Clear Link-list mode status.	01b = Clear end of LL mode. 10b = Clear LLD trigger.

P_SDMA0_IMR
0x92000048
Interrupt mask register.
P_SDMA1_IMR
0x920000A8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Function	-	-	-	-	-	-	-	-	MDERR						MDIDX	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	-	MFIN	MSERR						MSIDX	MLL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:24]			Reserved.	
23	MDERR	R/W	Mask destination bus error interrupt. Actual error status is at P_SDMAX_STATUS [31:28].	Write 0 = Enable interrupt. Write 1 = Mask interrupt.

Bit	Function	Type	Description	Condition
[22:18]	MDIDX	R/W	Mask destination Index mode status interrupt.	00001b = End of half-block. 00010b = End of block. 00100b = End of fame. 01000b = End of packet. 10000b = End of index mode.
[17:9]			Reserved.	
8	MFIN	R/W	Mask DMA finish interrupt.	Write 0 = Enable interrupt. Write 1 = Mask interrupt.
7	MSERR	R/W	Mask source bus error interrupt. Actual error status is at P_SDMAX_STATUS [31:28].	Write 0 = Enable interrupt. Write 1 = Mask interrupt.
[6:2]	MSIDX	R/W	Mask source Index mode status interrupt.	00001b = End of half-block. 00010b = End of block. 00100b = End of fame. 01000b = End of packet. 10000b = End of index mode.
[1:0]	MLL	R/W	Mask link-list mode status interrupt.	01b = Mask end of LL mode. 10b = Mask LLD trigger.

P_SDMA0IRR
0x9200004C
Interrupt status before masking.
P_SDMA1IRR
0x920000AC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	DERR			DIDX	-	-		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	FIN	SERR			SIDX		LL		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:24]			Reserved.	
23	DERR	R	Destination bus error. Actual error status is at P_SDMAX_STATUS [31:28].	Read 0 = Not occurred. Read 1 = Occurred.
[22:18]	DIDX	R	Destination Index mode status.	00001b = End of half-block. 00010b = End of block. 00100b = End of fame. 01000b = End of packet. 10000b = End of index mode.
[17:9]			Reserved.	
8	FIN	R	DMA finish.	Read 0 = Not occurred. Read 1 = Occurred.

Bit	Function	Type	Description	Condition
7	SERR	R	Source bus error. Actual error status is at P_SDMAX_STATUS [31:28].	Read 0 = Not occurred. Read 1 = Occurred.
[6:2]	SIDX	R	Source Index mode status.	00001b = End of half-block. 00010b = End of block. 00100b = End of fame. 01000b = End of packet. 10000b = End of index mode.
[1:0]	LL	R	Link-list mode status.	01b = End of LL mode. 10b = LLD is triggered.

P_SDMA0_STATUS

0x92000050

SDMA channel status.

P_SDMA1_STATUS

0x920000B0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	ERR				-	-	-	-	-	-	-	-	-	PAU	-	ACT
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	LPI	-	PAR	STOP	CHEN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:28]	ERR	R	Error status.	0 = No error. 1 = FIFO overflow. 2 = FIFO underflow. 3 = Write bus error. 4 = Read bus error.
[27:20]			Reserved.	
19			Reserved.	
18	PAU	R	Channel pause status.	0 = Channel not in pause. 1 = Channel is paused.
17			Reserved.	
16	ACT	R	Channel active status	0 = Channel is in idle state. 1 = Channel is active.
[15:5]			Reserved.	
4	LPI	W	List-list pool Increase.	Write 0 or 1 will increase LL pool.
3			Reserved.	
2	PAR	W	Pause and resume. Writing 1 when channel is running will pause the channel. Writing 1 when channel is passed will resume the channel. Writing 0 may cause un-predictable result.	

Bit	Function	Type	Description	Condition
1	STOP	R/W	Setting this bit will stop the transfer. Writing 0 may cause un-predictable result.	
0	CHEN	R/W	Setting this bit will start the transfer. Writing 0 may cause un-predictable result.	

P_SDMA0_SUM
0x92000054
Check sum register.
P_SDMA1_SUM
0x920000B4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CKSUM															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]			Reserved.	
[15:0]	CKSUM	R	Check sum.	

P_SDMA0_LLCNT
0x92000058
Link-list counter register.
P_SDMA1_LLCNT
0x920000B8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	LLCNT												
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:13]			Reserved.	
[12:0]	LLCNT	R	List list pool counter. When CHEN in P_SDMAX_STATUS is set, this counter will be reset immediately. Setting LPI bit in P_SDMAX_STATUS will increase this counter and when the one LLD is finish will decrease it.	

P_SDMA0_ECFG **0x9200005C** **Extended configuration register.**

P_SDMA1_ECFG **0x920000BC**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	COLCPY		EMASK				SMASK			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:10]			Reserved.	
[9:8]	COLCPY	R/W	Column copy.	01b = Duplicate LSB. 10b = Duplicate MSB. Other values will no effect.
[7:4]	EMASK	R/W	End byte write mask.	0001b = Byte 0 mask. 0010b = Byte 1 mask. 0100b = Byte 2 mask. 1000b = Byte 3 mask.
[3:0]	SMASK	R/W	Start byte write mask.	0001b = Byte 0 mask. 0010b = Byte 1 mask. 0100b = Byte 2 mask. 1000b = Byte 3 mask.

P_SDMA0_CFG **0x92000060** **Configuration register.**

P_SDMA1_CFG **0x920000C0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	SPEC	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	DBST	SBST	DSIZE	SSIZE	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:29]			Reserved.	
[28:27]	SPEC	R/W	Special addressing mode.	00b = Normal addressing mode. 01b = Column copy. 10b = Reserved 11b = Reserved
[26:14]			Reserved.	
[13:12]	DBST	R/W	Destination burst size	0 = Single. 1 = 4 burst. 2 = 8 burst. 3 = 16 burst.
[11:10]	SBST	R/W	Source burst size	0 = Single. 1 = 4 burst. 2 = 8 burst. 3 = 16 burst.
[9:8]	DSIZE	R/W	Destination data width.	0 = 8 bits. 1 = 16 bits. 2 = 32 bits.
[7:6]	SSIZE	R/W	Source data width.	0 = 8 bits. 1 = 16 bits. 2 = 32 bits.
[5:0]			Reserved.	

P_SDMA0_CTRL
0x92000064
SDMA control register.
P_SDMA1_CTRL
0x920000C4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DAM	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DAM	SAM			DID				SID				EDSW	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:18]			Reserved.	
[17:15]	DAM	R/W	Destination addressing mode.	0 = Non-increment mode. 1 = Increment mode. 2 = Index mode.

Bit	Function	Type	Description	Condition
[14:12]	SAM	R/W	Source addressing mode.	0 = Non-increment mode. 1 = Increment mode. 2 = Index mode. 3 = Constant fill. 4 = TOE. 6 = Link-list.
[11:7]	DID	R/W	Destination ID fo DMA request.	1 = Memory stick. 2 = Reserved 3 = UART (TX). 4 = UART (RX). 5 = SD0. 6 = SD1. 7 = Reserved 8 = APBDMA1 channel 0. 9 = APBDMA1 channel 1. 10 = SSP/SPI (RX). 11 = SSP/SPI (TX). 12 = APBDMA0 channel 0. 13 = APBDMA0 channel 1. 14 = APBDMA0 channel 2. 15 = Memory Others fo reserved.
[6:2]	SID	R/W	Source ID fo DMA request.	1 = Memory stick. 2 = Reserved 3 = UART (TX). 4 = UART (RX). 5 = SD0. 6 = SD1. 7 = Reserved 8 = APBDMA1 channel 0. 9 = APBDMA1 channel 1. 10 = SSP/SPI (RX). 11 = SSP/SPI (TX). 12 = APBDMA0 channel 0. 13 = APBDMA0 channel 1. 14 = APBDMA0 channel 2. 15 = Memory Others fo reserved.
1	EDSW	R/W	Endian switch. Endian swith for big to little or little to big	

Bit	Function	Type	Description	Condition
0			Reserved.	

P_SDMA0_SADR 0x92000068 Source data address.
P_SDMA1_SADR 0x920000C8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	SADR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SADR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	SADR	R/W	Source start address. Address must be aligned according the source data width SSIZE in P_SDMAX_CFG . For example, source data width is set as 4 bytes (32bits), so source start address must be 4bytes alignment.	

P_SDMA0_DADR 0x9200006C Destination data address.
P_SDMA1_DADR 0x920000CC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	DADR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DADR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	DADR	R/W	Destination start address. Address must be aligned according the destination data width DSIZE in P_SDMAX_CFG .	

P_SDMA0_LLDADR 0x92000070 Link list descriptor address.
P_SDMA1_LLDADR 0x920000D0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	LLDADR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LLDADR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description													Condition
[31:0]	LLDADR	R/W	Link list descriptor address. This register is used for link-list mode only. For link list mode, user must set the first LLD address to this register. It must be set 4bytes alignment.													

P_SDMA0_SBSIZE **0x92000078** **Source block size.**

P_SDMA1_SBSIZE **0x920000D8**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	SBSIZE			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SBSIZE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description													Condition
[31:20]			Reserved.													
[19:0]	SBSIZE	R/W	Source block size (unit: byte). Block size must be times of source data width SSIZE in P_SDMAX_CFG . For example, source data width is set as 4 bytes (32bits), so source block size must be multiple of 4. Source block size has to equal to destination block size. For index mode, it must be multiple of 4.													

P_SDMA0_DBSIZE **0x9200007C** **Destination block size.**

P_SDMA1_DBSIZE **0x920000DC**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	DBSIZE			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DBSIZE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:20]			Reserved.	
[19:0]	DBSIZE	R/W	Destination block size (unit: byte). Block size must be times of destination data width DSIZE in P_SDMAX_CFG . Destination block size must equal to source block size. For index mode, it must be multiple of 4.	

P_SDMA0_SFSIZE
0x92000080
Source frame size.
P_SDMA1_SFSIZE
0x920000E0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SFSIZE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																SFSIZE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SDMA0_SFSIZE
0x92000080
Source frame size.
P_SDMA1_SFSIZE
0x920000E0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DFSIZE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																DFSIZE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SDMA0_SFSIZE
0x92000080
Source packet size.
P_SDMA1_SFSIZE
0x920000E0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SPSIZE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SDMA0_SPSIZE
0x92000088
Source packet size.
P_SDMA1_SPSIZE
0x920000E8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SPSIZE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SPSIZE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:20]			Reserved.	
[19:0]	SPSIZE	R/W	Source packet size (unit: number of frame).	

P_SDMA0_DPSIZE 0x9200008C Destination packet size.

P_SDMA1_DPSIZE 0x920000EC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16													
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DPSIZE														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DPSIZE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:20]			Reserved.	
[19:0]	DPSIZE	R/W	Destination packet size (unit: number of frame).	

P_SDMA0_SBSTEP 0x92000090 Source block step size.

P_SDMA1_SBSTEP 0x920000F0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16													
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SBSTEP														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SBSTEP															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:20]			Reserved.	
[19:0]	SBSTEP	R/W	Source block step size (unit: byte). Block step size must be times of source data width SSIZE in P_SDMAX_CFG . For example, source data width is set as 4 bytes (32bits), so source block step size must be multiple of 4.	

P_SDMA0_DBSTEP 0x92000094 Destination block step size.

P_SDMA1_DBSTEP 0x920000F4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	DBSTEP			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DBSTEP															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description														Condition
[31:20]			Reserved.														
[19:0]	DBSTEP	R/W	Destination block step size (unit: byte). Block step size must be times of source data width DSIZE in P_SDMAX_CFG .														

P_SDMA0_SFSTEP 0x92000098 Source frame step size.

P_SDMA1_SFSTEP 0x920000F8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SFSTEP			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SFSTEP															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description														Condition
[31:20]			Reserved.														
[19:0]	SFSTEP	R/W	Source fram step size (unit: byte). Frame step size must be times of source data width SSIZE in P_SDMAX_CFG . For example, source data width is set as 4 bytes (32bits), so source frame step size must be multiple of 4.														

P_SDMA0_DFSTEP 0x9200009C Destination frame step size.

P_SDMA1_DFSTEP 0x920000FC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DFSTEP		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DFSTEP															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:20]			Reserved.	
[19:0]	DFSTEP	R/W	Destination fram step size (unit: byte). Frame step size must be times of destination data width DSIZE in P_SDMAX_CFG .	

11 XDMA

11.1 Introduction

The XS Direct Memory Access (XDMA) is part of the GPL32900 platform. The XDMA transfers data from a source to a destination without any Core intervention. This transfer is carried out in either an untouched data format or in a restructured data format, depending on the requirements of the particular application. A simple example of a data transfer is copying a contiguous memory space from one source module to a destination module in a non-restructured manner. The XDMA contains up-to 4 independent programmable channels that support 12 different contexts (data transfers) for the XDMA operation.

11.2 Features

- 4 configurable XDMA channels.
- Three-dimensional XDMA transfer.
- Access to the entire CEVA-X internal data memory.
- 8/16/32/64-bit data transfer support.
- Configurable burst length.
- Programmable channel priority.
- Programmable source and destination addresses with a post-modification option.
- Interrupt generation.
- Pause and resume operations.
- Chaining-channels operating mode.
- Linked list-transfer operating mode.
- XDMA manager support.
- Halt on breakpoint.
- Eight-stage memory buffer FIFO.

12 PPU Introduction

12.1 General Description

A high-performance Picture Processing Unit (PPU) is built in GP12 with a QVGA/VGA/D1 virtual 3D engine to accelerate the graphic processing. Up to four Text layers with 1024 sprites and 4096 extended sprites are available; plus, maximum of 1024 palette index colors can be displayed on screen. In addition to these graphical features, other peripherals such as TFT LCD, STN LCD, CMOS sensor, and light pen interlaces are all supported in GP12 for variety of TV game requirements.

12.2 Significant Features

TV Encoder:

- Supports multi-standards: NTSC-(M, N, J)/PAL-(B, D, G, H, I, M, N, Nc).
- Supports QVGA/VGA/D1 composite video signal output.
- Supports both interlace and non-interlace operations.
- Supports CCIR-601/656 output.
- Built-in color bar adjustment.
- Built-in 10-bit Video DAC.
- Saturation/Hue/Brightness/Contrast adjustable.
- Edge enhancement.
- Horizontal and vertical screen position adjustment.
- Luminance delay adjustment.

Screen Structure:

- Supports both line-base and frame-base architectures.
- Supports line + frame base operation for TV and TFT work at the same time.
- Supports dual frame base operation for TV and TFT work at the same time.
- TV Horizontal 320 X Vertical 240 (QVGA)
- TV Horizontal 640 X Vertical 480 (VGA)
- TV Horizontal 640 X Vertical 240 (HVGA)
- TV Horizontal 720 X Vertical 480 (D1)
- Supports arbitrary output resolution with in 1920x1024.
- For TFT output application, it supports two kinds of resolution: 320x240 and 640x480.
- For STN output application, it supports only one resolution: 320x240.
- Eight depth layers: 4 layers for TEXT and 4 layers for sprites.
- Four programmable windows with priority control for all TEXT layers and sprites.

TEXT:

- Total of 4 Text Layers.
- Each TEXT supports color 4/16/64/256/32768/65536/RGBG/YUYV color modes.
- Each TEXT supports character mode and bitmap mode.
- Each TEXT supports 8 kinds of TEXT size: 512x256, 512x512, 1024x512, 1024x1024, 2048x1024, 2048x2048, 4096x2048, and 4096x4096.
- Each TEXT supports 64-level blending function.
- Each TEXT supports horizontal movement function.
- Each TEXT supports vertical compression function.
- Each TEXT support rotating and zoom functions.
- Each TEXT support transparent character mode.
- Only TEXT1 supports horizontal compression function.
- Only TEXT3 supports virtual 3D function with Y compression.

Sprite:

- Maximum 1024 (internal) + 4096 (extended) sprites can be used at the same time.
- Each sprite contains 1 character data.
- Each sprite supports color 4/16/64/256/32768/65536/8 + 6 bits blending/ARGB4444/RGBG/YUYV/RGBA color modes.
- Each sprite supports 64-level zoom-in/out function.
- Each sprite supports 64-level rotate function.
- Each sprite supports 4-level mosaic function.
- Each sprite supports 64-level blending function.
- Two kinds of coordinate mode are supported.
- Each sprite supports virtual 3D texture mapping function.
- Each sprite supports alpha-channel function under color 256 mode.
- Rectangle color dither sprites support internal 512 and extended 4096 sprites totally which do not consume bandwidth during painting.
- Supports alpha-channel + blending function at the same time.
- Supports bi-linear interpolation on each sprite.
- Supports 2 bits fraction on the coordinate of sprites.

Character:

- Sixteen character's sizes supported for TEXTs from TEXT size 512x256 to 1024x1024 and sprites.
8x8, 8x16, 8x32, 8x64,
16x8, 16x16, 16x32, 16x64,
32x8, 32x16, 32x32, 32x64,

64x8, 64x16, 64x32, 64x64.

- Additional sixteen character's sizes supported for TEXTs with TEXT size from 2048x1024 to 4096x4096 and sprites
 - 32x32, 32x64, 32x128, 32x256,
 - 64x32, 64x64, 64x128, 64x256,
 - 128x32, 128x64, 128x128, 128x256,
 - 256x32, 256x64, 256x128, 256x256.
- Eight color modes supported: 4/16/64/256/32768/65536/8 + 6 bits blending/
ARGB4444/YUYV/RGBG/RGBA colors
- Each character supports horizontal/vertical flip.
- Supports both related and direct addressing modes for each character.

Others:

- Embedded 2 random-number generators.
- Embedded geometry transformation engine (GTE).
- Built-in ROM saving mode.
- Supports hardware fade effect for entire screen.
- Supports RGB color mapping table for TFT-LCD display.
- Built-in DMA for sprite-data/palette-data/rotate-data movement.
- Supports both 16-bit (1T/RGB555) and 25-bit (1T/RGB888) palette index colors in 4/16/64/256 color modes.
- Supports single palette mode (TEXT and sprite use the same palette).
- Supports dual palette mode (TEXT and sprite use different palette).
- Supports the sharing function of 1024 palette index colors (1T/RGB555) for both TEXT and sprite.

PPU Interrupt Sources:

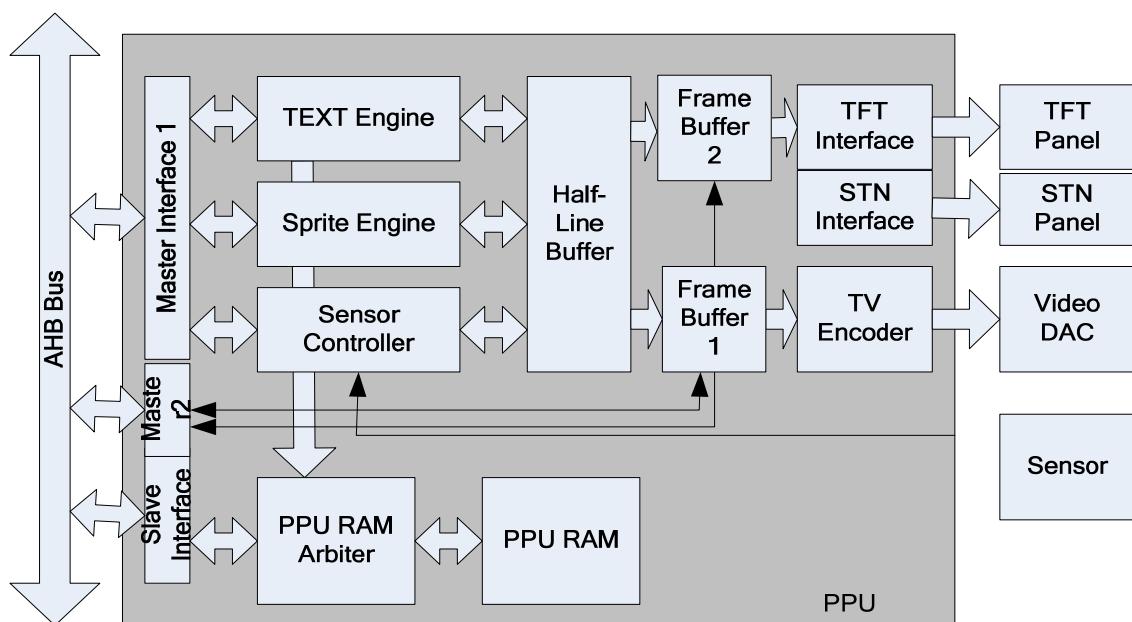
- PPU vertical blanking IRQ
- Video timing IRQ
- DMA transfer done IRQ
- Palette write error IRQ
- TEXT engine under-run IQR
- Sprite engine under-run IQR
- Sensor frame end IRQ
- Sensor motion detect frame end IRQ
- Sensor position hit IRQ
- Sensor motion detect FIFO under-run IRQ
- TV frame buffer FIFO under-run IRQ
- TV vertical blanking IRQ

- TFT-LCD frame buffer under-run IRQ
- TFT-LCD vertical blanking IRQ
- PPU horizontal blanking IRQ
- Sensor receive FIFO under-run IQR

Light Pen:

- Built-in light pen interface

12.3 Block Diagram



12.4 PPU Registers

Name	Address	Description
P_PPU_TEXT3_X_POSITION	0x93020000	TEXT 3 X position register
P_PPU_TEXT3_Y_POSITION	0x93020004	TEXT 3 Y position register
P_PPU_TEXT3_X_OFFSET	0x93020008	TEXT 3 X offset register
P_PPU_TEXT3_Y_OFFSET	0x9302000C	TEXT 3 Y offset register
P_PPU_TEXT3_ATTRIBUTE	0x93020010	TEXT 3 attribute register
P_PPU_TEXT3_CTRL	0x93020014	TEXT 3 control register
P_PPU_TEXT3_N_PTR	0x93020018	TEXT 3 number array pointer register
P_PPU_TEXT3_A_PTR	0x9302001C	TEXT 3 attribute array pointer register
P_PPU_TEXT4_X_POSITION	0x93020020	TEXT 4 X position register
P_PPU_TEXT4_Y_POSITION	0x93020024	TEXT 4 Y position register
P_PPU_TEXT4_X_OFFSET	0x93020028	TEXT 4 X offset register

Name	Address	Description
P_PPU_TEXT4_Y_OFFSET	0x9302002C	TEXT 4 Y offset register
R_PPU_TEXT4_ATTRIBUTE	0x93020030	TEXT 4 attribute register
P_PPU_TEXT4_CTRL	0x93020034	TEXT 4 control register
P_PPU_TEXT4_N_PTR	0x93020038	TEXT 4 number array pointer register
P_PPU_TEXT4_A_PTR	0x9302003C	TEXT 4 attribute array pointer register
P_PPU_TEXT1_X_POSITION	0x93020040	TEXT 1 X position register
P_PPU_TEXT1_Y_POSITION	0x93020044	TEXT 1 Y position register
P_PPU_TEXT1_X_OFFSET	0x93020340	TEXT 1 X offset register
P_PPU_TEXT1_Y_OFFSET	0x93020344	TEXT 1 Y offset register
R_PPU_TEXT1_ATTRIBUTE	0x93020048	TEXT 1 attribute register
P_PPU_TEXT1_CTRL	0x9302004C	TEXT 1 control register
P_PPU_TEXT1_N_PTR	0x93020050	TEXT 1 number array pointer register
P_PPU_TEXT1_A_PTR	0x93020054	TEXT layer 1 attribute array pointer register
P_PPU_TEXT2_X_POSITION	0x93020058	TEXT 2 X position register
P_PPU_TEXT2_Y_POSITION	0x9302005C	TEXT 2 Y position register
P_PPU_TEXT2_X_OFFSET	0x93020350	TEXT 2 X offset register
P_PPU_TEXT2_Y_OFFSET	0x93020354	TEXT 2 Y offset register
R_PPU_TEXT2_ATTRIBUTE	0x93020060	TEXT 2 attribute register
P_PPU_TEXT2_CTRL	0x93020064	TEXT 2 control register
P_PPU_TEXT2_N_PTR	0x93020068	TEXT 2 number array pointer register
P_PPU_TEXT2_A_PTR	0x9302006C	TEXT 2 attribute array pointer register
P_PPU_VCOMP_VALUE	0x93020070	Vertical compression value register
P_PPU_VCOMP_OFFSET	0x93020074	Vertical compression offset register
P_PPU_VCOMP_STEP	0x93020078	Vertical compression step register
P_PPU_TEXT1_SEGMENT	0x93020080	TEXT 1 segment address register
P_PPU_TEXT2_SEGMENT	0x93020084	TEXT 2 segment address register
P_PPU_SPRITE_SEGMENT	0x93020088	Sprite segment address register
P_PPU_TEXT3_SEGMENT	0x9302008C	TEXT 3 segment address register
P_PPU_TEXT4_SEGMENT	0x93020090	TEXT 4 segment address register
P_PPU_TEXT1_COSINE	0x93020348	TEXT 1 cosine register
P_PPU_TEXT1_SINE	0x9302034C	TEXT 1 sine register
P_PPU_TEXT2_COSINE	0x93020358	TEXT 2 cosine register
P_PPU_TEXT2_SINE	0x9302035C	TEXT 2 sine register
P_PPU_TEXT4_COSINE	0x930200A0	TEXT 4 cosine register
P_PPU_TEXT4_SINE	0x930200A4	TEXT 4 sine register
P_PPU_BLENDING	0x930200A8	TEXTs 4-level blending control register
P_PPU_FADE_CTRL	0x930200C0	Fade effect control register

Name	Address	Description
P_PPU_IRQTMV	0x930200D8	Vertical hit IRQ control register
P_PPU_IRQTMH	0x930200DC	Horizontal hit IRQ control register
P_PPU_LINE_COUNTER	0x930200E0	TV line counter register
P_PPU_LIGHTPEN_CTRL	0x930200E4	Light pen control register
P_PPU_PALETTE_CTRL	0x930200E8	Palette control register
P_PPU_LPHPOSITION	0x930200F8	Light pen horizontal position register
P_PPU_LPPOSITION	0x930200FC	Light pen vertical position register
P_PPU_Y25D_COMPRESS	0x93020104	Y compress parameter under virtual 3D mode
P_PPU_SPRITE_CTRL	0x93020108	Sprite control register
P_PPU_WINDOW0_X	0x93020120	Window 0 X control register
P_PPU_WINDOW0_Y	0x93020124	Window 0 Y control register
P_PPU_WINDOW1_X	0x93020128	Window 1 X control register
P_PPU_WINDOW1_Y	0x9302012C	Window 1 Y control register
P_PPU_WINDOW2_X	0x93020130	Window 2 X control register
P_PPU_WINDOW2_Y	0x93020134	Window 2 Y control register
P_PPU_WINDOW3_X	0x93020138	Window 3 X control register
P_PPU_WINDOW3_Y	0x9302013C	Window 3 Y control register
P_PPU_IRQ_EN	0x93020188	PPU IRQ enable register
P_PPU_IRQ_STATUS	0x9302018C	PPU IRQ status register
P_PPU_SPRITE_DMA_SOURCE	0x930201C0	Sprite RAM DMA source address register
P_PPU_SPRITE_DMA_TARGET	0x930201C4	Sprite RAM DMA target address register
P_PPU_SPRITE_DMA_NUMBER	0x930201C8	Sprite RAM DMA transfer number register
P_PPU_HB_CTRL	0x930201CC	Horizontal blank control register
P_PPU_HB_GO	0x930201D0	Horizontal blank active register
P_TV_FBI_ADDR	0x930201E0	Frame buffer pointer for TV
P_TFT_FBI_ADDR	0x9302033C	Frame buffer pointer for TFT
P_PPU_FBO_ADDR	0x930201E8	Frame buffer pointer for PPU output
P_PPU_FB_GO	0x930201F0	Start PPU in frame buffer mode
P_PPU_BLD_COLOR	0x930201F4	Transparent color mode register.
P_PPU_MISC	0x930201F8	PPU Misc Control register.
P_PPU_ENABLE	0x930201FC	PPU enable register
P_PPU_SPRITE_X0	0x93020300	Sprite Coordinate Convert Input X0
P_PPU_SPRITE_Y0	0x93020304	Sprite Coordinate Convert Input Y0
P_PPU_SPRITE_X1	0x93020308	Sprite Coordinate Convert Input X1
P_PPU_SPRITE_Y1	0x9302030C	Sprite Coordinate Convert Input Y1
P_PPU_SPRITE_X2	0x93020310	Sprite Coordinate Convert Input X2
P_PPU_SPRITE_Y2	0x93020314	Sprite Coordinate Convert Input Y2

Name	Address	Description
P_PPU_SPRITE_X3	0x93020318	Sprite Coordinate Convert Input X3
P_PPU_SPRITE_Y3	0x9302031C	Sprite Coordinate Convert Input Y3
P_PPU_SPRITE_W0	0x93020320	Sprite SRAM Word 0
P_PPU_SPRITE_W1	0x93020324	Sprite SRAM Word 1
P_PPU_SPRITE_W2	0x93020328	Sprite SRAM Word 2
P_PPU_SPRITE_W3	0x9302032C	Sprite SRAM Word 3
P_PPU_SPRITE_W4	0x93020330	Sprite SRAM Word 4
P_PPU_EXTENDSPRITE_CONTROL	0x93020334	Extend sprite control register.
P_PPU_EXTENDSPRITE_ADDR	0x93020338	Extend sprite start address register.
P_PPU_RGB_OFFSET	0x9302037C	PPU special effect RGB offset register.
P_RANDOM0	0x93020380	Random number 0
P_RANDOM1	0x93020384	Random number 1
P_GTE_DIVOF	0xF60040A8	Division overflow register
P_GTE_DIVA	0xF60040A0	Dividend register
P_GTE_DIVB	0xF60040A4	Divisor register
P_GTE_DIVO	0xF60040AC	Quotient register
P_GTE_DIVR	0xF60040B0	Residue register
P_FREE_SIZE	0x9302036C	Free size control register.
P_DEF_PARA	0x93020370	De-flicker parameter control.
P_TFT_3D_Offset	0x930203A8	Frame buffer pointer offset for TFT (in 3D mode).
P_TFTFBI_Addr_R	0x930203AC	Frame buffer pointer for TFT Right field (in 3D mode).

12.5 TV Registers

Name	Address	Description
P_TV1_CTRL	0x930200F0	TV Control Register
P_TV1_SATURATION	0x93020200	TV Saturation Register
P_TV1_HUE	0x93020204	TV Hue Register
P_TV1_BRIGHTNESS	0x93020208	TV Brightness Register
P_TV1_SHARPNESS	0x9302020C	TV Sharpness Register
P_TV1_Y_GAIN	0x93020210	TV Y Gain Register
P_TV1_Y_DELAY	0x93020214	TV Y Delay Register
P_TV1_V_POSITION	0x93020218	TV Vertical Position Register
P_TV1_H_POSITION	0x9302021C	TV Horizontal Position Register
P_TV1_VIDEO_DAC	0x93020220	TV Video DAC Control Register

12.6 Sensor Registers

Name	Address	Description
P_CSI1_TG_CTRL0	0x93020240	Sensor timing generator control register 0
P_CSI1_TG_CTRL1	0x93020244	Sensor timing generator control register 1
P_CSI1_TG_HLSTART	0x93020248	Sensor horizontal latch start register
P_CSI1_TG_HEND	0x9302024C	Sensor horizontal end register
P_CSI1_TG_VL0START	0x93020250	Sensor field 0 vertical latch start register
P_CSI1_MD_FBADDR	0x93020254	Motion detect buffer start address register
P_CSI1_TG_VEND	0x93020258	Sensor vertical end register
P_CSI1_TG_HSTART	0x9302025C	Sensor horizontal start register
P_CSI1_MD_RGBL	0x93020260	Motion detect RGB/YUV register low word
P_CSI1_SEN_CTRL	0x93020264	Sensor attribute control register
P_CSI1_TG_BSUPPER	0x93020268	Blue screen upper limit control register
P_CSI1_TG_BSLOWER	0x9302026C	Blue screen lower limit control register
P_CSI1_MD_RGBH	0x93020270	Motion detect RGB/YUV register high word
P_CSI1_MD_CTRL	0x93020274	Motion detect control register
P_CSI1_TG_FBSADDR	0x93020278	Sensor frame buffer start address register
P_CSI1_TG_VL1START	0x93020280	Sensor field 1 vertical latch start register
P_CSI1_TG_HWIDTH	0x93020284	Sensor horizontal width control register
P_CSI1_TG_VHEIGHT	0x93020288	Sensor vertical height control register
P_CSI1_TG_CUTSTART	0x9302028C	Sensor cut region start address register
P_CSI1_TG_CUTSIZE	0x93020290	Sensor cut size register
P_CSI1_TG_VSTART	0x93020294	Sensor vertical start register
P_CSI1_TG_HRATIO	0x9302029C	Sensor horizontal compress ratio control register
P_CSI1_TG_VRATIO	0x930202A0	Sensor vertical compress ratio control register
P_CSI1_MD_HPOS	0x930202A4	Motion detect horizontal hit position register
P_CSI1_MD_VPOS	0x930202A8	Motion detect vertical hit position register

Name	Address	Description
P_CSI1_SEN2_IRQSTS	0x930202B8	Sensor 2 IRQ status register.
P_CSI1_SEN2_IRQEN	0x930202BC	Sensor 2 IRQ enable register.
P_CSI1_TG2_CTRL1	0x930202C0	Sensor 2 timing generator control register 1.
P_CSI1_TG2_CTRL2	0x930202C4	Sensor 2 timing generator control register 1.
P_CSI1_TG2_HLSTART	0x930202C8	Sensor 2 horizontal latch start register.
P_CSI1_TG2_HEND	0x930202CC	Sensor 2 horizontal end register.
P_CSI1_TG2_VL0START	0x930202D0	Sensor 2 field 0 vertical latch start register.
P_CSI1_TG2_VEND	0x930202D4	Sensor 2 vertical end register.

Name	Address	Description
P_CSI1_TG2_HSTART	0x930202D8	Sensor 2 horizontal start register.
P_CSI1_TG2_FBSADDR	0x930202DC	Sensor 2 frame buffer start address register.
P_CSI1_TG2_VL1START	0x930202E0	Sensor 2 field 1 vertical latch start register.
P_CSI1_TG2_H_WIDTH	0x930202E4	Sensor 2 horizontal width control register.
P_CSI1_TG2_V_WIDTH	0x930202E8	Sensor 2 vertical width control register.
P_CSI1_TG2_CUTSTART	0x930202EC	Sensor 2 cut region start address register.
P_CSI1_TG2_CUTWIDTH	0x930202F0	Sensor 2 cut size register.
P_CSI1_TG2_VSTART	0x930202F4	Sensor 2 vertical start register.
P_CSI1_TG2_H_RATIO	0x930202F8	Sensor 2 horizontal compress ratio control register.
P_CSI1_TG2_V_RATIO	0x930202FC	Sensor 2 vertical compress ratio control register.

12.7 PPU Special Effect RAM

12.7.1 Horizontal Movement Control RAM

Name	Address	Description
P_PPU_TEXT_H_OFFSET0	0x93020400	Horizontal movement register of line 0
P_PPU_TEXT_H_OFFSET1	0x93020404	Horizontal movement register of line 1
...
P_PPU_TEXT_H_OFFSET238	0x930207B8	Horizontal movement register of line 238
P_PPU_TEXT_H_OFFSET239	0x930207BC	Horizontal movement register of line 239

12.7.2 Horizontal Compression Control RAM

(Valid only when TEXT3 virtual 3D is disabled)

Name	Address	Description
P_PPU_TEXT_HCMP_VALUE0	0x93020800	Horizontal compression control register of line 0
P_PPU_TEXT_HCMP_VALUE1	0x93020804	Horizontal compression control register of line 1
...
P_PPU_TEXT_HCMP_VALUE238	0x93020BB8	Horizontal compression control register of line 238
P_PPU_TEXT_HCMP_VALUE239	0x93020BBC	Horizontal compression control register of line 239

12.7.3 TEXT3 Virtual 3D Control Registers

Name	Address	Description
P_PPU_TEXT3_COS0	0x93020800	TEXT 3 Cosine value of line 0
P_PPU_TEXT3_SIN0	0x93020804	TEXT 3 Sine value of line 0

Name	Address	Description
P_PPU_TEXT3_COS1	0x93020808	TEXT 3 Cosine value of line 1
P_PPU_TEXT3_SIN1	0x9302080C	TEXT 3 Sine value of line 1
...
P_PPU_TEXT3_COS238	0x93020F70	TEXT 3 Cosine value of line 238
P_PPU_TEXT3_SIN238	0x93020F74	TEXT 3 Sine value of line 238
P_PPU_TEXT3_COS239	0x93020F78	TEXT 3 Cosine value of line 239
P_PPU_TEXT3_SIN239	0x93020F7C	TEXT 3 Sine value of line 239

12.8 PPU Palette RAM

There are totally 4 palette bank RAMs.

Name	Address	Description
P_PPU_PALETTE_RAM0_0	0x93021000	Palette RAM 0 word 0
P_PPU_PALETTE_RAM0_1	0x93021004	Palette RAM 0 word 1
...
P_PPU_PALETTE_RAM0_254	0x930213F8	Palette RAM 0 word 254
P_PPU_PALETTE_RAM0_255	0x930213FC	Palette RAM 0 word 255
P_PPU_PALETTE_RAM1_0	0x93021400	Palette RAM 1 word 0
P_PPU_PALETTE_RAM1_1	0x93021404	Palette RAM 1 word 1
...
P_PPU_PALETTE_RAM1_254	0x930217F8	Palette RAM 1 word 254
P_PPU_PALETTE_RAM1_255	0x930217FC	Palette RAM 1 word 255
P_PPU_PALETTE_RAM2_0	0x93021800	Palette RAM 2 word 0
P_PPU_PALETTE_RAM2_1	0x93021804	Palette RAM 2 word 1
...
P_PPU_PALETTE_RAM2_254	0x93021BF8	Palette RAM 2 word 254
P_PPU_PALETTE_RAM2_255	0x93021BFC	Palette RAM 2 word 255
P_PPU_PALETTE_RAM3_0	0x93021C00	Palette RAM 3 word 0
P_PPU_PALETTE_RAM3_1	0x93021C04	Palette RAM 3 word 1
...
P_PPU_PALETTE_RAM3_254	0x93021FF8	Palette RAM 3 word 254
P_PPU_PALETTE_RAM3_255	0x93021FFC	Palette RAM 3 word 255

12.9 PPU Sprite Attribute RAM

The sprite RAM is used to control each sprite's attribute. Each sprite requires 16 bytes parameter and total of 1024 sprites is built inside the GP12; so, there is $16 \times 1024 = 16$ KBytes sprite RAM in total. When the PPU function is disabled, this RAM can be used as program RAM for general purpose usage.

Name	Address	Description
P_PPU_SPRITE0_CHARNUM	0x93022000	Character number of sprite 0
P_PPU_SPRITE0_X_POSITION	0x93022002	X position and rotate angle of sprite 0
P_PPU_SPRITE0_Y_POSITION	0x93022004	Y position and zoom parameter of sprite 0
P_PPU_SPRITE0_ATTRIBUTE0	0x93022006	Attribute of sprite 0
P_PPU_SPRITE0_ATTRIBUTE1	0x93022008	Attribute 1 of sprite 0
P_PPU_SPRITE0_X1	0x9302200A	X1 of sprite 0
P_PPU_SPRITE0_X2	0x9302200C	X2 of sprite 0
P_PPU_SPRITE0_X3	0x9302200E	X3 of sprite 0
P_PPU_SPRITE1_CHARNUM	0x93022010	Character number of sprite 1
P_PPU_SPRITE1_X_POSITION	0x93022012	X position and rotate angle of sprite 1
P_PPU_SPRITE1_Y_POSITION	0x93022014	Y position and zoom parameter of sprite 1
P_PPU_SPRITE1_ATTRIBUTE0	0x93022016	Attribute of sprite 1
P_PPU_SPRITE1_ATTRIBUTE1	0x93022018	Attribute 1 of sprite 1
P_PPU_SPRITE1_X1	0x9302201A	X1 of sprite 1
P_PPU_SPRITE1_X2	0x9302201C	X2 of sprite 1
P_PPU_SPRITE1_X3	0x9302201E	X3 of sprite 1
...
P_PPU_SPRITE1022_CHARNUM	0x93025FE0	Character number of sprite 1022
P_PPU_SPRITE1022_X_POSITION	0x93025FE2	X position and rotate angle of sprite 1022
P_PPU_SPRITE1022_Y_POSITION	0x93025FE4	Y position and zoom parameter of sprite 1022
P_PPU_SPRITE1022_ATTRIBUTE0	0x93025FE6	Attribute of sprite 1022
P_PPU_SPRITE1022_ATTRIBUTE1	0x93025FE8	Attribute 1 of sprite 1022
P_PPU_SPRITE1022_X1	0x93025FEA	X1 of sprite 1022
P_PPU_SPRITE1022_X2	0x93025FEC	X2 of sprite 1022
P_PPU_SPRITE1022_X3	0x93025FEE	X3 of sprite 1022
P_PPU_SPRITE1023_CHARNUM	0x93025FF0	Character number of sprite 1023
P_PPU_SPRITE1023_X_POSITION	0x93025FF2	X position and rotate angle of sprite 1023
P_PPU_SPRITE1023_Y_POSITION	0x93025FF4	Y position and zoom parameter of sprite 1023
P_PPU_SPRITE1023_ATTRIBUTE0	0x93025FF6	Attribute of sprite 1023
P_PPU_SPRITE1023_ATTRIBUTE1	0x93025FF8	Attribute 1 of sprite 1023
P_PPU_SPRITE1023_X1	0x93025FFA	X1 of sprite 1023
P_PPU_SPRITE1023_X2	0x93025FFC	X2 of sprite 1023
P_PPU_SPRITE1023_X3	0x93025FFE	X3 of sprite 1023

The sprite extend RAM is used to control the extend attribute of each sprite. Each sprite require 4 bytes parameter and total 1024 sprites are built in GP12, so there are total $4 \times 1024 = 4$ kbytes sprite extend RAM.

Name	Address	Description
P_Sprite0_Ext	0x6000	Extend attribute of sprite 0.
P_Sprite1_Ext	0x6004	Extend attribute of sprite 1.
P_Sprite2_Ext	0x6008	Extend attribute of sprite 2.
.....		
P_Sprite1023_Ext	0x6FFE	Extend attribute of sprite 1023.

12.10 PPU Color Mapping RAM

The color mapping RAM is used to control the output color of PPU for TFT-LCD display. Programmer can adjust the RGB mapping curve individually to correct color while displaying on TFT-LCD.

Name	Address	Description
P_PPU_TFT_COLOR_MAP_B_0	0x93027000	B mapping RAM 0
P_PPU_TFT_COLOR_MAP_B_1	0x93027004	B mapping RAM 1
...
P_PPU_TFT_COLOR_MAP_B_254	0x930273F8	B mapping RAM 254
P_PPU_TFT_COLOR_MAP_B_255	0x930273FC	B mapping RAM 255
P_PPU_TFT_COLOR_MAP_G_0	0x93027400	G mapping RAM 0
P_PPU_TFT_COLOR_MAP_G_1	0x93027404	G mapping RAM 1
...
P_PPU_TFT_COLOR_MAP_G_254	0x930277F8	G mapping RAM 254
P_PPU_TFT_COLOR_MAP_G_255	0x930277FC	G mapping RAM 255
P_PPU_TFT_COLOR_MAP_R_0	0x93027800	R mapping RAM 0
P_PPU_TFT_COLOR_MAP_R_1	0x93027804	R mapping RAM 1
...
P_PPU_TFT_COLOR_MAP_R_254	0x93027BF8	R mapping RAM 254
P_PPU_TFT_COLOR_MAP_R_255	0x93027BFC	R mapping RAM 255



13 PPU Control

13.1 Screen Structure

GP12 supports four resolutions for TV (+TFT) applications.

320x240 (QVGA mode): Frame rate 60 fps in NTSC and 50 fps in PAL. This mode can supports TV or/and TFT application. Line base or frame base mode can be used.

640x240 (HVGA mode): Frame rate 60 fps in NTSC and 50 fps in PAL. This mode can supports only TV application. Line base or frame base mode can be used.

640x480 (VGA mode): Frame rate 30 fps in NTSC and 25 fps in PAL. This mode can support TV or/and TFT application. Line base or frame base mode can be used.

720x480 (D1 mode): Frame rate 30 fps in NTSC and 25 fps in PAL. This mode supports TV application only. Only the frame base mode can be used.

Free mode: The output resolution can be adjusted arbitrary under this mode, but only frame buffer mode is supported under this mode.

The following register describes the control of screen mode.

VGA_EN is 0 => QVGA mode, for TV or TFT application only.

VGA_EN is 1 => VGA mode, for TV (+TFT) application.

VGA_NOINTL is 0 => VGA interlace mode; even line and odd line have different values, valid only when VGA_EN is “1”

VGA_NOINTL is 1 => VGA non-interlace mode (HVGA); even line and odd line have the same value, valid only when VGA_EN is "1"

Note: The value setting here only affects the PPU, not TV encoder. The real output mode of the TV signal is controlled by the TV encoder's control register.

The following table lists out the control of PPU output resolutions for TV, TFT-LCD, and STN-LCD.

VGA_EN	TV Resolution (TFT_SIZE)		TFT Resolution (TFT_SIZE)		STN Resolution (TFT_SIZE)	
0	0	320x240 (line or frame base)	0	320x240 (line or frame base)	0	320x240 (line or frame base)
	1	Not support	1	640x480 (line or frame base)	1	640x480 (line or frame base)
	2	Not support	2	480x234 (line or frame base)	2	Not support
	3	Not support	3	480x272 (line or frame base)	3	Not support
	4	720x480 (frame base only)	4	720x480 (frame base only)	4	Not support
	5	Not support	5	800x480 (frame base only)	5	Not support
	6	Not support	6	800x600 (frame base only)	6	Not support
	7	Not support	7	1024x768 (frame base only)	7	Not support
1	0	640x480 ¹ (line or frame base)	0	320x240 ¹ (line base only)	0	320x240 ¹ (line base only)
	1	Not support	1	Not support	1	Not support
	2	Not support	2	Not support	2	Not support
	3	Not support	3	Not support	3	Not support
	4	Not support	4	Not support	4	Not support
	5	Not support	5	Not support	5	Not support
	6	Not support	6	Not support	6	Not support
	7	Not support	7	Not support	7	Not support

Note 1: The image can't be shown on TV and TFT/STN at the same time in this condition.

When TFT/STN is enabled(TV is disabled), the resolution will be 320x240 but it can only support line base mode.

When TV is enabled(TFT/STN is disabled), the resolution will be 640x480 and it can support both frame/line base mode.

GPL32900A supports arbitrary resolution for PPU's output, which is called free mode. When programmer wants to use the free mode, the following register must be configured correctly to enter the free mode. The free mode will be entered automatically when the FREE_HWIDTH and FREE_VWIDTH are set as non zero value.

P_FREE_SIZE
0x036C
PPU Size Control Register

Bit	31	30	29	28	27	26	25	24
Function	INTL	-						FREE_HWIDTH[10:8]
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
FREE_HWIDTH[7:0]							

0	0	0	0	0	0	0	0
-							
15	14	13	12	11	10	9	8
FREE_VWIDTH[10:8]							

7	6	5	4	3	2	1	0
FREE_VWIDTH[7:0]							

INTL is 0 => Non-interlace free mode. (For TFT type application)

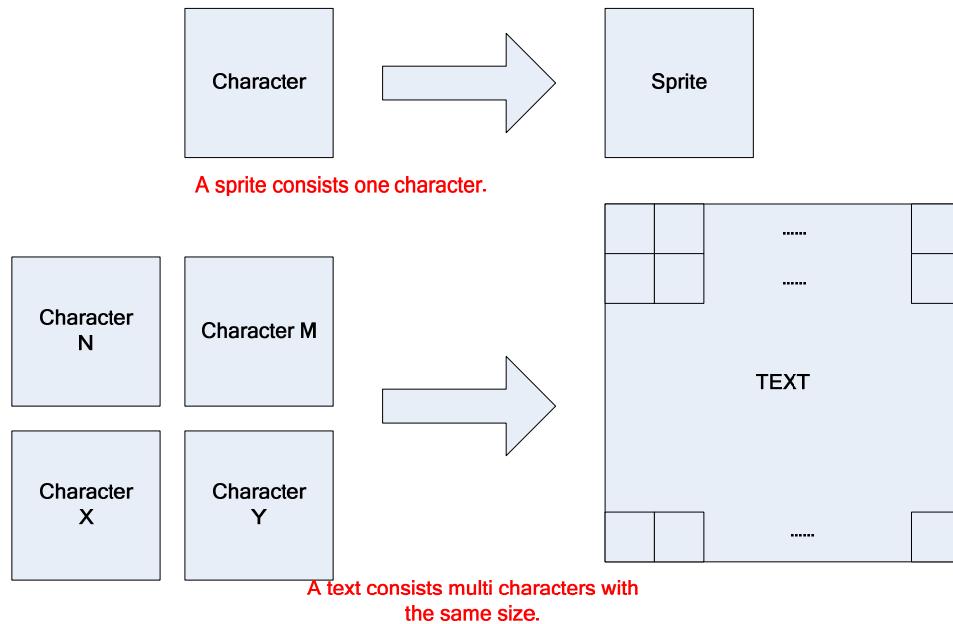
INTL is 1 => Interlace free mode. (For TV type application)

FREE_HWIDTH: Horizontal width selection. Range 16~1920. This value must be a multiply of 16 (i.e. the last 4 bits is 0)

FREE_VWIDTH: Vertical width selection. Range 1~1024.

13.2 Character

In each mode, the screen may have up to 4 TEXTs, 1024 internal sprites and 4096 extended sprites. Both TEXTs and sprites can be formed by character. A character is a rectangular block in which the vertical and horizontal size can be defined independently among 8, 16, 32 and 64 pixels or 32, 64, 128 and 256 pixels when the large TEXT N size is used. The following figure shows the relationships between character and TEXT sprite.



The TEXT N is capable of expressing a background picture. The TEXT N size is configurable as follows: 512x256, 512x512, 1024x512, 1024x1024, 2048x1024, 2048x2048, 4096x2048 and 4096x4096. The sprite is a small object, representing a roll in a scene, which can be placed anywhere on the screen and it is able to perform rotation, zoom-in, and zoom-out functions on GP12. Both TEXTs and sprites have separate configurable attribute, or called depth, which is used to express the top-down relation.

A character is the basic unit of TEXTs and sprites under character mode. The following table shows the attributes of a character.

Attribute	In TEXT	In Sprite
Color mode	Controlled by P_PPU_TEXTN_ATTRIBUTE	Controlled by P_PPU_SPRITEN_ATTRIBUTE0
	4/16/64/256/32768/65536/8+6 bits blending/ARGB4444/RGBG/YUY/ RGBA	4/16/64/256/32768/65536/8+6 bits blending/ARGB4444/RGBG/YUYV/ RGBA
Palette Selection	Controlled by P_PPU_TEXTN_ATTRIBUTE or Attribute RAM	Controlled by P_PPU_SPRITEN_ATTRIBUTE0 and P_PPU_SPRITEN_ATTRIBUTE1
Depth	Controlled by P_PPU_TEXTN_ATTRIBUTE	Controlled by P_PPU_SPRITEN_ATTRIBUTE0
	Layer 0, 2, 4, 6	Layer 1, 3, 5, 7
Horizontal/Vertical Size	Controlled by P_PPU_TEXTN_ATTRIBUTE	Controlled by P_PPU_SPRITEN_ATTRIBUTE0
	For TEXT size 512x256~1024x1024 {8, 16, 32, 64 (H)}X{8, 16, 32, 64 (V)} For TEXT size 2048x1024~4096x4096 {32, 64, 128, 256 (H)}X{32, 64, 128, 256 (V)}	For SP_LS is 0 {8, 16, 32, 64 (H)}X{8, 16, 32, 64 (V)} For SP_LS is 1 {32, 64, 128, 256 (H)}X{32, 64, 128, 256 (V)}
Horizontal/Vertical Flip	Controlled by P_PPU_TEXTN_ATTRIBUTE or Attribute RAM	Controlled by P_PPU_SPRITEN_ATTRIBUTE0
	No flip, H flip, V flip, HV flip	No flip, H flip, V flip, HV flip
Blending Effect	Controlled by P_PPU_TEXTN_CTRL	Controlled by P_PPU_SPRITEN_ATTRIBUTE1
	64-level: 1 ~ 63	64-level: 1 ~ 63
Rotate	Entire TEXT layer at any angle (0° ~ 360°)	Controlled by P_PPU_SPRITEN_X_POSITION
		0°, 5.625°, ..., 354.375° (64-level)
Zoom In/Out	Entire TEXT layer at any factor	Controlled by P_PPU_SPRITEN_Y_POSITION
		64 scale from 1/32 to 8.75 times
Mosaic Effect	Not support	Controlled by P_PPU_SPRITEN_ATTRIBUTE1
		For SP_LS is 0 No mosaic, 2x2, 4x4, 8x8 mosaic For SP_LS is 1 No mosaic, 8x8, 16x16, 32x32 mosaic

Attribute	In TEXT	In Sprite
Character Number	Stored in Number RAM and/or Attribute RAM	Stored in P_PPU_SPRITEN_CHARNUM and P_PPU_SPRITEN_ATTRIBUTE1
Virtual 3D Mode	TEXT 3 only	Each sprite support virtual 3D mode.
Bi-linear interpolation	yes	Yes

13.2.1 Color Mode

Color mode defines the number of bits per pixel. GP12 supports 8 color modes: color 4 (2-bit/pixel), color 16 (4-bit/pixel), color 64 (6-bit/pixel), color 256 (8-bit/pixel), color 32768 (15-bit/pixel), color 65536 (16-bit/pixel), RGBG (32-bit/two pixels), and YUYV (32-bit/two pixels) for both TEXT and sprite. The color register control is depicted as follows:

P_PPU_TEXTN_ATT 0x93020048 0x93020060 0x93020010 TEXT N Attribute Register								
RIBUTE 0x93020030 (N=1,2,3,4)								
Bit	23	22	21	20	19	18	17	
Function	TXN_PB[1:0]	TXN_EFF	-	TXN_WINDOW	TXN_SIZE[2]			
Default	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	
	TXN_SIZE[1:0]	TXN_DEPTH	TXN_PALETTE					
	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	
	TXN_VS	TXN_HS	TXN_FLIP	TXN_COLOR				
	0	0	0	0	0	0	0	

P_PPU_TEXTN_CTRL 0x9302004C 0x93020064 0x93020014 TEXT N Control Register							
0x93020034 (N=1,2,3,4)							
Bit	15	14	13	12	11	10	9
Function	TXN_BLDLVL					TXN_BLDMODE	TXN_BLD
Default	0	0	0	0	0	0	0
	7	6	5	4	3	2	1
	TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP
	0	0	0	0	0	0	0

P_PPU_SPRITEN_ATTRIBUTE0								0x93022006+(16*N)								Sprite N Attribute Register 0								
Bit	15	14	13	12	11	10	9	8																
Function	SPB_PB[1]	SPN_BLD	SPN_DEPTH				SPN_PALETTE																	
Default	-	-	-	-	-	-	-	-																
	7	6	5	4	3	2	1	0																
	SPN_VS				SPN_HS				SPN_FLIP				SPN_COLOR											
	-	-	-	-	-	-	-	-																

The following table shows the relationship between register setting and real color mode.

TEXT			
TXN_RGBM	TXN_COLOR	Color Bits	
0	2'b00	2	From Palette
	2'b01	4	From Palette
	2'b10	6	From Palette
	2'b11	8	From Palette
1	2'b00	TXN_PALETTE	Color Bits
		4'b0000	15
		4'b1001	8 + 6 bits blending
		4'b1011	ARGB4444
		Others	Reserved
	2'b01	16	Pattern match
		RGBG (TXT_RGBA is 0)	
		32 bits RGBA (TXT_RGBA is 1)	
		YUYV	

Sprites			
SPN_COLOR	Color Bits		Transparent
2'b00	2		From Palette
2'b01	4		From Palette
2'b10	6		From Palette
2'b11	SPN_PALETTE[3:0]	Color Bits	
	4'b0000	8	From Palette
	4'b0010	8	From Palette
	4'b0100	8	From Palette
	4'b0110	8	From Palette
	4'b0001	15	From bit 15 of data
	4'b0011	16	Pattern match
	4'b0101	RGBG (SPR_RGBA is 0)	Pattern match
		32 bits RGBA (SPR_RGBA is 1)	From A part in data
	4'b0111	YUYV	Pattern match
	4'b1001	8 + 6 bits blending	From Palette
	4'b1011	ARGB4444	From A part in data

To use the TEXT and sprite's RGBA mode, the following register must be set to enable this function.

P_PPU_Misc 0x930201F8 PPU Misc Control Register								
Bit	23	22	21	20	19	18	17	16
Function			TFT_3D					SBMP_MODE
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
TV_LB		TFTVTQ	DELGO	INTP_MODE	NEW_CMP	TXT_ALPHA	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ADDR_X2	DUAL_BLD			SPR_RGB	TXT_RGB	FB_LOCK	
0	0	0	0	0	0	0	0

TXT_RGB is 0 => Disable TEXT's RGBA mode, TEXT only has RGBG mode.

TXT_RGB is 1 => Enable TEXT's RGBA mode, RGBG mode will be disabled.

SPR_RGB is 0 => Disable sprite's RGBA mode, TEXT only has RGBG mode.

SPR_RGB is 1 => Enable sprite's RGBA mode, RGBG mode will be disabled.

TXT_ALPHA is 0 => Disable TXT alpha channel/ARGB4444 function.

TXT_ALPHA is 0 => Disable TXT alpha channel/ARGB4444 function.

For RGB565, RGBG and YUYV color mode, the transparent color is defined by the P_BLD_COLOR register. When color is match the data defined in the P_BLD_Color register, the pixel will be treated as transparent on the screen. Following is the definition of P_BLD_Color register.

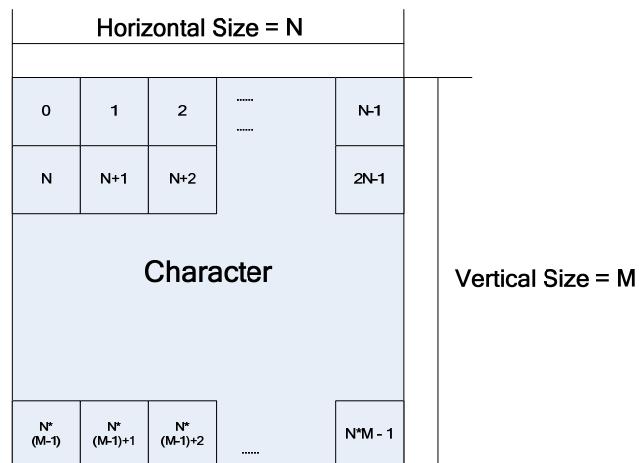
P_PPU_BLD_COLOR 0x930201F4 Blending Color Register								
Bit	31	30	29	28	27	26	25	24
Function				-		BLDMODE	TRAEN	
Default	0	0	0	0	0	0	0	0
 23 22 21 20 19 18 17 16 BLD_Color[23:16]								
0	0	0	0	0	0	0	0	0
 15 14 13 12 11 10 9 8 BLD_Color[15:8]								
0	0	0	0	0	0	0	0	0
 7 6 5 4 3 2 1 0 BLD_Color[7:0]								
0	0	0	0	0	0	0	0	0

The following table shows the transparent operation method under each mode.

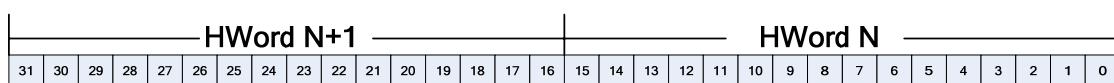
Color Mode	Transparent Condition	Notes
RGB565	RGB565[15:0] == BLD_Color[15:0] & TRAEN ==1	BLD_Color [23:16] must be 0.
RGBG	RGB[23:0] == BLD_Color[23:0] & TRAEN ==1	Since every two pixels share the same R & B, so the transparent function works in the Unit of every 2 pixels.
YUYV	When BLDMODE is 0 YUV[23:0] == BLD_Color[23:0] & TRAEN ==1	Since every two pixels share the same U & V, so the transparent function works in the Unit of every 2 pixels.
	When BLDMODE is 1 YUV[23:16] == BLD_Color[23:16] & TRAEN ==1	Since every pixel has its own Y, so this mode can be used to be a transparent Method for each pixel.

Since there is only one P_BLD_COLOR register, so only one of the RGB565/RGBG/YUYV mode can be selected to have transparent mode.

The following diagram is the pixel order in a character.



Under various color modes, the bit's sequences in a 16-bit data (half-word) are different; the following figure shows the bit's sequences in a word between various color modes.





2 bits color mode	<table border="1"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td colspan="2">Pixel 0</td><td colspan="2">Pixel 1</td><td colspan="2">Pixel 2</td><td colspan="2">Pixel 3</td><td colspan="2">Pixel 4</td><td colspan="2">Pixel 5</td><td colspan="2">Pixel 6</td><td colspan="2">Pixel 7</td></tr></table>	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	Pixel 0		Pixel 1		Pixel 2		Pixel 3		Pixel 4		Pixel 5		Pixel 6		Pixel 7																																																	
7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8																																																																		
Pixel 0		Pixel 1		Pixel 2		Pixel 3		Pixel 4		Pixel 5		Pixel 6		Pixel 7																																																																			
4 bits color mode	<table border="1"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td colspan="4">Pixel 0</td><td colspan="4">Pixel 1</td><td colspan="4">Pixel 2</td><td colspan="4">Pixel 3</td></tr></table>	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	Pixel 0				Pixel 1				Pixel 2				Pixel 3																																																			
7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8																																																																		
Pixel 0				Pixel 1				Pixel 2				Pixel 3																																																																					
6 bits color mode	<table border="1"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td colspan="4">Pixel 0</td><td colspan="4">Pixel 1</td><td colspan="4">Pixel 2</td><td colspan="4">Pixel 3</td></tr><tr><td colspan="4">Pixel 4</td><td colspan="4">Pixel 5</td><td colspan="4">Pixel 6</td><td colspan="4">Pixel 7</td></tr><tr><td colspan="4">Pixel 8</td><td colspan="4">Pixel 9</td><td colspan="4">Pixel 10</td><td colspan="4">Pixel 11</td></tr><tr><td colspan="4">Pixel 12</td><td colspan="4">Pixel 13</td><td colspan="4">Pixel 14</td><td colspan="4">Pixel 15</td></tr></table>	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	Pixel 0				Pixel 1				Pixel 2				Pixel 3				Pixel 4				Pixel 5				Pixel 6				Pixel 7				Pixel 8				Pixel 9				Pixel 10				Pixel 11				Pixel 12				Pixel 13				Pixel 14				Pixel 15			
7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8																																																																		
Pixel 0				Pixel 1				Pixel 2				Pixel 3																																																																					
Pixel 4				Pixel 5				Pixel 6				Pixel 7																																																																					
Pixel 8				Pixel 9				Pixel 10				Pixel 11																																																																					
Pixel 12				Pixel 13				Pixel 14				Pixel 15																																																																					
8 bits color mode	<table border="1"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td colspan="4">Pixel 0</td><td colspan="4">Pixel 1</td><td colspan="4">Pixel 2</td><td colspan="4">Pixel 3</td></tr></table>	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	Pixel 0				Pixel 1				Pixel 2				Pixel 3																																																			
7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8																																																																		
Pixel 0				Pixel 1				Pixel 2				Pixel 3																																																																					
15 bits color mode	<table border="1"><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>T</td><td colspan="4">R[4:0]</td><td colspan="4">G[4:0]</td><td colspan="4">B[4:0]</td><td colspan="3">Pixel N</td></tr></table>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	T	R[4:0]				G[4:0]				B[4:0]				Pixel N																																																		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																		
T	R[4:0]				G[4:0]				B[4:0]				Pixel N																																																																				
16 bits color mode	<table border="1"><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="4">Pixel 0</td><td colspan="4">Pixel 1</td><td colspan="4">Pixel 2</td><td colspan="4">Pixel 3</td></tr><tr><td colspan="4">Pixel 4</td><td colspan="4">Pixel 5</td><td colspan="4">Pixel 6</td><td colspan="4">Pixel 7</td></tr></table>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Pixel 0				Pixel 1				Pixel 2				Pixel 3				Pixel 4				Pixel 5				Pixel 6				Pixel 7																																			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																		
Pixel 0				Pixel 1				Pixel 2				Pixel 3																																																																					
Pixel 4				Pixel 5				Pixel 6				Pixel 7																																																																					

When RGBG or YUYV mode is selected, the pixel data becomes 32-bit align. The data structure depends on the settings in YUV_TYPE.

The following table shows the data structure on each type.

Mode	YUV_TYPE[1:0]	Data Structure
RGBG	0	{B[7:0], G1[7:0], R[7:0], G0[7:0]}
	1	{G1[7:0], B[7:0], G0[7:0], R[7:0]}
	2	{R[7:0], G1[7:0], B[7:0], G0[7:0]}
	3	{G1[7:0], R[7:0], G0[7:0], B[7:0]}
YUYV	0	{V[7:0], Y1[7:0], U[7:0], Y0[7:0]}
	1	{Y1[7:0], V[7:0], Y0[7:0], U[7:0]}
	2	{U[7:0], Y1[7:0], V[7:0], Y0[7:0]}
	3	{Y1[7:0], U[7:0], Y0[7:0], V[7:0]}

The YUV_TYPE [2] is used to control if the U/V is signed or unsigned under YUYV mode.

0: UV is unsigned.

1: UV is signed.

When color 8 bits + 6 bits blending mode is selected, the sprite can have alpha channel function, i.e. each pixel can have its own blending factor. The following figure shows the data format under this mode.



The EN bit is used to turn on the blending function of this pixel. When EN is “0”, this pixel is not blending.

The color of this pixel is stored in the palette RAM with index PixelN [7:0]. **Under this mode, the blending factor stored in the sprite RAM becomes useless.**

An additional kind of alpha channel mode is supported in GP12, which is called ARGB4444, the following figure shows the data format under this mode.



Under this mode, the blending factor become only 4 bits which is range from 0 (transparent) to 15 (non-transparent), the the RGB value has 12 bits which can show up-to 4096 colors.

Under these two alpha channel modes, the blending factor stored in the sprite RAM becomes useless. If programmer wishes to enable the blending factor at this time, the following register must be set to enable the alpha channel and blending factor at the same time.

P_PPU_Misc	0x930201F8								PPU Misc Control Register							
Bit	23	22	21	20	19	18	17	16								
Function	TFT_3D								SBMP_MODE							
Default	0	0	0	0	0	0	0	0								

15	14	13	12	11	10	9	8
TV_LB		TFTVTQ	DELGO	INTP_MODE	NEW_CMP	TXT_ALPHA	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ADDR_X2	DUAL_BLD			SPR_RGBA	TXT_RGBA	FB_LOCK	
0	0	0	0	0	0	0	0

DUAL_BLD is 0 => blending factor is useless under alpha channel mode.

DUAL_BLD is 1 => Blending factor is useful under alpha channel mode.

13.2.2 Palette Control

When the color mode of a character is set to one of the following color modes, 4, 16, 64 or 256 colors, the hardware will use the color stored in the palette RAM for screen display. The palette parameter is stored in the following register/RAM.

P_PPU_TEXTN_ATT 0x93020048 0x93020060 0x93020010 TEXT N Attribute Register

RIBUTE 0x93020030 (N=1,2,3,4)							
Bit	23	22	21	20	19	18	17
Function	TXN_PB[1:0]	TXN_EFF	-	TXN_WINDOW	TXN_SIZE[2]		
Default	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
TXN_SIZE[1:0]	TXN_DEPTH	TXN_PALETTE					
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TXN_VS	TXN_HS	TXN_FLIP				TXN_COLOR	
0	0	0	0	0	0	0	0

Note: When TXN_REGM of TEXT N control register is configured as "0", the TEXT engine will use the parameter stored in attribute RAM as the palette parameter.

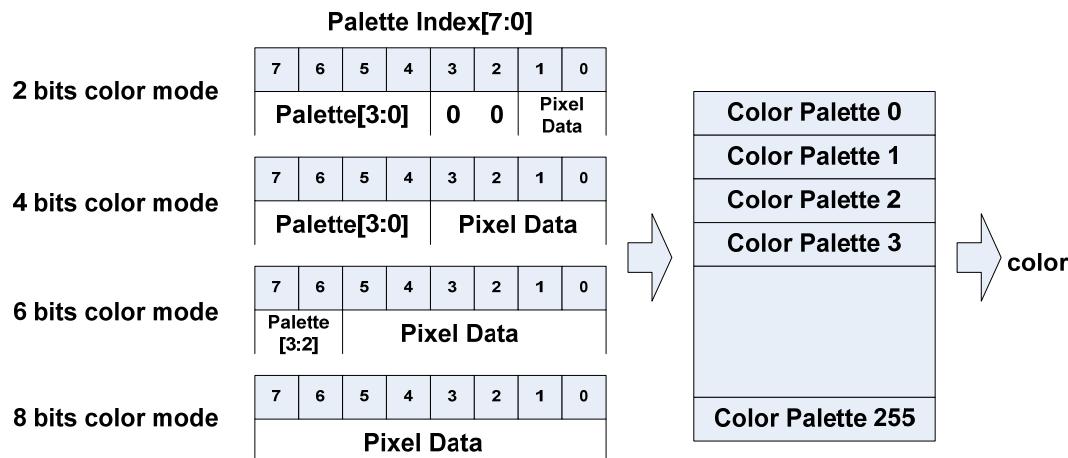
P_PPU_SPRITEN_ATTRIBUTE0 0x93022006+(16*N) Sprite N Attribute Register 0

Bit	15	14	13	12	11	10	9	8
Function	SPB_PB[1]	SPN_BLD	SPN_DEPTH		SPN_PALETTE			
Default	-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
SPN_VS	SPN_HS	SPN_FLIP				SPN_COLOR	
-	-	-	-	-	-	-	-

P_PPU_SPRITEN_ATTRIBUTE1								0x93022008+(16*N)								Sprite N Attribute Register 1								
Bit	15	14	13	12	11	10	9	8	SPN_MOSAIC								SPN_BLDLVL							
Function																								
Default	-	-	-	-	-	-	-	-																
	7	6	5	4	3	2	1	0	SPN_PB[0]								SPN_CHARNUM[22:16]							
	-	-	-	-	-	-	-	-																

The following diagram shows how the hardware uses the palette register and pattern data to find the index of palette table.



By programming PAL_TYPE [1:0] of palette control register (P_PPU_PALETTE_CTRL - 0x930200E8), programmer can determine to use the same or different palette RAM on TEXT and sprite character. Two kinds of palette RAM (6 bits or 25 bits) can be chosen by the same register.

P_PPU_PALETTE_CTRL 0x930200E8								Palette Control Register																
Bit	7	6	5	4	3	2	1	0	P1024								PAL_TYPE							
Function	-				-																			
Default	0	0	0	0	0	0	0	0																

The following table shows the configuration of palette RAM according to palette control register.

P1024	PAL_TYPE	TEXT Palette RAM			Sprite Palette RAM	
0	2'b00	Bank 0	16 bits Palette, uses palette RAM 0	Bank 0	16 bits Palette, uses palette RAM 0	
		Bank 1	16 bits Palette, uses palette RAM 0	Bank 1	16 bits Palette, uses palette RAM 0	
		Bank 2	16 bits Palette, uses palette RAM 2	Bank 2	16 bits Palette, uses palette RAM 2	
		Bank 3	16 bits Palette, Uses palette RAM 2	Bank 3	16 bits Palette, uses palette RAM 2	
0	2'b01	Bank 0	16 bits Palette, uses palette RAM 0	Bank 0	16 bits Palette, uses palette RAM 1	
		Bank 1	16 bits Palette, uses palette RAM 0	Bank 1	16 bits Palette, uses palette RAM 1	
		Bank 2	16 bits Palette, uses palette RAM 2	Bank 2	16 bits Palette, uses palette RAM 3	
		Bank 3	16 bits Palette, uses palette RAM 2	Bank 3	16 bits Palette, uses palette RAM 3	
0	2'b10	25 bits Palette, uses palette RAM 0 and 2			25 bits palette, uses palette RAM 0 and 2	
0	2'b11	25 bits Palette, uses palette RAM 0 and 2			25 bits palette, uses palette RAM 1 and 3	
1	Don't Care	Bank 0	16 bits Palette, uses palette RAM 0	Bank 0	16 bits Palette, uses palette RAM 0	
		Bank 1	16 bits Palette, uses palette RAM 1	Bank 1	16 bits Palette, uses palette RAM 1	
		Bank 2	16 bits Palette, uses palette RAM 2	Bank 2	16 bits Palette, uses palette RAM 2	
		Bank 3	16 bits Palette, uses palette RAM 3	Bank 3	16 bits Palette, uses palette RAM 3	

Under 16 bits palette mode (i.e. PAL_TYPE [1] is 0), PPU supports palette bank selection for each TEXT layer or sprite. The following registers are used to control the palette bank.

P_PPU_TEXTN_ATT 0x93020048 0x93020060 0x93020010 TEXT N Attribute Register

RIBUTE		0x93020030							(N=1,2,3,4)	
Bit	23	22	21	20	19	18	17	16		
Function	TXN_PB[1:0]		TXN_EFF			-	TXN_WINDOW		TXN_SIZE[2]	
Default	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
TXN_SIZE[1:0]		TXN_DEPTH			TXN_PALETTE		

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
TXN_VS		TXN_HS			TXN_FLIP		TXN_COLOR

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

P_PPU_SPRITEN_ATTRIBUTE0 0x93022006+(16*N) Sprite N Attribute Register 0

Bit	15	14	13	12	11	10	9	8
Function	SPB_PB[1]	SPN_BLD	SPN_DEPTH			SPN_PALETTE		

Default - - - - - - - -

7	6	5	4	3	2	1	0
SPN_VS		SPN_HS			SPN_FLIP		SPN_COLOR

-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

P_PPU_SPRITEN_ATTRIBUTE1 0x93022008+(16*N) Sprite N Attribute Register 1

Bit	15	14	13	12	11	10	9	8
Function	SPN_MOSAIC		SPN_BLDLVL					

Default - - - - - - - -

7	6	5	4	3	2	1	0
SPN_PB[0]	SPN_CHARNUM[22:16]						

-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

TXN_PB[1:0]: TEXT N palette bank selection.

SPN_PB[1:0]: Sprite number N palette bank selection. The SPN_PB[0] is used only when “P1024” bit of P_PALETTE_CONTROL is set to 1, by default it is use as SPN_CHARNUM[23].

When P1024 of P_PALETTE_CONTROL is set to “0”, PPU will use palette RAM 2 for TEXT palette RAM if TXN_PB [1] is set to “1”; PPU will use palette RAM 2 or 3 for sprite palette RAM based on the PAL_TYPE [1] if SPN_PB[1] is set to “1”.

It is not allowed to access palette RAM arbitrarily when TEXT or sprite engine is active. The palette RAM can be changed only in vertical blanking time or when PPU is disabled; otherwise, the PAL_ERROR IRQ of P_PPU_IRQ_STATUS (0x9302018C) will be asserted.

Under 16-bit palette mode, the maximum color number is 32768 colors (15-bit). The following table shows the format of the 16 bits palette.

P_PPU_PALETTE_RAM0_N	0x93021000~0x930213FC	Palette RAM 0	
P_PPU_PALETTE_RAM1_N	0x93021400~0x930217FC	Palette RAM 1	
P_PPU_PALETTE_RAM2_N	0x93021800~0x93021BFC	Palette RAM 2	
P_PPU_PALETTE_RAM3_N	0x93021C00~0x93021FFC	Palette RAM 3	
Bit	15 14 13 12 11 10 9 8		
Function	T	R[4:0]	G[4:3]
Default	- - - - - - - -		
	7 6 5 4 3 2 1 0		
	G[2:0]	B[4:0]	
	- - - - - - - -		

Note: T is the transparent flag.

Under 25-bit palette mode, the maximum color number is 16777216 colors (24-bit). The following table shows the format of the 25 bits palette.

P_PPU_PALETTE_RAM0/1_N	0x93021000~0x930213FC	Palette RAM 0						
	0x93021400~0x930217FC	Palette RAM 1						
Bit	15 14 13 12 11 10 9 8							
Function	G[7:0]							
Default	0 0 0 0 0 0 0 0							
	7 6 5 4 3 2 1 0							
	B[7:0]							
	0 0 0 0 0 0 0 0							

P_PPU_PALETTE_RAM2/3_N	0x93021800~0x93021BFC	Palette RAM 2						
	0x93021C00~0x93021FFC	Palette RAM 3						
Bit	15 14 13 12 11 10 9 8							
Function	-							
Default	0 0 0 0 0 0 0 0	T						
	7 6 5 4 3 2 1 0							
	R[7:0]							
	0 0 0 0 0 0 0 0							

Note: T is the transparent flag.

13.2.3 Depth

Depth defines the relationship between TEXT and sprite. There are total of eight depth layers - four for TEXT and four for sprite. The following registers show the register location of depth for TEXT and sprite.

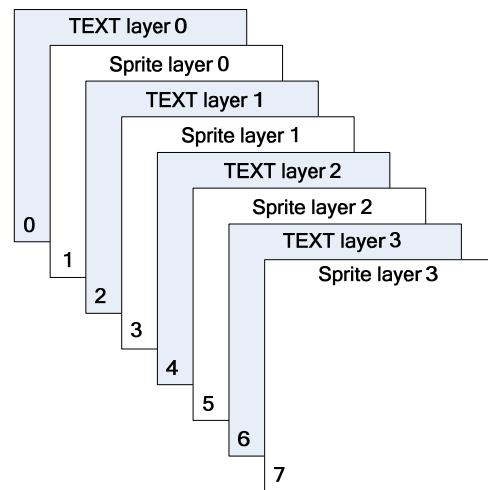
P_PPU_TEXTN_ATT 0x93020048 0x93020060 0x93020010 TEXT N Attribute Register

RIBUTE									0x93020030									(N=1,2,3,4)									
Bit	23	22	21	20	19	18	17	16	Function	TXN_PB[1:0]	TXN_EFF	-	TXN_WINDOW	TXN_SIZE[2]	Default	0	0	0	0	0	0	0	0	0	0		
	15	14	13	12	11	10	9	8		TXN_SIZE[1:0]	TXN_DEPTH	TXN_PALETTE						0	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0		TXN_VS	TXN_HS	TXN_FLIP			TXN_COLOR				0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0																			

P_PPU_SPRITEN_ATTRIBUTE0 0x93022006+(16*N) Sprite N Attribute Register 0

Bit	15	14	13	12	11	10	9	8	
Function	SPB_PB[1]	SPN_BLD	SPN_DEPTH	SPN_PALETTE					
Default	-	-	-	-	-	-	-	-	
	7	6	5	4	3	2	1	0	
	SPN_VS	SPN_HS	SPN_FLIP			SPN_COLOR			
	-	-	-	-	-	-	-	-	

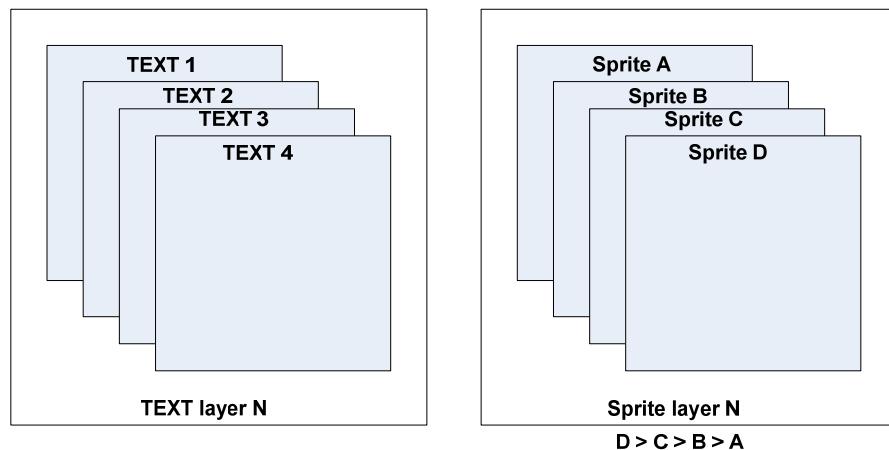
The following diagram shows the relationship between these 8 layers.



When multiple objects are on the same layer, the following relationship will apply.

TEXT	TEXT 4 > TEXT 3 > TEXT 2 > TEXT 1
Sprite	Sprite 1023 > Sprite 1022 > Sprite 1021 > Sprite 1020Sprite 1 > Sprite 0

The following diagram is the result on the screen when multiple objects are on the same layer.



13.2.4 Character Size

The character size of each sprite can be different, but the character size of a single TEXT must be the same. The character size is determined by the following registers.

P_PPU_TEXTN_ATT 0x93020048 0x93020060 0x93020010 TEXT N Attribute Register

RIBUTE	0x93020030								(N=1,2,3,4)	
Bit	23	22	21	20	19	18	17	16		
Function	TXN_PB[1:0]		TXN_EFF			-	TXN_WINDOW		TXN_SIZE[2]	
Default	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
TXN_SIZE[1:0]		TXN_DEPTH			TXN_PALETTE		

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
TXN_VS		TXN_HS		TXN_FLIP		TXN_COLOR	

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

P_PPU_SPRITEN_ATTRIBUTE0 0x93022006+(16*N) Sprite N Attribute Register 0

Bit	15	14	13	12	11	10	9	8
Function	SPB_PB[1]	SPN_BLD	SPN_DEPTH				SPN_PALETTE	

Default	-	-	-	-	-	-	-	-
---------	---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
SPN_VS		SPN_HS		SPN_FLIP		SPN_COLOR	

-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

The following table shows the relations between the register setting and real character size for sprite and TEXT with size among 512x256 ~ 1024x1024.

P_Sp_Control 0x93020108 Sprite Control Register

Bit	23	22	21	20	19	18	17	16
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Sprite_Number							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

SP_LS is 0 => Disable large sprite function, the SpN_LS will become useless under this mode.

SP_LS is 1 => Enable large sprite function, the SpN_LS will be used to control the sprite size according to the following table.

		TXN_HS, SPN_HS			
		0	1	2	3
TXN_VS, SPN_VS	0	8x8	16x8	32x8	64x8
	1	8x16	16x16	32x16	64x16
	2	8x32	16x32	32x32	64x32
	3	8x64	16x64	32x64	64x64

The following table shows the relations between the register setting and real character size for sprite with LS is 1 (or SP_LS is 1) and TEXT with size among 2048x1024 ~ 4096x4096.

		TXN_HS, SPN_HS			
		0	1	2	3
TXN_VS SPN_VS	0	32x32	64x32	128x32	256x32
	1	32x64	64x64	128x64	256x64
	2	32x128	64x128	128x128	256x128
	3	32x256	64x256	128x256	256x256

13.2.5 Character Flip

The flip function is designed to perform the mirror effect. The vertical and horizontal flip can be applied independently. The following table shows the register that controls the flip effect.

P_PPU_TEXTN_ATT 0x93020048 0x93020060 0x93020010 TEXT N Attribute Register

		0x93020030 (N=1,2,3,4)							
Bit	23	22	21	20	19	18	17	16	
Function	TXN_PB[1:0]		TXN_EFF		-	TXN_WINDOW		TXN_SIZE[2]	
Default	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	
	TXN_SIZE[1:0]		TXN_DEPTH		TXN_PALETTE				
	0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0	
	TXN_VS		TXN_HS		TXN_FLIP		TXN_COLOR		
	0	0	0	0	0	0	0	0	0

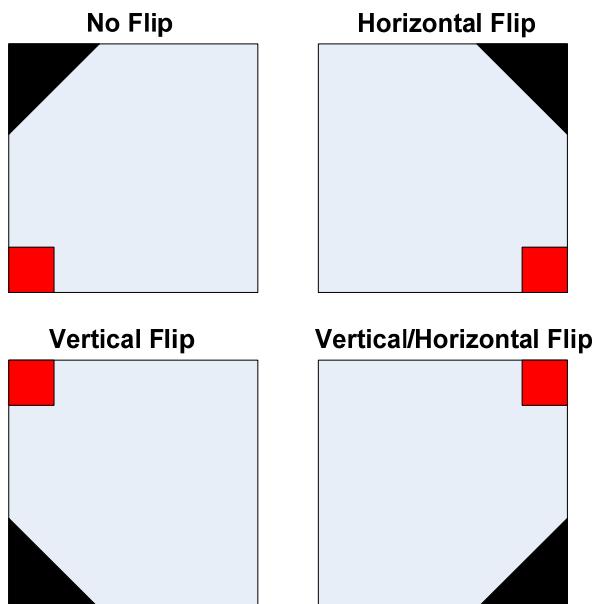
Note: When TXN_REGM of TEXT N is configured as "0", the TEXT engine will use the parameter stored in attribute RAM as the TXN_FLIP parameter.

P_PPU_SPRITEN_ATTRIBUTE0								0x93022006+(16*N)								Sprite N Attribute Register 0											
Bit	15	14	13	12	11	10	9	8	SPN_PALETTE																		
Function	SPB_PB[1]	SPN_BLD	SPN_DEPTH	SPN_PALETTE																							
Default	-	-	-	-	-	-	-	-																			
	7	6	5	4	3	2	1	0	SPN_VS								SPN_HS								SPN_COLOR		
	-	-	-	-	-	-	-	-																			

The following table shows the relation between register setting and real flip effect.

TXN_FLIP, SPN_FLIP	Action
2'b00	No flip
2'b01	Horizontal flip
2'b10	Vertical flip
2'b11	Horizontal flip and vertical flip

The following diagram shows the result of flip.



13.2.6 Blending Effect

The blending effect is to perform the “half-transparency” effect. The following registers are utilized to control the blending effect.

P_PPU_TEXTN_CTRL								0x9302004C	0x93020064	0x93020014	TEXT N Control Register	
								0x93020034			(N=1,2,3,4)	
Bit	15	14	13	12	11	10	9	8				
Function	TXN_BLDLVL						TXN_BLDMODE	TXN_BLD				
Default	0	0	0	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0				
	TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP					
	0	0	0	0	0	0	0	0	0	0	0	

Note: When TXN_REGM of TEXT N is configured as 0, the TEXT engine will use the parameter stored in attribute RAM as the TXN_BLD parameter.

P_PPU_SPRITEN_ATTRIBUTE0								0x93022006+(16*N)	Sprite N Attribute Register 0			
Bit	15	14	13	12	11	10	9	8				
Function	SPB_PB[1]	SPN_BLD	SPN_DEPTH		SPN_PALETTE							
Default	-	-	-	-	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0				
	SPN_VS		SPN_HS		SPN_FLIP		SPN_COLOR					
	-	-	-	-	-	-	-	-	-	-	-	-

P_PPU_SPRITEN_ATTRIBUTE1								0x93022008+(16*N)	Sprite N Attribute Register 1			
Bit	15	14	13	12	11	10	9	8				
Function	SPN_MOSAIC		SPN_BLDLVL									
Default	-	-	-	-	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0				
	SPN_PB[0]		SPN_CHARNUM[22:16]									
	-	-	-	-	-	-	-	-	-	-	-	-

P_PPU_SPRITE_CTRL								0x93020108	Sprite Control Register			
Bit	23	22	21	20	18	18	17	16				
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN				
Default	0	0	0	0	0	0	0	0				

15	14	13	12	11	10	9	8
SPRITE_NUMBER							

0	0	0	0	0	0	0	0
SP_ZOOMEN SP_ROTEN SP_MOSEN SP_DIRECT - SP_BLDMODE COORD_SEL SP_EN							

P_PPU_BLENDING **0x930200A8** **Blending Control Register**

Bit	7	6	5	4	3	2	1	0
Function	-							BLDLVL
Default	0	0	0	0	0	0	0	0

The following table shows the relationship between register setting and real blending effect.

TEXT			Sprite		
TXN_BLD	TXN_BLDMODE	Result	SPN_BLD	SP_BLDMODE	Result
0	--	No blending	0	--	No blending
1	0	BLDLVL/4	1	0	BLDLVL/4
	1	TXN_BLDLVL/64		1	SPN_BLDLVL/64

The following equation is used to calculate the final result of blending.

$$\text{Final RGB} = ((\text{Upper TEXT/sprite layer RGB}) * (\text{BLD_LVL})) + (\text{Lower TEXT/sprite layer RGB}) * (64-\text{BLD_LVL})/64$$

The following example is 64-level blending effect between sprite and TEXT.



In order to guarantee the correct blending result, the following rule must be followed.

1. Set TX_BOTUP (bit 3 of 0x930201FC) bit to 1, and this will enable the blending correct function.
2. Re-order your sprites from lower depth to higher depth, which means sprite with lowr sprite number must have lower depth setting.

13.2.7 Rotate Effect

The GP12 has a powerful rotating engine which can perform the sprite rotation in a unit of 5.625 degree. Only the character in sprite can be rotated. The rotate function will be turned off when sprite virtual 3D mode is turned on, but programmer can still use sprite virtual 3D function to facilitate the rotating function by adjusting the coordination of each vertex of a sprite. The following registers are the rotating control register.

P_PPU_SPRITEN_X_POSITION								0x93022002+(16*N)								Sprite N X/Rotate Register								
Bit	15	14	13	12	11	10	9	8	ANGLE								X [9:8]							
Function	-	-	-	-	-	-	-	-	X [7:0]								X [7:0]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P_PPU_SPRITE_CTRL								0x93020108								Sprite Control Register								
Bit	23	22	21	20	18	18	17	16	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN	SPRITE_NUMBER							
Function	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
								SPRITE_NUMBER																
								SPRITE_NUMBER																
								SPRITE_NUMBER																
								SPRITE_NUMBER																
Bit	15	14	13	12	11	10	9	8	SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN	0	0	0	0	0	0	0	0
Function	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

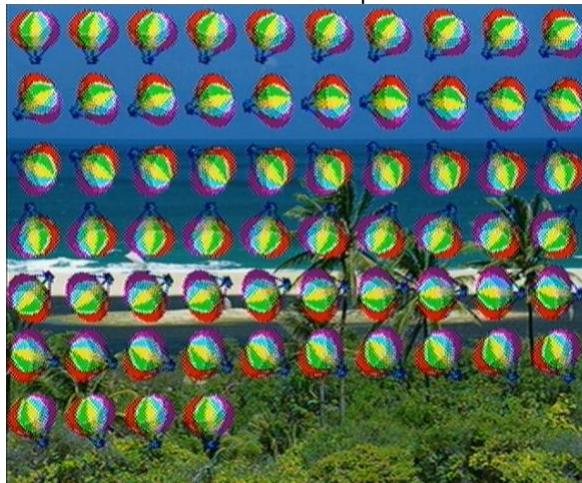
Note: SP_ROTEN must be set to 1 before enabling the sprite rotate function.

The following table shows the real rotate angle in “clockwise” of each ANGLE setting.

ANGLE	Real	ANGLE	Real	ANGLE	Real	ANGLE	Real
0	0.000°	16	90.000°	32	180.000°	48	270.000°
1	5.625°	17	95.625°	33	185.625°	49	275.625°
2	11.250°	18	101.250°	34	191.250°	50	281.250°
3	16.875°	19	106.875°	35	196.875°	51	286.875°
4	22.500°	20	112.500°	36	202.500°	52	292.500°
5	28.125°	21	118.125°	37	208.125°	53	298.125°
6	33.750°	22	123.750°	38	213.750°	54	303.750°
7	39.375°	23	129.375°	39	219.375°	55	309.375°
8	45.000°	24	135.000°	40	225.000°	56	315.000°

ANGLE	Real	ANGLE	Real	ANGLE	Real	ANGLE	Real
9	50.625°	25	140.625°	41	230.625°	57	320.625°
10	56.250°	26	146.250°	42	236.250°	58	326.250°
11	61.875°	27	151.875°	43	241.875°	59	331.875°
12	67.500°	28	157.500°	44	247.500°	60	337.500°
13	73.125°	29	163.125°	45	253.125°	61	343.125°
14	78.750°	30	168.750°	46	258.750°	62	348.750°
15	84.375°	31	174.375°	47	264.375°	63	354.375°

The following example is a 64-level rotate effect of a sprite from 0° to 354.375°.



13.2.8 Zoom Effect

The GP12 has a powerful zoom in/out engine which can perform the sprite zoom in and zoom out in the range from 1/32 to 8.75. Only the character in sprite can do zoom in/out. The zoom function will be turned off when sprite virtual 3D mode is turned on, but programmer can still use sprite virtual 3D function to facilitate the zoom function by adjusting the coordination of each vertex of a sprite. The following registers are the zoom control registers.

P_PPU_SPRITEN_Y_POSITION 0x93022004+(16*N) Sprite N Y/Zoom Register							
Bit	15	14	13	12	11	10	9
Function	ZOOM						Y[9:8]
Default	-	-	-	-	-	-	-
	7	6	5	4	3	2	1
	Y[7:0]						
	-	-	-	-	-	-	-

P_PPU_SPRITE_CTRL								Sprite Control Register								
Bit	23	22	21	20	18	18	17	16								
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN								
Default	0	0	0	0	0	0	0	0								
	15	14	13	12	11	10	9	8								
	SPRITE_NUMBER															
	0	0	0	0	0	0	0	0								
	7	6	5	4	3	2	1	0								
	SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN								
	0	0	0	0	0	0	0	0								

Note: SP_ZOOMEN must be set to 1 before enabling the zoom function.

The following table shows the real zoom size of each ZOOM setting.

ZOOM	Zoom Out	ZOOM	Zoom Out	ZOOM	Zoom In	ZOOM	Zoom In
0	No Zoom	16	16/32	32	No Zoom	48	20/4
1	1/32	17	17/32	33	5/4	49	21/4
2	2/32	18	18/32	34	6/4	50	22/4
3	3/32	19	19/32	35	7/4	51	23/4
4	4/32	20	20/32	36	8/4	52	24/4
5	5/32	21	21/32	37	9/4	53	25/4
6	6/32	22	22/32	38	10/4	54	26/4
7	7/32	23	23/32	39	11/4	55	27/4
8	8/32	24	24/32	40	12/4	56	28/4
9	9/32	25	25/32	41	13/4	57	29/4
10	10/32	26	26/32	42	14/4	58	30/4
11	11/32	27	27/32	43	15/4	59	31/4
12	12/32	28	28/32	44	16/4	60	32/4
13	13/32	29	29/32	45	17/4	61	33/4
14	14/32	30	30/32	46	18/4	62	34/4
15	15/32	31	31/32	47	19/4	63	35/4

The following pictures are some zoom effect examples.



Zoom-out x 1/4

**Original
(64x64 sprite)**

Zoom-in x 2.0

Zoom-in x 8.75

13.2.9 Mosaic Effect

The mosaic function is to perform mosaic effect on a sprite. GP12 supports 4-level mosaic effect for sprites and each one can have independent mosaic level. The characters inside a TEXT do not support the mosaic function. The following registers are the mosaic control registers.

P_PPU_SPRITEN_ATTRIBUTE1 0x93022008+(16*N) Sprite N Attribute Register 1								
Bit	15	14	13	12	11	10	9	8
Function	SPN_MOSAIC SPN_BLDLVL							
Default	-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
SPN_PB[0]	SPN_CHARNUM[22:16]						

P_PPU_SPRITE_CTRL 0x93020108 Sprite Control Register								
Bit	23	22	21	20	18	18	17	16
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SPRITE_NUMBER							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN

Note: SP_MOSEN must set to "1" before enabling the mosaic function.

The following table shows the relationship between mosaic register setting and real mosaic effect.

SPN_LS	SPN_MOSAIC	Mosaic Effect
0	2'b00	No effect
0	2'b01	2 pixels mosaic effect
0	2'b10	4 pixels mosaic effect
0	2'b11	8 pixels mosaic effect
1	2'b00	No effect
1	2'b01	8 pixels mosaic effect
1	2'b10	16 pixels mosaic effect
1	2'b11	32 pixels mosaic effect

The following example is 4-level mosaic effect for a sprite.



13.2.10 Fade Effect

Fade effects contain fade-in and fade-out to the entire screen. It is controlled by the register “P_FADE_CONTROL”. The FADE_OFFSET value defines the fade-out level. In other words, the intensity of all pixels on the visible region will be subtracted with the FADE_OFFSET value. The fade-out effect can be done by changing the FADE_OFFSET value from 0x00 to 0xFF, and in contrast, fade-in effect is achieved by changing the FADE_OFFSET value from 0xFF to 0x00.

P_PPU_FADE_CTRL		0x930200C0								Fade Control Register							
Bit	7	6	5	4	3	2	1	0	Function	FADE_OFFSET							
Default	0	0	0	0	0	0	0	0		FADE_OFFSET							

The fade effects are shown as follows:



13.2.11 Character Number

GP12 has two addressing mode (relative addressing mode and direct addressing mode) to find the character address. The character number is a way to find the start address of a character.

Relative Addressing Mode (Default):

Real Address = Base Address + (Character Number [15:0] X Character Size X 2)

It should be noted that the unit of character number is character size, and the unit of character size is half-word (16 bits); therefore, the real address should be multiplying by two, in order to transform to byte address.

Direct Addressing Mode:

Real Address = Base Address + {Character Number [31:1], 1'b0} for character of TEXT.

Real Address = Base Address + {Character Number [23:0], 1'b0} for character of sprite.

The base address is defined in segment register of each TEXT or sprite.

TX1 Base Address	{SEGMENT_TX1[31:1], 1'b0}
TX2 Base Address	{SEGMENT_TX2[31:1], 1'b0}
TX3 Base Address	{SEGMENT_TX3[31:1], 1'b0}
TX4 Base Address	{SEGMENT_TX4[31:1], 1'b0}
Sprite Base Address	{SEGMENT_SP[31:1], 1'b0}

Note:

1. When TEXT N BMP mode is set, the base address will be zero regardless what is setting in SEGMENT_TXN.
2. The last bit of base address will be neglected, which means the base address should be half-word aligned.

The following registers are used to select the TEXT sprite addressing mode.

P_PPU_ENABLE								0x930201FC								PPU Control Register								
Bit	31	30	29	28	27	26	25	24																
Function	-							DEFEN								TFTLB								
Default	0	0	0	0	0	0	0	0																
	23	22	21	20	19	18	17	16																
	-	YUV_TYPE							LB								TFT_SIZE							
	0	0	0	0	0	0	0	0																

15	14	13	12	11	10	9	8
SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO	SPR25D	FB_FORMAT	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
FB_EN	-	VGA_NOINTL	VGA_EN	TX_BOTUP	TX_DIRECT	CH0_BLK	PPU_EN
0	0	0	0	0	0	0	1

TX_DIRECT is 0 => TEXT engine uses relative addressing mode.

TX_DIRECT is 1 => TEXT engine uses direct addressing mode.

P_PPU_SPRITE_CTRL 0x93020108 Sprite Control Register

Bit	23	22	21	20	18	18	17	16
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SPRITE_NUMBER							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN
0	0	0	0	0	0	0	0

SP_DIRECT is 0 => Sprite engine uses relative addressing mode.

SP_DIRECT is 1 => Sprite engine uses direct addressing mode.

P_PPU_TEXTN_SEG 0x93020080 0x93020084 0x9302008C TEXT N Segment Register

MENT	31	30	29	28	27	26	25	24
SEGMENT_TXN [31:24]								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
SEGMENT_TXN [23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SEGMENT_TXN [15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SEGMENT_TXN [7:0]							
0	0	0	0	0	0	0	0

P_PPU_SPRITE_SEGMENT 0x93020088								Sprite Segment Register									
Bit	31	30	29	28	27	26	25	24									
Function	SEGMENT_SP [31:24]																
Default	0	0	0	0	0	0	0	0									
								SEGMENT_SP [23:16]									
								0	0	0	0	0	0	0	0		
								SEGMENT_SP [15:8]									
								0	0	0	0	0	0	0	0		
								SEGMENT_SP [7:0]									
								0	0	0	0	0	0	0	0		

In the relative addressing mode, the address is calculated from the character size. The following table shows the character size in the unit of half-word (16 bits) under various conditions.

		Horizontal Size (Pixels)							
		8		16		32		64	
Vertical Size (Pixels)	8	Color	8x8	Color	16x8	Color	32x8	Color	64x8
	8	4	8	4	16	4	32	4	64
	8	16	16	16	32	16	64	16	128
	8	64	24	64	48	64	96	64	192
	8	256	32	256	64	256	128	256	256
	16	Color	8x16	Color	16x16	Color	32x16	Color	64x16
	16	4	16	4	32	4	64	4	128
	16	16	32	16	64	16	128	16	256
	16	64	48	64	96	64	192	64	384
	16	256	64	256	128	256	256	256	512
	32	Color	8x32	Color	16x32	Color	32x32	Color	64x32
	32	4	32	4	64	4	128	4	256
	32	16	64	16	128	16	256	16	512
	32	64	96	64	192	64	384	64	768
	32	256	128	256	256	256	512	256	1024
	64	Color	8x64	Color	16x64	Color	32x64	Color	64x64
	64	4	64	4	128	4	256	4	512
	64	16	128	16	256	16	512	16	1024
	64	64	192	64	384	64	768	64	1536
	64	256	256	256	512	256	1024	256	2048

Note: The character size in bitmap mode or RGB mode is “1”, which means the character number of each line is equal to the start address of the line.

13.2.11.1 TEXT Character Number

The character number of TEXT is stored in the number and/or attribute array in anywhere of system memory. The following registers are used to control the number and attribute array.

P_PPU_TEXTN_CTRL 0x9302004C 0x93020064 0x93020014 TEXT N Control Register
0x93020034
(N=1,2,3,4)

Bit	15	14	13	12	11	10	9	8
Function	TXN_BLDLVL						TXN_BLDMODE	TXN_BLD
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP	

0 0 0 0 0 0 0 0

TXN_BMP is 0 => Character mode

TXN_BMP is 1 => Bitmap mode

TXN_REGM is 0 => TEXT attribute on RAM

TXN_REGM is 1 => TEXT attribute in register

TXN_RGBM is 0 => Disable high-color mode (4/16/64/256 colors)

TXN_RGBM is 1 => Enable high-color mode (32768/65536/RGBG/YUYV colors)

P_PPU_TEXTN_N_PTR 0x93020050 0x93020068 0x93020018 TEXT N Number Array Register
0x93020038
(N=1,2,3,4)

Bit	31	30	29	28	27	26	25	24
Function	TXN_N_PTR [31:24]							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
TXN_N_PTR [23:16]							

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
TXN_N_PTR [15:8]							

0 0 0 0 0 0 0 0

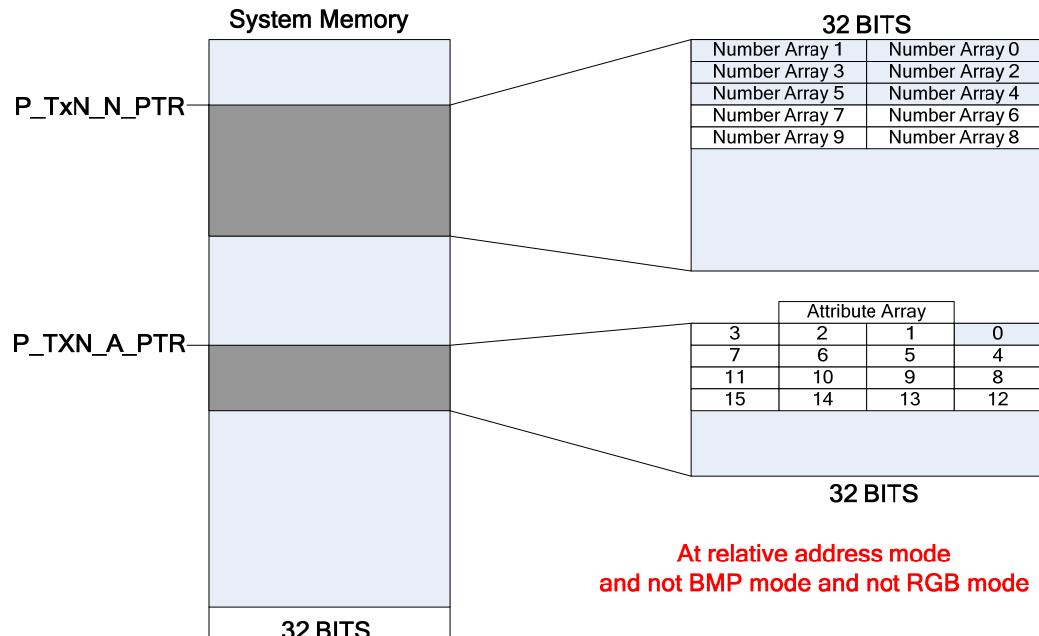
7	6	5	4	3	2	1	0
TXN_N_PTR [7:0]							

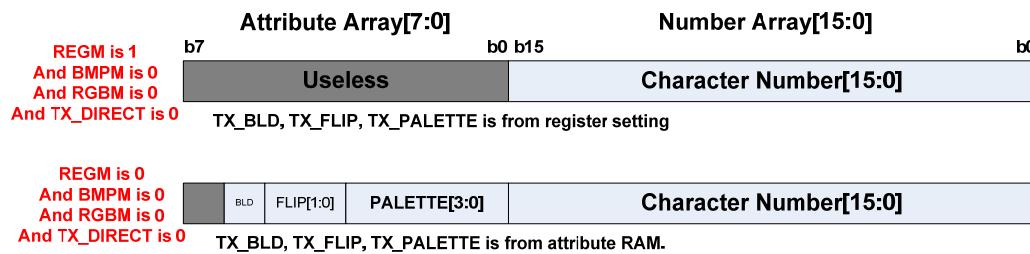
0 0 0 0 0 0 0 0

P_PPU_TEXTN_A_PTR 0x93020054 0x9302006C 0x9302001C TEXT N Attribute Array Register
0x9302003C
(N=1,2,3,4)

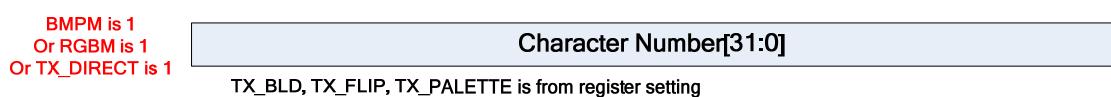
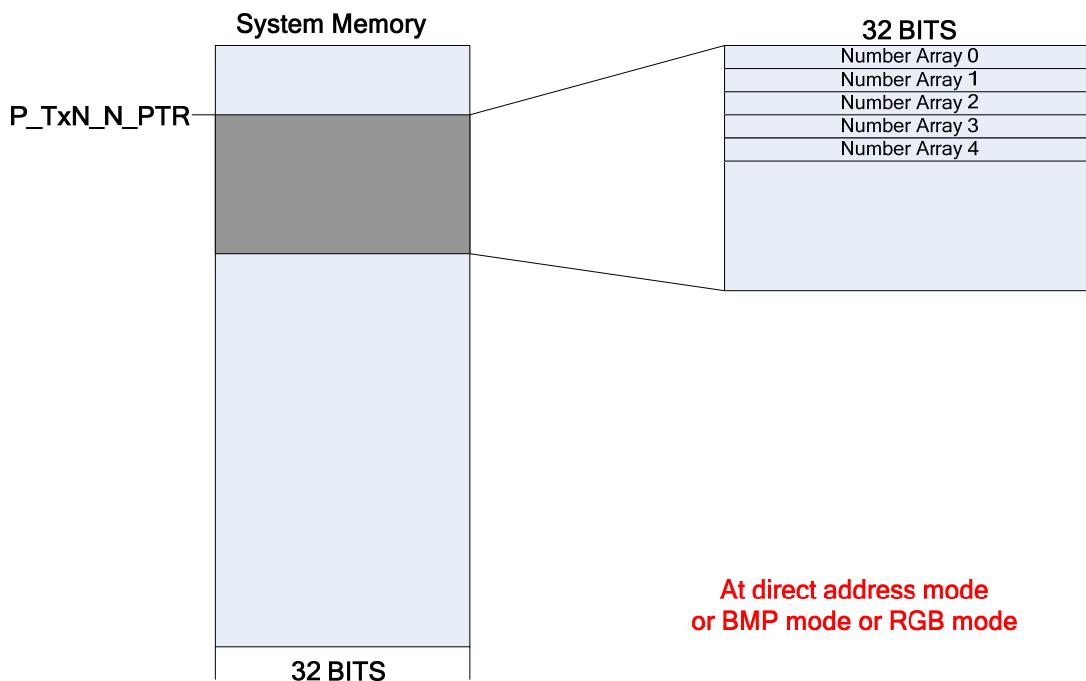
Bit	31	30	29	28	27	26	25	24
Function	TXN_A_PTR [31:24]							
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	TXN_A_PTR [23:16]							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	TXN_A_PTR [15:8]							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	TXN_A_PTR [7:0]							
	0	0	0	0	0	0	0	0

The following diagram shows the TEXT arrays' format when relative address mode is applied, and when RGBM and BMPM are not enabled.





When direct addressing mode is applied or BMPM or RGBM are enabled, the following diagram shows the TEXT number arrays' format.



The following table shows more details about the TEXT addressing mode under various settings.

TX_DIRECT	REGM	BMPM	RGBM	TX_HS	Start Address	TX_BLD, TX_FLIP, TX_PALETTE
1	X	X	X	X	{CharNum[31:1], 1'b0}	From register
X	1	1	X	3	{SEGMENT_TXN[31:1], 1'b0}	From register
0	X	1	X	X	{CharNum[31:1], 1'b0}	From register
0	X	X	1	X	{CharNum[31:1], 1'b0}	From register
0	0	0	0	X	CharNum[15:0]*CharSize*2	From attribute RAM
0	1	0	0	X	CharNum[15:0]*CharSize*2	From register

Note 1: X means don't care.

Note 2: {SEGMENT_TXN[31:1], 1'b0} will be added to the start address on real system.

Note 3: A special mode for BMP mode is added for pure BMP application. Under this mode, it is not necessary to use system RAM. Programmer only needs to specify the start address and line length of the BMP data. Under this mode, the hardware will have better performance because it does not need to read number and attribute RAM.

The following table shows the line length in special BMP mode.

Available Text Size	TX_VS[1:0]	Pixels per Line	Color	Half Words per Line
512x256, 512x512, 1024x512, 1024x1024	0	SBMP_MODE is 0 => 320. SBMP_MODE is 1 => Same as width of screen size.	RGB	320
			4 colors	40
			16 colors	80
			64 colors	120
			256 colors	160
1024x512, 1024x1024	1	640	RGB	640
			4 colors	80
			16 colors	160
			64 colors	240
			256 colors	320
512x256, 512x512, 1024x512, 1024x1024	2	512	RGB	512
			4 colors	64
			16 colors	128
			64 colors	192
			256 colors	256
1024x512, 1024x1024	3	1024	RGB	1024
			4 colors	128
			16 colors	256
			64 colors	384
			256 colors	512
2048x1024, 2048x2048	Don't care	2048	RGB	2048
			4 colors	256
			16 colors	512
			64 colors	768
			256 colors	1024
4096x2048, 4096x4096	Don't care	4096	RGB	4096
			4 colors	512
			16 colors	1024
			64 colors	1536
			256 colors	2048

Note: When TX_HS = 3 and TX_REGM = 1 and TX_BMPM = 1, the special BMP mode is active.

For example, when TX_VS is setting as 1, and color mode is 16 colors and the start address of line X = {SEGMENT_TXN [31:1], 1'b0} + (X * 160 * 2).

Sprite Character Number

The same as the TEXT engine, sprite engine has both the relative mode and direct mode. The control register is SP_DIRECT (P_PPU_SPRITE_CTRL[4]). The following registers are used for character number.

P_PPU_SPRITEN_CHARNUM 0x93022000+(16*N) Sprite N CharNum Register								
Bit	15	14	13	12	11	10	9 8	
Function	SPN_CHARNUM[15:8]							
Default	-	-	-	-	-	-	-	
Bit	7	6	5	4	3	2	1 0	
	SPN_CHARNUM[7:0]							
	-	-	-	-	-	-	-	

P_PPU_SPRITEN_ATTRIBUTE1 0x93022008+(16*N) Sprite N Attribute Register 1								
Bit	15	14	13	12	11	10	9 8	
Function	SPN_MOSAIC		SPN_BLDLVL					
Default	-	-	-	-	-	-	-	
Bit	7	6	5	4	3	2	1 0	
	SPN_PB[0] SPN_CHARNUM[22:16]							
	-	-	-	-	-	-	-	

Note: SPN_PB[0] is used only when P1024 (P_PALETTE_CONTROL[4]) is set to "1"; by default, it is used as SPN_CHARNUM[23].

The following table shows more details about the sprite addressing mode under various settings.

SP_DIRECT	Start Address
1	SP_FAR = 1 => SPN_CHARNUM [23:0] * 16
	SP_FAR = 0 => SPN_CHARNUM [23:0] * 2
0	CHARNUM [15:0] * CHARSIZE * 2

Note:

1. X means don't care.
2. {SEGMENT_SP [31:1], 1'b0} will be added to the start address on real system.
3. SPN_PB [0] is used as CHARNUM [23] when P1024 (P_PALETTE_CONTROL[4]) is set to "0".

13.3 TEXT Control

13.3.1 TEXT Register Summary

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_TEXT3_X_POSITION	0x93020000	-															TX3_X_POSITION [11:0]
R_PPU_TEXT3_Y_POSITION	0x93020004	-															TX3_Y_POSITION [11:0]
P_PPU_TEXT3_X_OFFSET	0x93020008	-															TX3_X_OFFSET [9:0]
P_PPU_TEXT3_Y_OFFSET	0x9302000C	-															TX3_Y_OFFSET [9:0]
P_PPU_TEXT3_ATTRIBUTE	0x93020010	SIZE[1:0]	DEPTH														COLOR
P_PPU_TEXT3_ATTRIBUTE	0x93020012	-				CMASK		PB [1:0]			EFF	-		WINDOW		SIZE [2]	
P_PPU_TEXT3_CTRL	0x93020014		BLDLVL		BLDM	BLD	RGBM		MODE	MVE	EN	WALL	REGM			BMPM	
P_PPU_TEXT3_N_PTR	0x93020018															TX3_N_PTR [15:0]	
P_PPU_TEXT3_N_PTR	0x9302001A															TX3_N_PTR [31:16]	
P_PPU_TEXT3_A_PTR	0x9302001C															TX3_A_PTR [15:0]	
P_PPU_TEXT3_A_PTR	0x9302001E															TX3_A_PTR [31:16]	
P_PPU_TEXT4_X_POSITION	0x93020020	-														TX4_X_POSITION [11:0]	
R_PPU_TEXT4_Y_POSITION	0x93020024	-														TX4_Y_POSITION [11:0]	
P_PPU_TEXT4_X_OFFSET	0x93020028	-														TX4_X_OFFSET [9:0]	
P_PPU_TEXT4_Y_OFFSET	0x9302002C	-														TX4_Y_OFFSET [9:0]	
P_PPU_TEXT4_ATTRIBUTE	0x93020030	SIZE[1:0]	DEPTH													COLOR	
P_PPU_TEXT4_ATTRIBUTE	0x93020032	-				CMASK		PB[1:0]			EFF	-		WINDOW		SIZE [2]	

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_TEXT4_CTRL	0x930 20034			BLDLVL		BLDM	BLD	RGBM		MODE	MVE	EN	WALL	REGM	BMPM		
P_PPU_TEXT4_N_PTR	0x930 20038									TX4_N_PTR [15:0]							
P_PPU_TEXT4_N_PTR	0x930 2003A									TX4_N_PTR [31:16]							
P_PPU_TEXT4_A_PTR	0x930 2003C									TX4_A_PTR [15:0]							
P_PPU_TEXT4_A_PTR	0x930 2003D									TX4_A_PTR [31:16]							
P_PPU_TEXT1_X_POSITION	0x930 20040	-								TX1_X_POSITION [11:0]							
R_PPU_TEXT1_Y_POSITION	0x930 20044	-								TX1_Y_POSITION [11:0]							
P_PPU_TEXT1_X_OFFSET	0x930 20340		-							TX1_X_OFFSET [9:0]							
P_PPU_TEXT1_Y_OFFSET	0x930 20344		-							TX1_Y_OFFSET [9:0]							
P_PPU_TEXT1_ATTRIBUTE	0x930 20048	SIZE	DEPTH		PALETTE			VS		HS		FLIP		COLOR			
P_PPU_TEXT1_ATTRIBUTE	0x930 2004A		-		CMASK		PB[1:0]		EFF			WINDOW		SIZE [2]			
P_PPU_TEXT1_CTRL	0x930 2004C		BLDLVL		BLDM	BLD	RGBM	MODE	MVE	EN	WALL	REGM	BMPM				
P_PPU_TEXT1_N_PTR	0x930 20050									TX1_N_PTR [15:0]							
P_PPU_TEXT1_N_PTR	0x930 20052									TX1_N_PTR [31:16]							
P_PPU_TEXT1_A_PTR	0x930 20054									TX1_A_PTR [15:0]							
P_PPU_TEXT1_A_PTR	0x930 20056									TX1_A_PTR [31:16]							
P_PPU_TEXT2_X_POSITION	0x930 20058	-								TX2_X_POSITION [11:0]							
R_PPU_TEXT2_Y_POSITION	0x930 2005C	-								TX2_Y_POSITION [11:0]							
P_PPU_TEXT2_X_OFFSET	0x930 20350	-								TX2_X_OFFSET [9:0]							



NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	
P_PPU_TEXT2_Y_OFFSET	0x93020354	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TX2_Y_OFFSET [9:0]	
P_PPU_TEXT2_ATTRIBUTE	0x93020060	SIZE[1:0]	DEPTH	PALETTE				VS		HS		FLIP		COLOR				
P_PPU_TEXT2_ATTRIBUTE	0x93020062	-	-	CMASK				PB[1:0]		EFF		-	WINDOW			SIZE [2]		
P_PPU_TEXT2_CTRL	0x93020064	BLDLVL				BLDM	BLD	RGBM	MODE		MVE	EN	WALL	REGM	BMPM			
P_PPU_TEXT2_N_PTR	0x93020068	TX2_N_PTR [15:0]																
P_PPU_TEXT2_N_PTR	0x9302006A	TX2_N_PTR [31:16]																
P_PPU_TEXT2_A_PTR	0x9302006C	TX2_A_PTR [15:0]																
P_PPU_TEXT2_A_PTR	0x9302006E	TX2_A_PTR [31:16]																
P_PPU_VCOMP_VALUE	0x93020070	-				VCOMP_VALUE[8:0]												
P_PPU_VCOMP_OFFSET	0x93020074	-				VCOMP_OFFSET[8:0]												
P_PPU_VCOMP_STEP	0x93020078	-				VCOMP_STEP[8:0]												
P_PPU_TEXT1_SEGMENT	0x93020080	SEGMENT_TX1 [15:0]																
P_PPU_TEXT1_SEGMENT	0x93020082	SEGMENT_TX1 [31:16]																
P_PPU_TEXT2_SEGMENT	0x93020084	SEGMENT_TX2 [15:0]																
P_PPU_TEXT2_SEGMENT	0x93020086	SEGMENT_TX2 [31:16]																
P_PPU_TEXT3_SEGMENT	0x9302008C	SEGMENT_TX3 [15:0]																
P_PPU_TEXT3_SEGMENT	0x9302008E	SEGMENT_TX3 [31:16]																
P_PPU_TEXT4_SEGMENT	0x93020090	SEGMENT_TX4 [15:0]																
P_PPU_TEXT4_SEGMENT	0x93020092	SEGMENT_TX4 [31:16]																

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_TEXT1_COSINE	0x93020348	-															TX1_COSINE [12:0]
P_PPU_TEXT1_SINE	0x9302034C	-															TX1_SINE [12:0]
P_PPU_TEXT2_COSINE	0x93020358	-															TX2_COSINE [12:0]
P_PPU_TEXT2_SINE	0x9302035C	-															TX2_SINE [12:0]
P_PPU_TEXT4_COSINE	0x930200A0	-															TX4_COSINE [12:0]
P_PPU_TEXT4_SINE	0x930200A4	-															TX4_SINE [12:0]
P_PPU_Y25D_COMPRESS	0x93020104																Y_COMPRESS [5:0]
P_PPU_BLD_COLOR	0x930201F4																BLD_COLOR [15:0]
P_PPU_MISC	0x930201F8																SBMP_MOD_E
P_PPU_MISC	0x930201F6	TV_LB -	TFT_VTQ	-	INTP_M_ODE	NEW_CM_P	TXT_ALPH_A	SP_A_DDR_X2	DUAL_BLD	-	SPR_RGB_A	TXT_RG_BA	FB_LOCK	-			-

13.3.2 TEXT SRAM Summary

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_TEXT_H_OFFSET0	0x93020400	-															TX_HOFFSET0[9:0]
P_PPU_TEXT_H_OFFSET1	0x93020404	-															TX_HOFFSET1[9:0]
...
P_PPU_TEXT_H_OFFSET238	0x930207B8	-															TX_HOFFSET238[9:0]
P_PPU_TEXT_H_OFFSET239	0x930207BC	-															TX_HOFFSET239[9:0]

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_TEXT_HCMP_VALUE0	0x93020800	-										HCMP_VALUE0[7:0]					
P_PPU_TEXT_HCMP_VALUE1	0x93020804	-										HCMP_VALUE1[7:0]					
...					
P_PPU_TEXT_HCMP_VALUE238	0x93020BB8	-										HCMP_VALUE238[7:0]					
P_PPU_TEXT_HCMP_VALUE239	0x93020BBC	-										HCMP_VALUE239[7:0]					

Note: This RAM is not valid when TEXT3 virtual 3D mode is enabled.

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_TEXT3_COS0	0x93020800				TX3_COS0[12:0]												
P_PPU_TEXT3_SIN0	0x93020804				TX3_SIN0[12:0]												
P_PPU_TEXT3_COS1	0x93020808				TX3_COS1[12:0]												
P_PPU_TEXT3_SIN1	0x9302080C				TX3_SIN1[12:0]												
...												
P_PPU_TEXT3_COS238	0x93020F70				TX3_COS238[12:0]												
P_PPU_TEXT3_SIN238	0x93020F74				TX3_SIN238[12:0]												
P_PPU_TEXT3_COS239	0x93020F78				TX3_COS239[12:0]												
P_PPU_TEXT3_SIN239	0x93020F7C				TX3_SIN239[12:0]												

13.3.3 TEXT Layer Structure

A TEXT can be constructed by characters (character mode) or lines (Bitmap mode). And the size of each TEXT can be configured individually by register. Following registers are used to control TEXT size and mode.

P_PPU_TEXTN_CTRL 0x9302004C 0x93020064 0x93020014 TEXT N Control Register (N=1,2,3,4)							
Bit	15	14	13	12	11	10	9 8
Function	TXN_BLDLVL						TXN_BLDMODE TXN_BLD
Default	0	0	0	0	0	0	0 0
	7	6	5	4	3	2	1 0
	TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP
	0	0	0	0	0	0	0 0

TXN_BMP is 0 => Character mode

TXN_BMP is 1 => Bitmap mode

TXN_WALL is 0 => Normal mode

TXN_WALL is 1 => Enable wallpaper mode, all the characters or lines are the same.

P_PPU_TEXTN_ATT 0x93020048 0x93020060 0x93020010 TEXT N Attribute Register (N=1,2,3,4)								
Bit	23	22	21	20	19	18	17 16	
Function	TXN_PB[1:0]		TXN_EFF			-	TXN_WINDOW TXN_SIZE[2]	
Default	0	0	0	0	0	0	0 0	
	15	14	13	12	11	10	9 8	
	TXN_SIZE[1:0]	TXN_DEPTH			TXN_PALETTE			
	0	0	0	0	0	0	0 0	
	TXN_VS	TXN_HS		TXN_FLIP		TXN_COLOR		
	0	0	0	0	0	0	0 0	

The following table shows the relationship between TXN_SIZE and real TEXT size.

TXN_SIZE	TX_HWIDTH	TX_VWIDTH	Character Size
3'b000	512	256	8x8~64x64
3'b001	512	512	8x8~64x64
3'b010	1024	512	8x8~64x64
3'b011	1024	1024	8x8~64x64
3'b100	2048	1024	32x32~256x256
3'b101	2048	2048	32x32~256x256

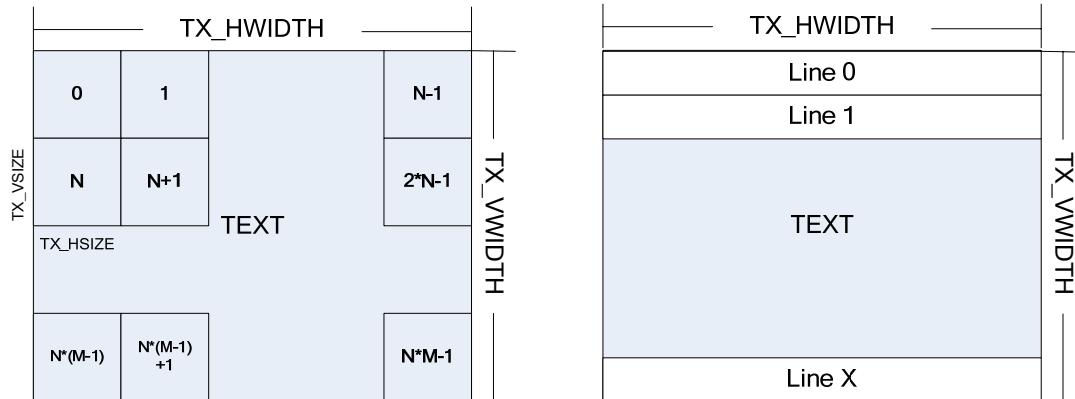
TXN_SIZE	TX_HWIDTH	TX_VWIDTH	Character Size
3'b110	4096	2048	32x32~256x256
3'b111	4096	4096	32x32~256x256

The following diagram shows the composite of a TEXT layer under these two modes.

WALLM = 0

Character Mode

Bitmap Mode



$$N = TX_HWIDTH / TX_HSIZE$$

$$M = TX_VWIDTH / TX_VSIZE$$

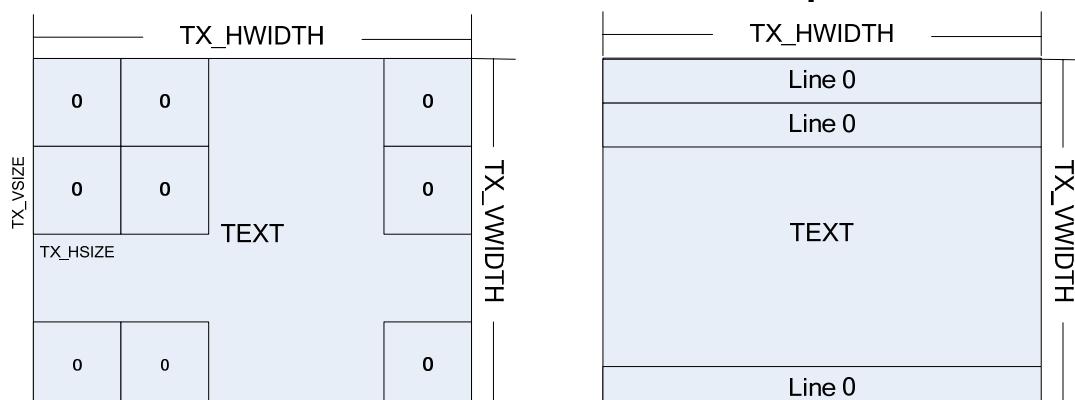
$$X = TX_VWIDTH - 1$$

The following diagram shows the composite of a TEXT layer under wallpaper mode.

WALLM = 1

Character Mode

Bitmap Mode



Since each character or line needs a character number register and/or attribute register, the table concludes the possible system memory needs for a single TEXT layer under various conditions.

TEXT Size	BMPM/RGBM	HS	VS	Number Array (HWords)	Attribute Array (HWords)	TEXT Size	BMPM/RGBM	HS	VS	Number Array (HWords)	Attribute Array (HWords)
512x 256	0/0	8	8	2048	1024	512x 512	0/0	8	8	4096	2048
			16	1024	512				16	2048	1024
			32	512	256				32	1024	512
			64	256	128				64	512	256
		16	8	1024	512			16	8	2048	1024
			16	512	256				16	1024	512
			32	256	128				32	512	256
			64	128	64				64	256	128
		32	8	512	256			32	8	1024	512
			16	256	128				16	512	256
			32	128	64				32	256	128
			64	64	32				64	128	64
		64	8	256	128			64	8	512	256
			16	128	64				16	256	128
			32	64	32				32	128	64
			64	32	16				64	64	32
	1/1	--	--	256*2	--		1/1	--	--	512*2	--
1024 x 512	0/0	8	8	8192	4096	1024x 1024	0/0	8	8	16384	8192
			16	4096	2048				16	8192	4096
			32	2048	1024				32	4096	2048
			64	1024	512				64	2048	1024
		16	8	4096	2048			16	8	8192	4096
			16	2048	1024				16	4096	2048
			32	1024	512				32	2048	1024
			64	512	256				64	1024	512
		32	8	2048	1024			32	8	4096	2048
			16	1024	512				16	2048	1024
			32	512	256				32	1024	512
			64	256	128				64	512	256
		64	8	1024	512			64	8	2048	1024
			16	512	256				16	1024	512
			32	256	128				32	512	256
			64	128	64				64	256	128
	1/1	--	--	512*2	--		1/1	--	--	1024*2	--

TEXT Size	BMPM/RGBM	HS	VS	Number Array (HWords)	Attribute Array (HWords)	TEXT Size	BMPM/RGBM	HS	VS	Number Array (HWords)	Attribute Array (HWords)
2048x 1024	0/0	32	32	2048	1024	2048x 2048	32	32	4096	2048	
			64	1024	512			64	2048	1024	
			128	512	256			128	1024	512	
			256	256	128			256	512	256	
		64	32	1024	512		64	32	2048	1024	
			64	512	256			64	1024	512	
			128	256	128			128	512	256	
			256	128	64			256	256	128	
		128	32	512	256		128	32	1024	512	
			64	256	128			64	512	256	
			128	128	64			128	256	128	
			256	64	32			256	128	64	
		256	32	256	128		256	32	512	256	
			64	128	64			64	256	128	
			128	64	32			128	128	64	
			256	32	16			256	64	32	
1/1	--	--	1024*2	--	--	1/1	--	--	2048*2	--	--
4096x 2048	0/0	32	32	8192	4096	4096x 4096	32	32	16384	8192	
			64	4096	2048			64	8192	4096	
			128	2048	1024			128	4096	2048	
			256	1024	512			256	2048	1024	
		64	32	4096	2048		64	32	8192	4096	
			64	2048	1024			64	4096	2048	
			128	1024	512			128	2048	1024	
			256	512	256			256	1024	512	
		128	32	2048	1024		128	32	4096	2048	
			64	1024	512			64	2048	1024	
			128	512	256			128	1024	512	
			256	256	128			256	512	256	
		256	32	1024	512		256	32	2048	1024	
			64	512	256			64	1024	512	
			128	256	128			128	512	256	
			256	128	64			256	256	128	
1/1	--	--	2048*2	--	--	1/1	--	--	4096*2	--	--

When REGM is 1 and BMPM=RGBM=TX_DIRECT=0, the attribute array is not necessary.

When WALLM is 1, only a single word number/attribute array is needed under each mode.

Please refer to [section of TEXT Character Number](#) for more details about the number/attribute array.

13.3.4 TEXT Mode Control

The mode of TEXT layer can be configured independently for all four TEXT layers. The register below is used to control the mode of each TEXT layer.

P_PPU_TEXTN_CTRL								0x9302004C	0x93020064	0x93020014	TEXT N Control Register (N=1,2,3,4)	
Bit	23	22	21	20	19	18	17	16				
Function	-							-	INTP			
Default	0	0	0	0	0	0	0	0				
Bit	15	14	13	12	11	10	9	8				
Function	TXN_BLDLVL						TXN_BLDMODE	TXN_BLD				
Default	0	0	0	0	0	0	0	0				
TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP						
0	0	0	0	0	0	0	0	0	0	0	0	

The following table shows the TEXT mode control definition.

TXN_MODE	TEXT 1	TEXT 2	TEXT 3	TEXT4
2'b00	2D mode	2D mode	2D mode	2D mode
2'b01	Rotate mode	Rotate mode	Rotate mode	Rotate mode
2'b10	NA	NA	Virtual 3D mode	NA
2'b11	NA	NA	NA	NA

If a programmer intends to enable the HCMP and VCMP function on TEXT 1 and TEXT 2, the following register must be set to "1".

P_PPU_ENABLE								0x930201FC	PPU Control Register				
Bit	31	30	29	28	27	26	25	24					
Function	-							DEFEN	TFTLB				
Default	0	0	0	0	0	0	0	0					
Bit	23	22	21	20	19	18	17	16					
Function	YUV_TYPE				LB	TFT_SIZE							
Default	0	0	0	0	0	0	0	0					



15	14	13	12	11	10	9	8
SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO	SPR25D	FB_FORMAT	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
FB_EN	-	VGA_NOINTL	VGA_EN	TX_BOTUP	TX_DIRECT	CH0_BLK	PPU_EN
0	0	0	0	0	0	0	1

HVCMP_DIS is 1 => Enable rotation for all TEXTs or Enable virtual 3D for TEXT 3.

HVCMP DIS is 0 => TEXT 1 and TEXT 2 can do HCMP or VCMP function.

The following table shows the mode control definition to each TEXT when HVCMP_DIS is “0”.

TXN_MODE	TEXT 1	TEXT 2	TEXT 3	TEXT4
2'b00	2D mode	2D mode	2D mode	2D mode
2'b01	HCMP mode	NA	Rotate mode	Rotate mode
2'b10	VCMP mode	VCMP mode	Virtual 3D mode	NA
2'b11	HCMP+VCMP	NA	NA	NA

Note: The TEXT 1 HCMP mode will be disabled automatically when TEXT 3 virtual 3D mode is turned on.

To enable a TEXT, TXN_EN must be set to “1”. Before enabling a TEXT layer, make sure that all the TEXT parameters are set to the correct values.

13.3.4.1 TEXT 2D mode

The 2D mode is supported by all 4 TEXTs; the TEXT coordinate under this mode is controlled by the following register.

P_PPU_TEXTN_X_P 0x93020040 0x93020058 0x93020000 TEXT N Position X Register

POSITION 0x93020020 (**N=1,2,3,4**)

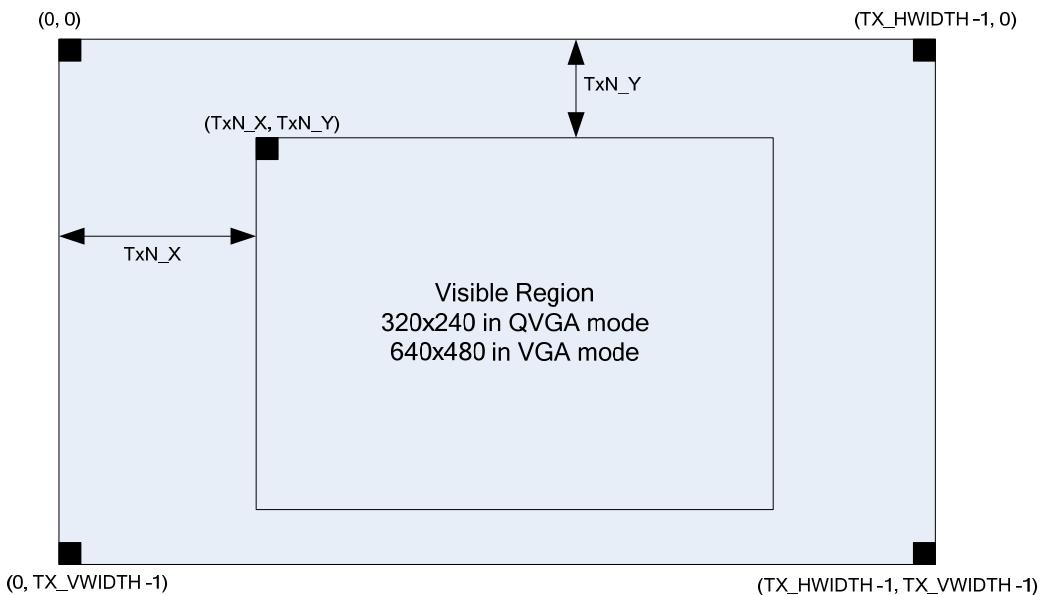
Bit	15	14	13	12	11	10	9	8
Function	TXN_X_POSITION[11:8]							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	TXN_X_POSITION[7:0]							
	0	0	0	0	0	0	0	0

P_PPU_TEXTN_Y_P 0x93020044 0x9302005C 0x93020004 **TEXT N Position Y Register**

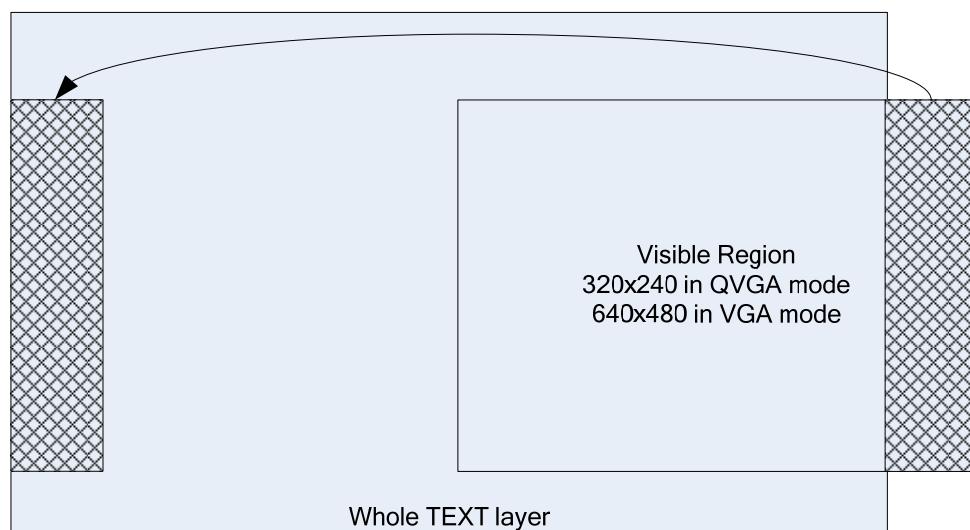
POSITION 0x93020024 (**N=1,2,3,4**)

7	6	5	4	3	2	1	0
TXN_Y_POSITION[7:0]							
0	0	0	0	0	0	0	0

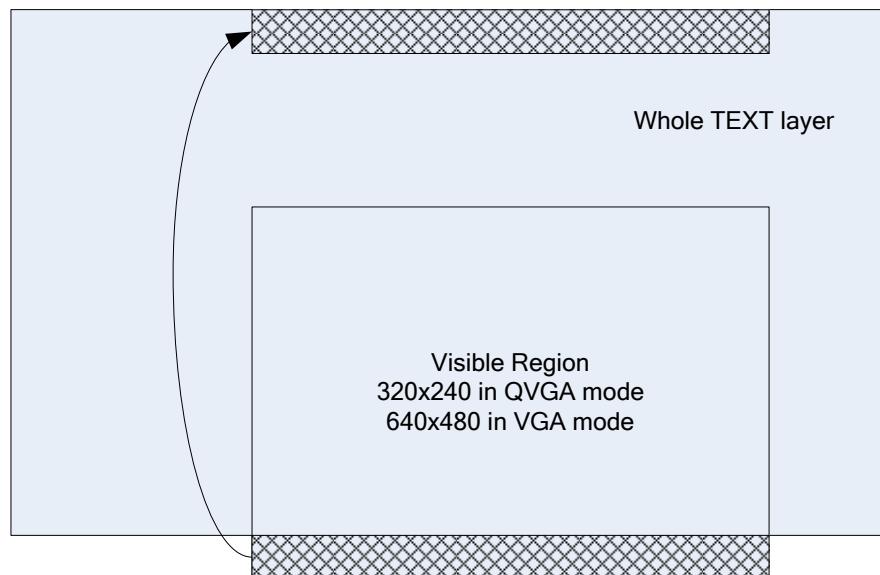
TXN_X_POSITION and TXN_Y_POSITION is unsigned 12-bit value which indicates a position of TEXT to be placed on the top-left of the screen. The following diagram shows the definition of TXN_X_POSITION (TxN_X) and TXN_Y_POSITION (TxN_Y).



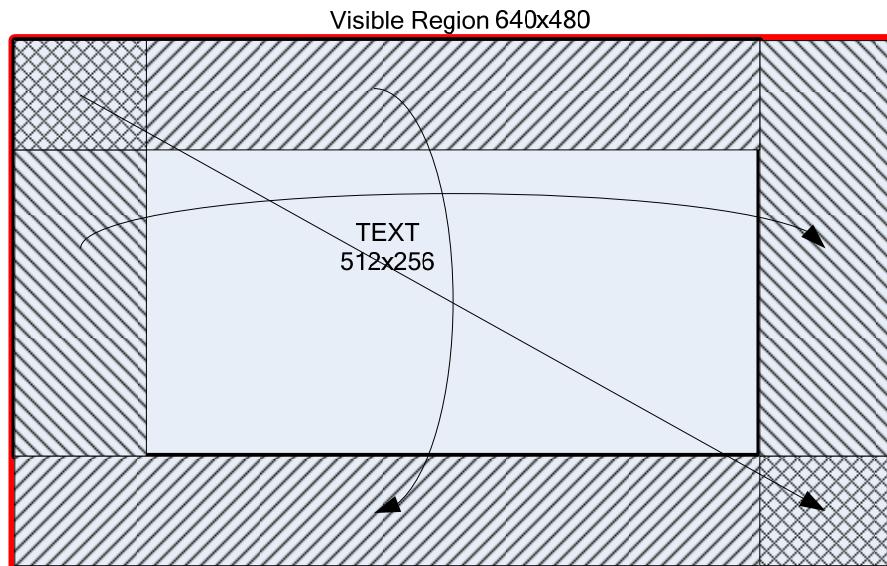
In 2D mode, the visible region will loop back to left/top side while reaching the right/bottom boundary. The following diagram shows the result when visible region hits the right boundary.



The following diagram indicates the result when visible region hits the bottom boundary.



For a special condition under the VGA resolution, if the TEXT size is set to 512x256 or 512x512 which is smaller than the screen size, the TEXT engine will automatically duplicate the TEXT pattern when visible region is out-of the TEXT boundary; see the following figure.



13.3.4.2 TEXT HCMP Mode

The horizontal compression (HCMP) function is used to emulate the distance effect. By adjusting the compression value of each line, programmer can generate a 3D-like pattern from a 2D TEXT layer. The compression amount of each line is stored in RAM P_PPU_TEXT_HCMP_VALUEN. Programmer must initialize this RAM before enabling the HCMP function. This table can be updated only in the vertical blanking time.

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_TEXT_HCMP_VALUE0	0x93020800																HCMP_VALUE0 [7:0]
P_PPU_TEXT_HCMP_VALUE1	0x93020804																HCMP_VALUE1 [7:0]
...
P_PPU_TEXT_HCMP_VALUE238	0x93020BB 8																HCMP_VALUE238 [7:0]
P_PPU_TEXT_HCMP_VALUE239	0x93020BB C																HCMP_VALUE239 [7:0]

The line here means the line in the visible region. In VGA mode, both even and odd lines share the same HCMP_VALUE. For example, line 0 and line 1 use HCMP_VALUE0 and line 2 and line 3 use HCMP_VALUE1.

The following equation shows the compression ratio according to different HCMP_VALUE.

$$\text{Compression Ratio} = \text{HCMP_VALUE} / 128, (\text{HCMP_VALUE from } 1 \sim 128)$$

When HCMP_VALUE is 0 or greater than 128, the HCMP engine cannot work correctly. The compression center is the center of the visible region.

Under VGA mode, the HCMP behavior is slightly different compared to QVGA mode, the following equations are used to describe the behavior of HCMP under VGA mode.

1. Compression Ratio = HCMP_VALUE / 64, (HCMP_VALUE from 1 ~ 128)
2. Real compression center = Current center of visible region(before compression) - 160 pixels.
3. Horizontal movement offset = (4 x TX_HVOFFSET [8:0])

In order to simplify the program flow under HCMP mode and match the behavior with that in QVGA, following register is used to control the behavior under HCMP mode.

PPU Misc Control Register							
Bit	23	22	21	20	19	18	17
Function			TFT_3D				SBMP_MODE
Default	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-		TFTVTQ	-	INTP_MODE		NEW_CMP	TXT_ALPHA
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
	DUAL_BLD			SPR_RGB	TXT_RGB	FB_LOCK	
0	0	0	0	0	0	0	0

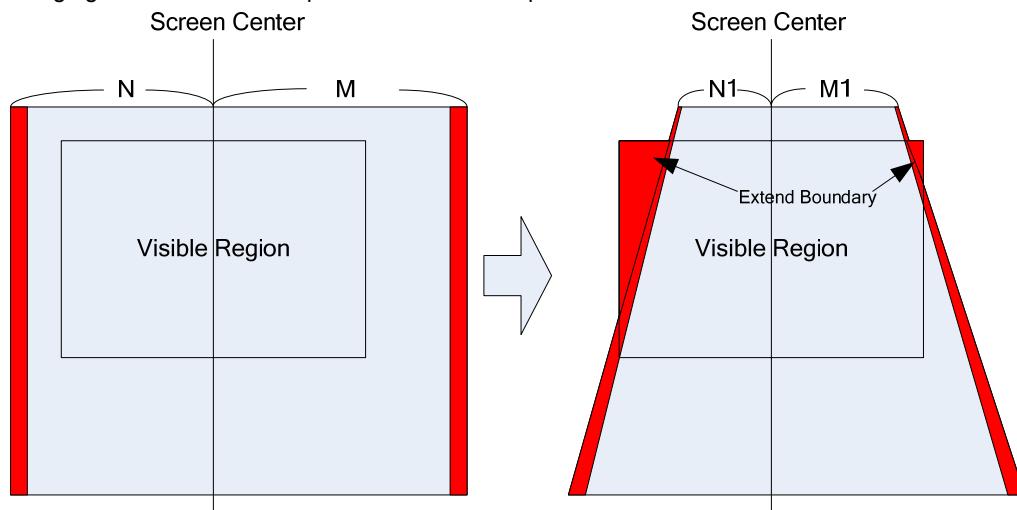
NEW_CMP is 0 => Keep old behavior described above under VGA mode.

NEW_CMP is 1 => The VGA HCMP/VCMP behavior will follow the following equation.

1. Compression ratio: HCMP_Value / 128, (HCMP_Value from 1 ~ 128)

2. Initial H offset: 0 pixels.
3. For MVE: $4 \times Tx_Hvoffset[9:0]$

Under HCMP mode, the loop function is disabled. When HCMP engine reaches the left or right boundary of TEXT, it will extend the boundary pixel of the TEXT to the end of visible region. The following figure shows an example of horizontal compression.



$$N1/N = M1/M = \text{Compression Ratio}$$

13.3.4.3 TEXT VCMP Mode

The TEXT vertical compression (VCMP) mode is to simulate the distance effect combining with HCMP effect. VCMP effect can work alone for a special function. The following registers are the VCMP control register.

P_PPU_VCOMP_VALUE 0x93020070								TEXT VCOMP Value Register								
Bit	15	14	13	12	11	10	9	8								
Function									[8]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
VCOMP_VALUE[7:0]																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_PPU_VCOMP_OFFSET 0x93020074								TEXT VCOMP Offset Register								
Bit	15	14	13	12	11	10	9	8								
Function									[8]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
VCOMP_OFFSET[7:0]																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_PPU_VCOMP_STEP 0x93020078								TEXT VCOMP Offset Register								
Bit	15	14	13	12	11	10	9	8								
Function									[8]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
VCOMP_STEP[7:0]																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

VCMP engine works around the top of the visible region + VCOMP_OFFSET. The following equation is used to compute the effect line (Y') on the TEXT at screen layer Y.

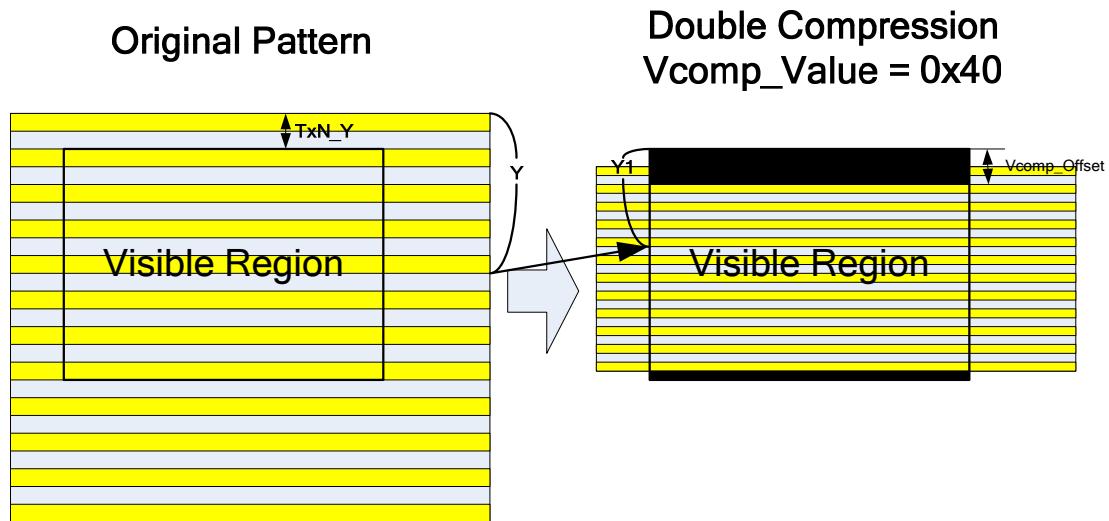
$$Y' = TxN_Y + (Y1 - VCOMP_OFFSET) * (VCOMP_VALUE / 0x20).$$

Suppose TxN_Y and VCOMP_OFFSET are 0 and VCOMP_VALUE is 0x20, where Y' = Y means no compression. When VCOMP_VALUE is 0x40 → Y' = 2 * Y, which means double compression. So the compress ratio can be expressed by the following equation.

$$\text{Compress Ratio} = (0x20 / VCOMP_VALUE).$$

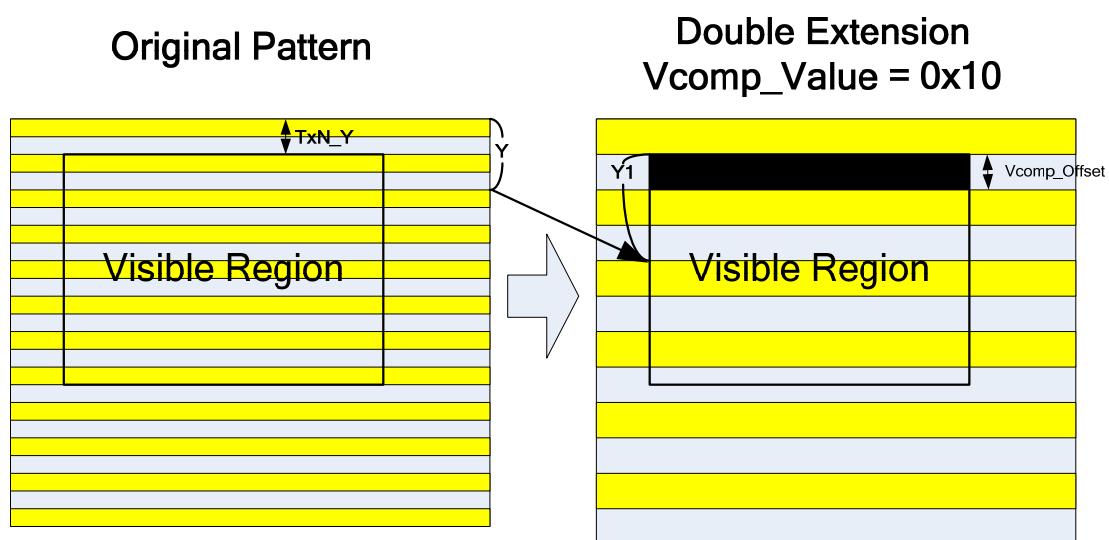
where VCOMP_VALUE > 0x20 means compression, and VCOMP_VALUE < 0x20 means extension.

The following diagram shows an example of compression.



$$Y = TxN_Y + (Y1 - VComp_Offset) * VComp_Value / 0x20$$

The following diagram is an example of extension.

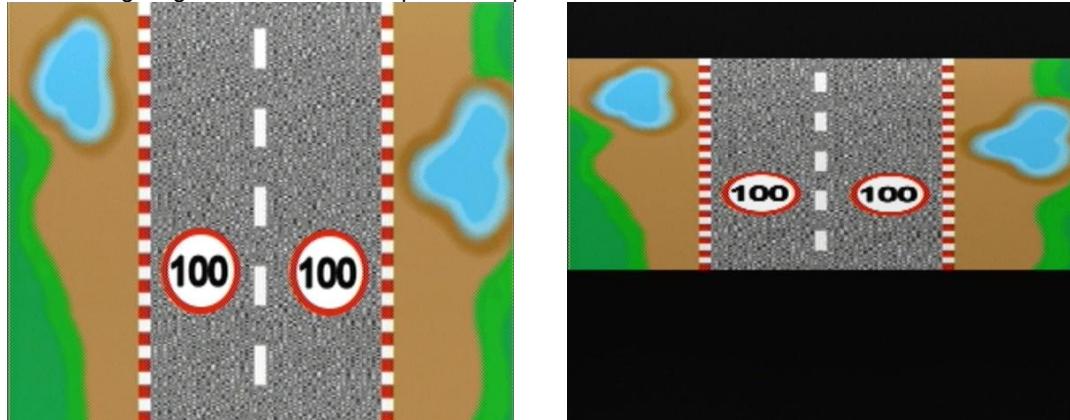


$$Y = TxN_Y + (Y1 - VComp_Offset) * VComp_Value / 0x20$$

The output result will become transparent when the following two conditions are met:

1. When screen layer < VCOMP_OFFSET.
2. Compression result is out of the bottom boundary of TEXT.

The following diagrams show an example of compression.



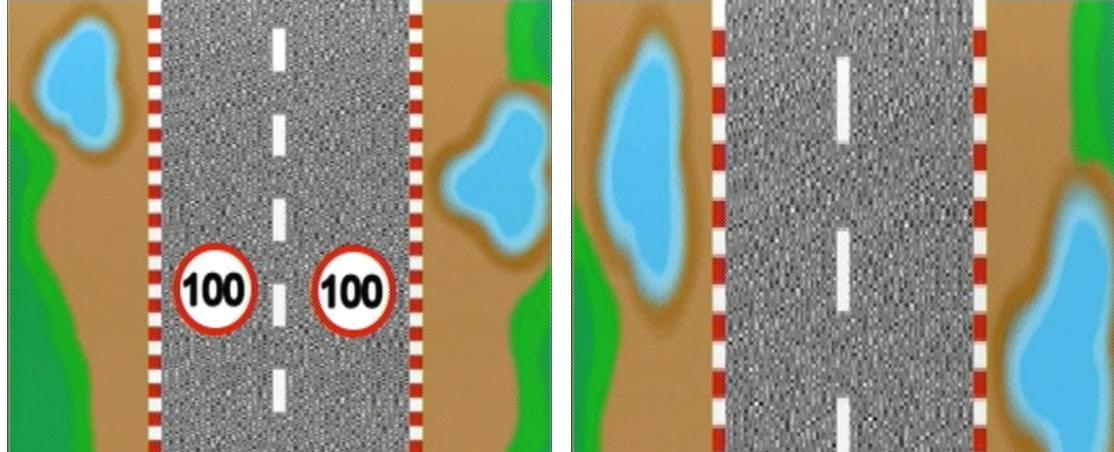
Original TEXT

VCMP_VALUE = 0X40

VCMP_STEP= 0X00

VCMP_OFFSET = 0X20

The following diagrams show an example of extension.



Original TEXT

VCMP_VALUE = 0X10

VCMP_STEP= 0X00

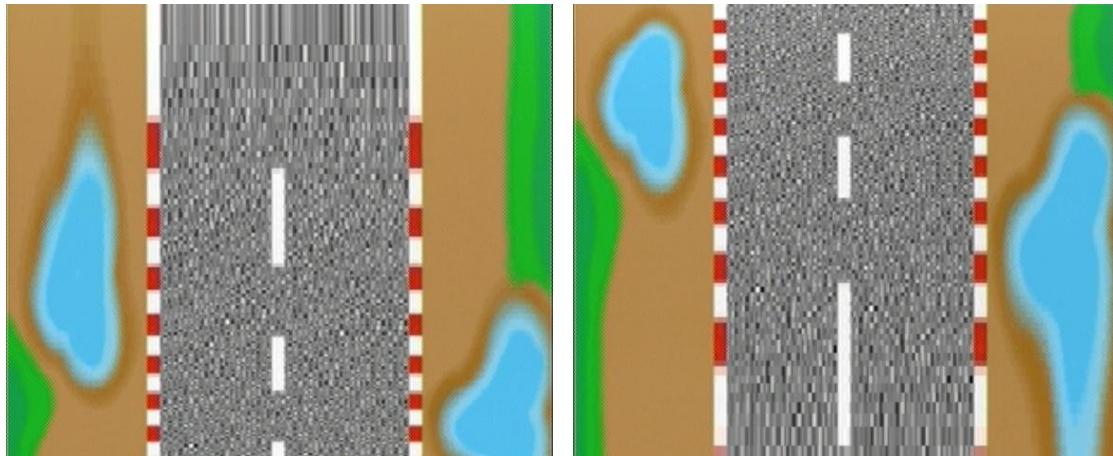
VCMP_OFFSET = 0X00

In reality, far objects (at the top of screen) are smaller than closer objects (at the bottom of screen). To simulate this behavior, we use VCOMP_STEP register to control the VCOMP_VALUE at each line. The following equation is used to compute VCOMP_VALUE of each line.

$$\text{VCOMP_VALUE}_Y = \text{VCOMP_VALUE}_0 + (Y * \text{VCOMP_STEP}) / 256. \text{ (for QVGA)}$$

$$\text{VCOMP_VALUE}_Y = \text{VCOMP_VALUE}_0 + (Y * \text{VCOMP_STEP}) / 512. \text{ (for VGA)}$$

Where VCOMP_VALUE0 is the register setting of P_VCOMP_VALUE; Y is the screen line number, and VCOMP_VALUEY is the VCOMP_VALUE of line Y. VCOMPSTEP has the range from -256 ~ 255. The following diagram shows an example of VCOMP_STEP.

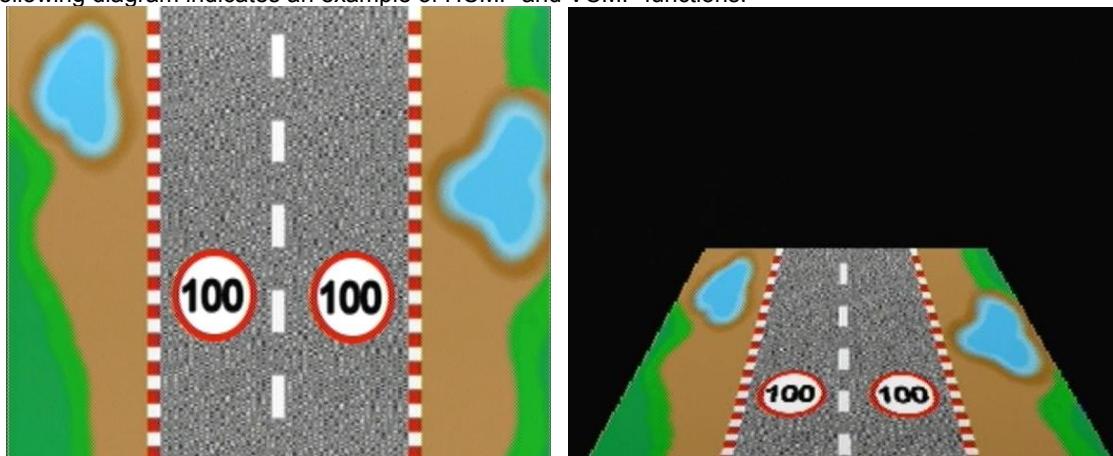


VCMP_VALUE = 0X00
VCMP_STEP= 0X16
VCMP_OFFSET = 0X00

VCMP_VALUE = 0X20
VCMP_STEP=0X1F0 = -16
VCMP_OFFSET = 0X00

13.3.4.4 TEXT HCMP+VCMP Mode

Combining HCMP and VCMP functions, programmer can simulate the distance effect on a TEXT. The following diagram indicates an example of HCMP and VCMP functions.



Original TEXT

Horizontal + vertical compression

13.3.4.5 TEXT Rotate Mode

The GP12 supports arbitrary angle rotation and zoom in/out function on all TEXTs. The following registers are used to control the rotating angle and zoom in/out factor.

P_PPU_TEXTN_COSINE								0x93020348	0x93020358	0x93020800	TEXT N Cosine0 Register (N=1,2,3,4)																												
P_PPU_TEXTN_SINE								0x9302034C								0x9302035C								0x93020804								TEXT N Sine0 Register (N=1,2,3,4)							
Bit	15	14	13	12	11	10	9	8	0x930200A0								0x930200A4								0x930200A4														
Function	-								TXN_COSINE0 [12:8]																														
Default	0	0	0	0	0	0	0	0	TXN_COSINE0 [7:0]																0	0	0	0											
	7	6	5	4	3	2	1	0	TXN_COSINE0 [7:0]																0	0	0	0											
	0	0	0	0	0	0	0	0	TXN_SINE0 [12:8]																0	0	0	0											
	7	6	5	4	3	2	1	0	TXN_SINE0 [7:0]																0	0	0	0											
	0	0	0	0	0	0	0	0	TXN_SINE0 [7:0]																0	0	0	0											

The TX1/2/3/4_COSINE [12:0] and TX1/2/3/4_SINE [12:0] have the following indications

X[12:0]			
X[12] == 0		X[12] == 1	
X[11:0] == 0x400 1.00	X[11:0] == 0xC00 -1.00	X[11:0] == 0x400 1.00 * 64	X[11:0] == 0xC00 -1.00 * 64
X[11:0] == 0x200 0.50	X[11:0] == 0xA00 -0.50	X[11:0] == 0x200 0.50 * 64	X[11:0] == 0xA00 -0.50 * 64

where X [12:0] can be TX1/2/3/4_COSINE [12:0] or TX1/2/3/4_SINE [12:0], X [12] means 64 times boost, X [11:0] is a signed 12 bits value with range from -1.999 ~ 1.999. By using the cosine and sine value we can do arbitrary angle rotation and zoom-in/out from infinite zoom-in to 1/128 zoom-out.

Not only the cosine and sine values, but we also need to know the rotate center and location of rotation center on the screen. The following registers are used to control the rotation center and location.

P_PPU_TEXTN_X_P 0x93020040 0x93020058 0x93020000 TEXT N Position X Register
OSITION
0x93020020
(N=1,2,3,4)

Bit	15	14	13	12	11	10	9	8
Function	TXN_X_POSITION[11:8]							

Default

0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TXN_X_POSITION[7:0]							

0 0 0 0 0 0 0 0 0

P_PPU_TEXTN_Y_P 0x93020044 0x9302005C 0x93020004 TEXT N Position Y Register
OSITION
0x93020024
(N=1,2,3,4)

Bit	15	14	13	12	11	10	9	8
Function	TXN_Y_POSITION[11:8]							

Default

0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TXN_Y_POSITION[7:0]							

0 0 0 0 0 0 0 0 0

Note: The position registers are used to control the rotate center.

P_PPU_TEXTN_X_O 0x93020340 0x93020350 0x93020008 TEXT N Offset X Register
FFSET
0x93020028
(N=1,2,3,4)

Bit	15	14	13	12	11	10	9	8
Function	TXN_X_OFFSET [11:8]							

Default

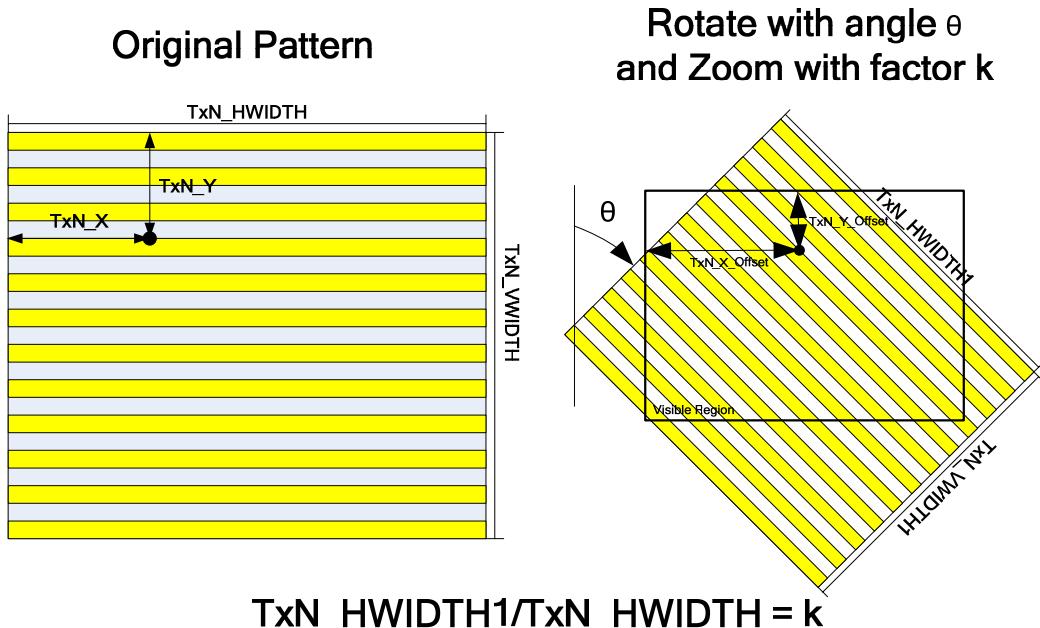
0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TXN_X_OFFSET [7:0]							

0 0 0 0 0 0 0 0 0

Note: The offset registers control the screen location of rotate center.

The following figures show how these parameters work on a TEXT.



If programmer intend to perform a rotation of an angle, θ (clockwise), and zoom with k factor; then, the $1/k*\cos\theta$ should be filled in TXN_COSINE, and the $1/k*\sin\theta$ should be filled in TXN_SINE as well.

The following table shows some examples of rotation and zoom-in/out.

Zoom	Rotate (Clockwise)	TXN_COSINE	TXN_SINE
No Zoom Effect	45 degree	$\text{Cos}(45^\circ) = 0.707 = 0x2D4$	$\text{Sin}(45^\circ) = 0.707 = 0x2D4$
No Zoom Effect	315 degree	$\text{Cos}(315^\circ) = 0.707 = 0x2D4$	$\text{Sin}(315^\circ) = -0.707 = 0xD2C$
Zoom-in with 2 times	0 degree	$(1/2)*\text{Cos}(0^\circ) = 0.500 = 0x0200$	$(1/2)*\text{Sin}(0^\circ) = 0.000 = 0x0000$

Zoom	Rotate (Clockwise)	TXN_COSINE	TXN_SINE
Zoom-in with 128 times	0 degree	$(1/128) * \text{Cos}(0^\circ) = 0.0078125$ = 0x0008	$(1/128) * \text{Sin}(0^\circ) = 0.000$ = 0x0000
Zoom-out with 1/2 times	0 degree	$(1/(1/2)) * \text{Cos}(0^\circ) = 2.000$ (>1.999) = 0x1020	$(1/(1/2)) * \text{Sin}(0^\circ) = 0.000$ = 0x0000
Zoom-out with 1/128 times	315 degree	$(1/(1/128)) * \text{Cos}(315^\circ)$ = 90.510 (>1.999) = 0x15A8	$(1/(1/128)) * \text{Sin}(315^\circ)$ = -90.510 (<-1.999) = 0x1A58

The resolution of zoom-out will become lower if the zoom-out factor is greater than 2, but it is still acceptable because the zoom-out will make the TEXT becoming smaller and the TEXT quality will become hard to be detected by human's vision.

In rotate mode, the loop function is disabled. If the TEXT boundary reaches to the end, screen will become transparent at that region.

The following diagrams are some rotate examples of TEXT.



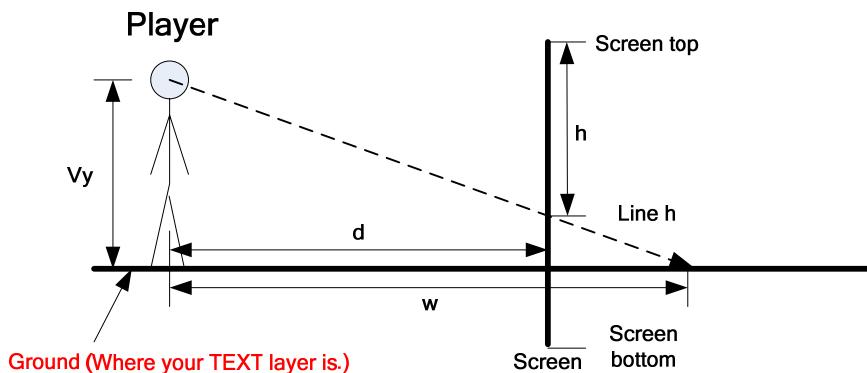
Rotate 45°, No Zoom Effect

Rotate 45°, Zoom-out 1/2 Effect

Rotate 45°, Zoom-in 1.5 Effect

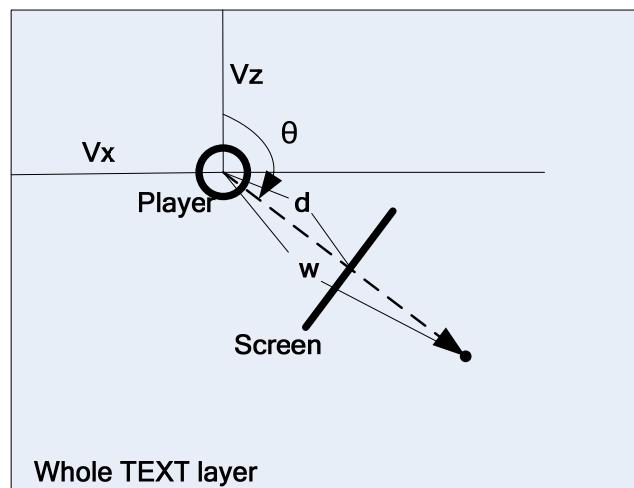
13.3.4.6 TEXT Virtual 3D Mode

The virtual 3D mode of TEXT is used to simulate a 3D environment. The principle of virtual 3D effect is very simple. Please look at the following diagram.



When we know Vy , d , h , we can find
 $w = Vy * (d/h) = (Vy*d)/h$

Above figure is plotted parallel to the TEXT layer and the screen. If we change the view point to orthogonal to the TEXT, we will find the following diagram.



Rotate Center
 $Vx = Tx3_X_Position$
 $Vz = Tx3_Y_Position$

Unlike pure rotate mode, each screen line can have independent zoom factor ($r = Vy*d/h$) in virtual 3D mode. The following SRAM is used to configure $r*\cos\theta$ and $(-r*\sin\theta)$ of each screen line.

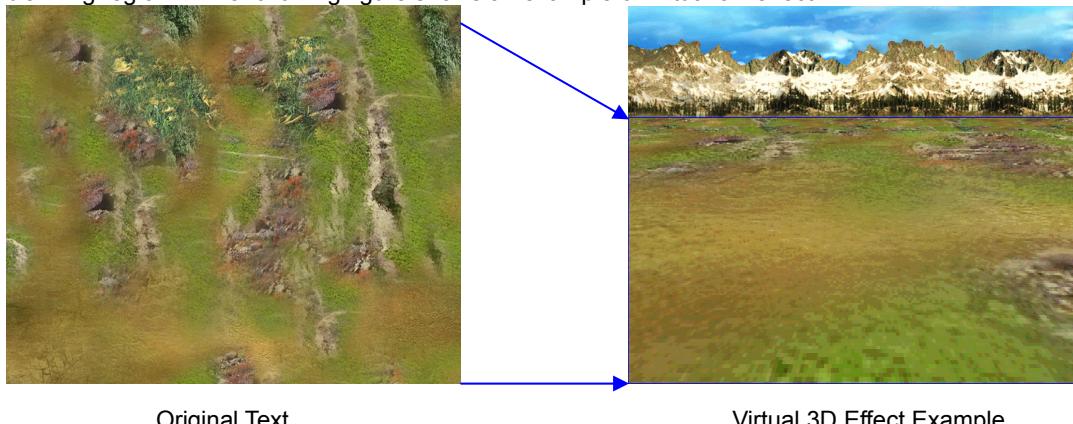
P_PPU_TEXT3_COSN								0x93020800+8*N								TEXT 3 Cosine N Register								
Bit	15	14	13	12	11	10	9	8																
Function	-							TX3_COSINEN [12:8]																
Default	0	0	0	0	0	0	0	0																

7	6	5	4	3	2	1	0
TX3_COSINEN [7:0]							
0	0	0	0	0	0	0	0

P_PPU_TEXT3_SINN 0x93020804+8*N TEXT 3 Sine N Register								
Bit	15	14	13	12	11	10	9	
Function	-	TX3_SINEN [12:8]						
Default	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
TX3_SINEN [7:0]							
0	0	0	0	0	0	0	0

The format of TX3_COSN and TX3_SINN are the same as that in rotate mode. In VGA mode, both even and odd lines share the same parameter. Programmer should fill $-r * \sin\theta$ in TX3_SINN since the hardware will invert it in correct angle. These parameters should be changed only in the vertical blanking region. The following figure shows an example of virtual 3D effect.



Original Text

Virtual 3D Effect Example

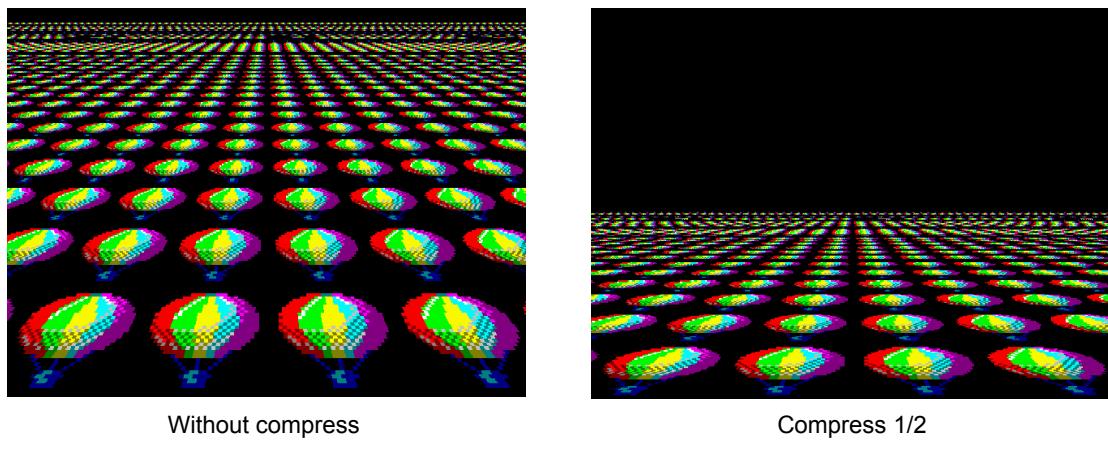
For some special applications, when users intend to compress the output result of the virtual 3D TEXT at the Y axis, the following register can be used for the Y-axis compress parameter.

P_PPU_Y25D_COMPRESS 0x93020104 TEXT3 Y Compress Register								
Bit	7	6	5	4	3	2	1	
Function	-	Y_COMPRESS [5:0]						
Default	0	0	0	0	0	0	0	

Y_COMP RESS	Ratio	Y_COMP RESS	Ratio	Y_COMP RESS	Ratio	Y_COMP RESS	Ratio
0x00	NA	0x10	16/16	0x20	16/32	0x30	16/48
0x01	16/1	0x11	16/17	0x21	16/33	0x31	16/49
0x02	16/2	0x12	16/18	0x22	16/34	0x32	16/50
0x03	16/3	0x13	16/19	0x23	16/35	0x33	16/51

Y_COMP RESS	Ratio	Y_COMP RESS	Ratio	Y_COMP RESS	Ratio	Y_COMP RESS	Ratio
0x04	16/4	0x14	16/20	0x24	16/36	0x34	16/52
0x05	16/5	0x15	16/21	0x25	16/37	0x35	16/53
0x06	16/6	0x16	16/22	0x26	16/38	0x36	16/54
0x07	16/7	0x17	16/23	0x27	16/39	0x37	16/55
0x08	16/8	0x18	16/24	0x28	16/40	0x38	16/56
0x09	16/9	0x19	16/25	0x29	16/41	0x39	16/57
0x0A	16/10	0x1A	16/26	0x2A	16/42	0x3A	16/58
0x0B	16/11	0x1B	16/27	0x2B	16/43	0x3B	16/59
0x0C	16/12	0x1C	16/28	0x2C	16/44	0x3C	16/60
0x0D	16/13	0x1D	16/29	0x2D	16/45	0x3D	16/61
0x0E	16/14	0x1E	16/30	0x2E	16/46	0x3E	16/62
0x0F	16/15	0x1F	16/31	0x2F	16/47	0x3F	16/63

The following figures show an example that sets Y_COMPRESS to 0x20, which performs a one-half compress on the Y axis. At the same time, the $r * \cos\theta$ and $r * \sin\theta$ need to do the same modification to form the one-half compress. For example, the parameter of line 0 moves to line 120, line 2 moving to line 121, ..., line 236 moving to line 238, and line 238 moving to line 239.



13.3.5 TEXT Horizontal Movement

The TEXT horizontal movement function is a special function that can add an offset on the TxN_X at each screen line. The following registers configure the horizontal movement function.

P_PPU_TEXTN_CTRL								0x9302004C 0x93020064 0x93020014	TEXT N Control Register
								0x93020034	(N=1,2,3,4)
Bit	15	14	13	12	11	10	9	8	
Function			-				-		INTP
Default	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8
Function	TXN_BLDLVL						TXN_BLDMODE	TXN_BLD
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP	

0 0 0 0 0 0 0 0

TXN_MVE = 0 => Disable horizontal movement function.

TXN_MVE = 1 => Enable horizontal movement function.

P_PPU_TEXT_H_OFFSETN								0x93020400+4*N								TEXT HV Offset Register																								
Bit	15	14	13	12	11	10	9	8																																
Function	-								TX_HOFFSET [9:8]																															
Default	0 0 0 0 0 0 0 0																																							

7	6	5	4	3	2	1	0
TX_HOFFSET [7:0]							

0 0 0 0 0 0 0 0

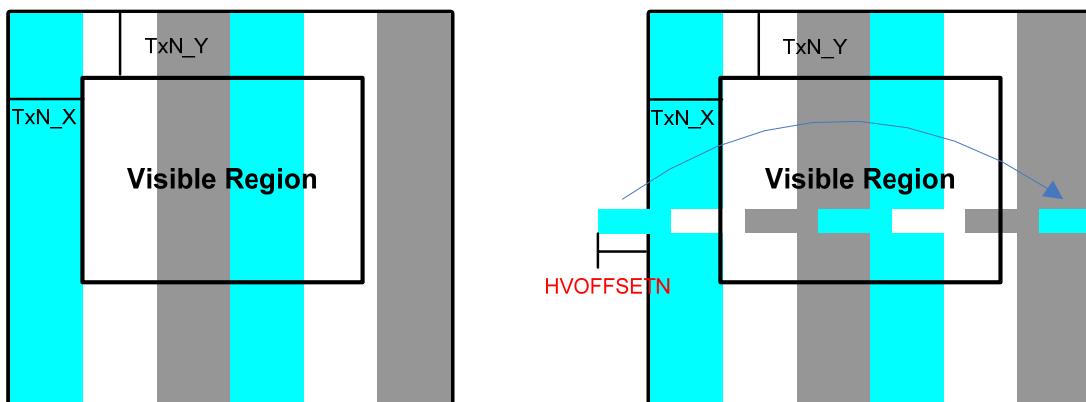
All TEXTs share the same offset value, but only 2D mode and HCMP mode can use this function. The meaning of TX_HOFFSET varies in 2D mode and HCMP mode. We will describe the details in the following section.

13.3.5.1 Horizontal Movement at 2D Mode

In 2D mode, TX_HOFFSET is an unsigned value. In QVGA mode, only TX_HOFFSET [8:0] is used.

In VGA mode, TX_HOFFSET [9:0] is used, but both even and odd lines share the same offset value.

The following figure shows an example of horizontal movement at 2D mode.

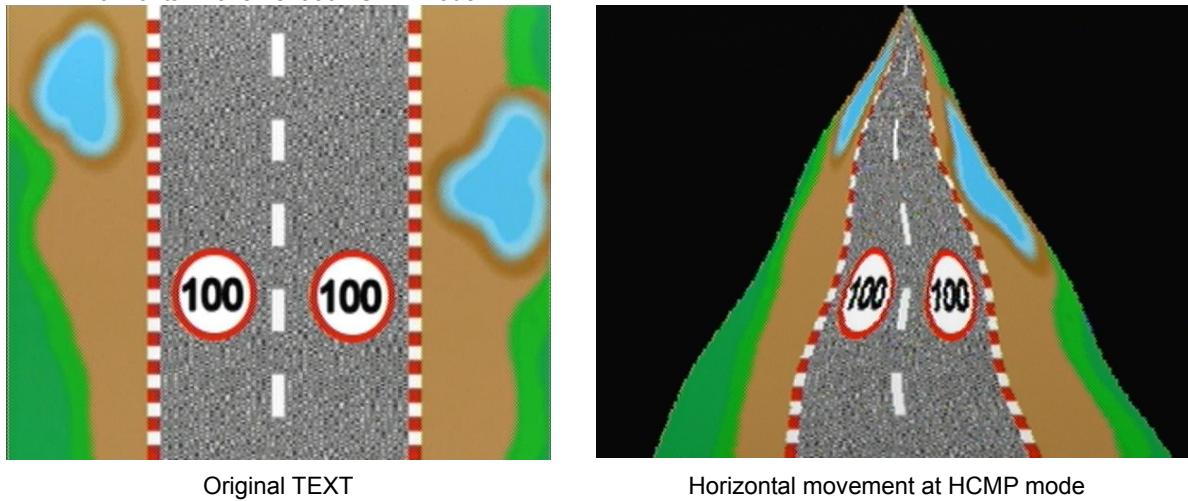


Add HOFFSET will move text line to left.

The loop behavior is as that in 2D mode; it will repeat the left pattern at the right side.

13.3.5.2 Horizontal Movement at HCMP mode

In HCMP mode, the horizontal movement is used to control the location of compress center. HVOFFSET becomes signed 9-bit value in this mode on both QVGA and VGA modes. In VGA mode, the even line and odd line share the same offset value. The following figure shows an example of horizontal movement at HCMP mode.



13.3.6 TEXT Calculation Order

The TEXT calculation sequence has two formats: bottom-up and top-down; see the following register for details.

P_PPU_ENABLE								0x930201FC								PPU Control Register								
Bit	31	30	29	28	27	26	25	24																
Function	-							DEFEN								TFTLB								
Default	0 0 0 0 0 0 0 0																							
	23	22	21	20	19	18	17	16																
	-	YUV_TYPE							LB								TFT_SIZE							
	0	0	0	0	0	0	0	0																
	15	14	13	12	11	10	9	8																
	SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO	SPR25D	FB_FORMAT	0																
	0	0	0	0	0	0	0	0																
	7	6	5	4	3	2	1	0																
	FB_EN	-	VGA_NOINTL	VGA_EN	TX_BOTUP	TX_DIRECT	CH0_BLK	PPU_EN																
	0	0	0	0	0	0	0	1																

TX_BOTUP is 0 => Top-down calculation mode (default).

TX_BOTUP is 1 => Bottom-up calculation mode.

The following table shows how these two modes work.

Cal. Order	TX_BOTUP = 0 (default)	TX_BOTUP = 1
1	TEXT at depth 3	TEXT at depth 0
2	TEXT at depth 2	TEXT at depth 1
3	TEXT at depth 1	TEXT at depth 2
4	TEXT at depth 0	TEXT at depth 3

When multiple TEXTs are at the same depth, the sequence becomes TEXT1 → TEXT2 → TEXT3 → TEXT4. The later TEXT calculation will overwrite the previous TEXT calculation. For example, if TEXT1 and TEXT2 are both at depth0, the TEXT2 will overwrite the TEXT1.

The calculation sequence of TEXTs is useful when the bandwidth is not enough or TEXT blending mode is turned on. The following table shows suggested value of TX_BOTUP under various conditions.

Blending	Bandwidth enough	Bandwidth not enough
Disable	TX_BOTUP has no effect on the final result, nothing will lose.	TX_BOTUP = 0 will get better result.
Enable	TX_BOTUP = 1 will get better result.	Both TX_BOTUP = 0 or = 1 will not get better result. But if frame buffer mode is turned on, TX_BOTUP = 1 will get the exactly correct result.

13.3.7 TEXT Character 0 Auto Transparent Mode

GP12 supports an auto-transparent mode for TEXT layer with character mode. When this mode is enabled, the TEXT will automatically treat character number 0 as transparent character without read data by the system bus. This mode is useful when most parts of the TEXT are transparent which will save a lot of bandwidths. This mode can be enabled by the following register.

P_PPU_ENABLE	0x930201FC								PPU Control Register								
Bit	31	30	29	28	27	26	25	24									
Function	-								DEFEN	TFTLB							
Default	0	0	0	0	0	0	0	0									
								23	22	21	20	19	18	17	16		
								-	YUV_TYPE				LB	TFT_SIZE			
								0	0	0	0	0	0	0	0	0	0
								15	14	13	12	11	10	9	8		
								SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO	SPR25D	FB_FORMAT			
								0	0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
FB_EN	-	VGA_NOINTL	VGA_EN	TX_BOTUP	TX_DIRECT	CH0_BLK	PPU_EN
0	0	0	0	0	0	0	1

CH0_BLK is 0 => Normal mode, character number 0 is like normal character (default).

CH0_BLK is 1 => Character 0 transparent mode, character number 0 is transparent.

When CH0_BLK is set to 1, it affects not only the relative addressing mode, but also the direct addressing mode and RGB 32 bits addressing mode in character mode. When relative mode is used, the character number, bit 15 to bit 0, is used to determine the character number 0. When direct RGB mode is used, the address bit 31 to bit 0 is used to determine the character number 0. This mode will not affect the bitmap mode.

13.4 Sprite Control

13.4.1 Sprite Register Summary

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_SPRITE_SEGMENT	0x930200 88																SEGMENT_SP [15:0]
P_PPU_SPRITE_SEGMENT	0x930200 8A																SEGMENT_SP [31:16]
P_PPU_SPRITE_CTRL	0x930201 08									ZOO MEN	ROTE N	MOSE N	DIR	-	BLDM	COOR	SPEN
P_PPU_SPRITE_CTRL	0x930201 0A								SP_ CMASK	SP_ FRAC	SP_GRP	SP_LS	SP_IN TP	SP_FAR	CDM	EFFEN	
P_PPU_SPRITE_DMA_SOURCE	0x930201 C0																SPDMA_SOURCE [15:0]
P_PPU_SPRITE_DMA_SOURCE	0x930201 C2																SPDMA_SOURCE [31:16]
P_PPU_SPRITE_DMA_TARGET	0x930201 C4	-															SPDMA_TARGET [14:0]
P_PPU_SPRITE_DMA_NUMBER	0x930201 C8		-														SPDMA_NUMBER [11:0]
P_PPU_ENABLE	0x930201 FC	-		SPR WIN		-		25D									-
P_PPU_SPRITE_X0	0x930203 00																SPR_X0 [15:0]
P_PPU_SPRITE_Y0	0x930203 04																SPR_Y0 [15:0]
P_PPU_SPRITE_X1	0x930203 08																SPR_X1 [15:0]

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_SPRITE_Y1	0x9302030C																SPR_Y1 [15:0]
P_PPU_SPRITE_X2	0x93020310																SPR_X2 [15:0]
P_PPU_SPRITE_Y2	0x93020314																SPR_Y2 [15:0]
P_PPU_SPRITE_X3	0x93020318																SPR_X3 [15:0]
P_PPU_SPRITE_Y3	0x9302031C																SPR_Y3 [15:0]
P_PPU_SPRITE_W0	0x93020320																SPR_X0 [9:0]
P_PPU_SPRITE_W1	0x93020324																SPR_Y0 [9:0]
P_PPU_SPRITE_W2	0x93020328																SPR_X1 [9:0]
P_PPU_SPRITE_W3	0x9302032C	SPR_Y3 [7:6]															SPR_X2 [9:0]
P_PPU_SPRITE_W4	0x93020330	SPR_Y3 [9:8]															SPR_X3 [9:0]

13.4.2 Sprite SRAM Summary

When Sprite color dither mode (CDM) is “0” (0x93020108 bit 17), the following format applies to sprite RAM.

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_SPRITE0_CHARNUM	0x93022000																SP0_CHARNUM [15:0]
P_PPU_SPRITE0_X_POSITION	0x93022002																SP0_X_POSITION [9:0]
P_PPU_SPRITE0_Y_POSITION	0x93022004																SP0_Y_POSITION [9:0]
P_PPU_SPRITE0_ATTRIBUTE0	0x93022006	PB	BLD	DEPTH													COLOR
P_PPU_SPRITE0_ATTRIBUTE1	0x93022008	MOSAIC															SP0_CHARNUM [23:16]
P_PPU_SPRITE0_X1	0x9302200A																SP0_X1 [9:0]
P_PPU_SPRITE0_Y3	0x9302200C	SP0_Y3															SP0_X2 [9:0]



NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
_X2	00C	[7:6]															
P_PPU_SPRITE0_X3	0x93022 00E	SP0_Y3 [9:8]		SP0_Y2 [9:6]													SP0_X3 [9:0]
P_PPU_SPRITE1_CHARNUM	0x93022 010							SP1_CHARNUM [15:0]									
P_PPU_SPRITE1_X_POSITION	0x93022 012			SP1_ROTATE [5:0]/SP1_Y1 [5:0]												SP1_X_POSITION [9:0]	
P_PPU_SPRITE1_Y_POSITION	0x93022 014			SP1_ZOOM [5:0]/SP1_Y2 [5:0]												SP1_Y_POSITION [9:0]	
P_PPU_SPRITE1_ATTRIBUTE0	0x93022 016	PB	BLD	DEPTH		PALETTE		VS		HS		FLIP		COLOR			
P_PPU_SPRITE1_ATTRIBUTE1	0x93022 018	MOSAIC		SP1_BLDLVL				SP1_CHARNUM [23:16]									
P_PPU_SPRITE1_X1	0x93022 01A		SP1_Y3 [5:0]					SP1_X1 [9:0]									
P_PPU_SPRITE1_X2	0x93022 01C	SP1_Y3 [7:6]		SP1_Y1 [9:6]				SP1_X2 [9:0]									
P_PPU_SPRITE1_X3	0x93022 01E	SP1_Y3 [9:8]		SP1_Y2 [9:6]				SP1_X3 [9:0]									
...									
P_PPU_SPRITE1023_CHARNUM	0x93025 FF0					SP1023_CHARNUM [15:0]											
P_PPU_SPRITE1023_X_POSITION	0x93025 FF2		SP1023_ROTATE [5:0]/SP1023_Y1 [5:0]					SP1023_X_POSITION [9:0]									
P_PPU_SPRITE1023_Y_POSITION	0x93025 FF4		SP1023_ZOOM [5:0]/SP1023_Y2 [5:0]					SP1023_Y_POSITION [9:0]									
P_PPU_SPRITE1023_ATTRIBUTE0	0x93025 FF6	PB	BLD	DEPTH		PALETTE		VS		HS		FLIP		COLOR			
P_PPU_SPRITE1023_ATTRIBUTE1	0x93025 FF8	MOSAIC		SP1023_BLDLVL				SP1023_CHARNUM [23:16]									
P_PPU_SPRITE1023_X1	0x93025 FFA		SP1023_Y3[5:0]					SP1023_X1 [9:0]									
P_PPU_SPRITE1023_X2	0x93025 FFC	SP1023_Y3 [7:6]		SP1023_Y1 [9:6]				SP1023_X2 [9:0]									



NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_SPRITE1023_X3	0x93025 FFE	SP1023_Y3 [9:8]		SP1023_Y2 [9:6]													SP1023_X3 [9:0]

When Sprite color dither mode (CDM) is "1" (0x93020108 bit 17), the following format applies to sprite RAM.

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_SPRITE0_CHARNUM	0x93022 000																SP0_CHARNUM [15:0]
P_PPU_SPRITE0_X_POSITION	0x93022 002																SP0_X_POSITION [9:0]
P_PPU_SPRITE0_Y_POSITION	0x93022 004																SP0_Y_POSITION [9:0]
P_PPU_SPRITE0_ATTRIBUTE0	0x93022 006	PB	BLD	DEPTH				PALETTE		VS		HS		FLIP		COLOR	
P_PPU_SPRITE0_ATTRIBUTE1	0x93022 008	MOSAIC						SP0_BLDLVL									SP0_CHARNUM [23:16]
P_PPU_SPRITE0_X1	0x93022 00A							SP0_Y3 [5:0]									SP0_X1 [9:0]
P_PPU_SPRITE0_X2	0x93022 00C	SP0_Y3 [7:6]						SP0_Y1 [9:6]									SP0_X2 [9:0]
P_PPU_SPRITE0_X3	0x93022 00E	SP0_Y3 [9:8]						SP0_Y2 [9:6]									SP0_X3 [9:0]
P_PPU_SPRITE0_RGB0L	0x93022 010							SP0_G0 [7:0]									SP0_B0 [7:0]
P_PPU_SPRITE0_RGB0H	0x93022 012	NCD						-									SP0_R0 [7:0]
P_PPU_SPRITE0_RGB1L	0x93022 014							SP0_G1 [7:0]									SP0_B1 [7:0]
P_PPU_SPRITE0_RGB1H	0x93022 016							-									SP0_R1 [7:0]
P_PPU_SPRITE0_RGB2L	0x93022 018							SP0_G2 [7:0]									SP0_B2 [7:0]
P_PPU_SPRITE0_RGB2H	0x93022 01A							-									SP0_R2 [7:0]
P_PPU_SPRITE0_RGB3L	0x93022 01C							SP0_G3 [7:0]									SP0_B3 [7:0]
P_PPU_SPRITE0_RGB3H	0x93022 01E							-									SP0_R3 [7:0]
...
P_PPU_	0x93025																SP511_CHARNUM [15:0]

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
SPRITE511_CHARNUM	FE0																
P_PPU_SPRITE511_X_POSITION	0x93025 FE2																SP511_X_POSITION [9:0]
P_PPU_SPRITE511_Y_POSITION	0x93025 FE4																SP511_Y_POSITION [9:0]
P_PPU_SPRITE511_ATTRIBUTE0	0x93025 FE6	PB	BLD	DEPTH			PALETTE			VS		HS		FLIP		COLOR	
P_PPU_SPRITE511_ATTRIBUTE1	0x93025 FE8	MOSAIC			SP511_BLDLVL											SP511_CHARNUM [23:16]	
P_PPU_SPRITE511_X1	0x93025 FEA			SP511_Y3 [5:0]												SP511_X1 [9:0]	
P_PPU_SPRITE511_X2	0x93025 FEC	SP511_Y3 [7:6]			SP511_Y1 [9:6]											SP511_X2 [9:0]	
P_PPU_SPRITE511_X3	0x93025 FEE	SP511_Y3 [9:8]			SP511_Y2 [9:6]											SP511_X3 [9:0]	
P_PPU_SPRITE511_RGB0L	0x93025 FF0				SP511_G0 [7:0]											SP511_B0 [7:0]	
P_PPU_SPRITE511_RGB0H	0x93025 FF2	NCD			-											SP511_R0 [7:0]	
P_PPU_SPRITE511_RGB1L	0x93025 FF4				SP511_G1 [7:0]											SP511_B1 [7:0]	
P_PPU_SPRITE511_RGB1H	0x93025 FF6				-											SP511_R1 [7:0]	
P_PPU_SPRITE511_RGB2L	0x93025 FF8				SP511_G2 [7:0]											SP511_B2 [7:0]	
P_PPU_SPRITE511_RGB2H	0x93025 FFA				-											SP511_R2 [7:0]	
P_PPU_	0x93025				SP511_G3 [7:0]											SP511_B3 [7:0]	

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
SPRITE511_RGB3L	FFC																
P_PPU_SPRITE511_RGB3H	0x93025 FFE	-															SP511_R3 [7:0]

13.4.3 Sprite General Control

Before enabling the sprite function, the following register must be set at first.

P_PPU_SPRITE_CTRL 0x93020108 Sprite Control Register							
Bit	23	18	21	20	19	18	17
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM
Default	0	0	0	0	0	0	0
SPRITE_NUMBER							
	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN
0	0	0	0	0	0	0	0

- SP_EN = 0 => Disables sprite functions.
- SP_EN = 1 => Enables sprite functions.
- COORD_SEL = 0 => Coordinate mode 0
- COORD_SEL = 1 => Coordinate mode 1, this will be described in [section 12.4.4](#).
- SP_BLDMODE = 0 => Uses P_BLENDING (4-level) for all sprites' blending parameters.
- SP_BLDMODE = 1 => Uses SPN_BLDLVL (64-level) blending factor of each sprite, see [section 12.2.6](#) for details.
- SP_DIRECT = 0 => Uses relative address mode for all sprites.
- SP_DIRECT = 1 => Uses direct address mode for all sprite, see [section 12.2.10.2](#) for details.
- SP_MOSEN = 0 => Disables mosaic effect for all sprites.
- SP_MOSEN = 1 => Enables mosaic effect for all sprites, see [section 12.2.9](#) for details.
- SP_ROTEN = 0 => Disables rotate effect for all sprites.
- SP_ROTEN = 1 => Enables rotate effect for all sprites, see [section 12.2.7](#) for details.
- SP_ZOOMEN = 0 => Disables zoom effect for all sprites.
- SP_ZOOMEN = 1 => Enables zoom effect for all sprites, see [section 12.2.8](#) for details.
- SPRITE_NUMBER = 0 => Enables sprite 0 ~ 1023.
- SPRITE_NUMMER = 1 => Enables sprite 0~3.
- SPRITE_NUMMER = 2 => Enables sprite 0~7.
- ...
SPRITE_NUMMER = 255=> Enables sprite 0~1019.

SP_EFFEN = 0 => Disables sprite special effect.
 SP_EFFEN = 1 => Enables sprite special effect, see [section 12.4](#) for details.
 CDM = 0 => Disables embedded sprite color dither mode.
 CDM = 1 => Enables embedded sprite color dither mode. At this mode, the maximum sprite numbers will become 512 and sprite virtual 3D mode will enforce to be enabled.
SP_FAR = 0 => Use direct address mode with multiply by 2.
SP_FAR = 1 => Use direct address mode with multiply by 16, see [section 12.2.11.2](#) for detail.
 SP_INTP is 0 => Disable sprite bi-linear interpolation mode.
 SP_INTP is 1 => Enable sprite bi-linear interpolation mode.
 SP_LS is 0 => Disable sprite large size function.
 SP_LS is 1 => Enable sprite large size function.
 SP_GRP is 0 => Disable sprite group function.
 SP_GRP is 1 => Enable sprite group function.
 SP_FRAC is 0 => Disable sprite fraction coordinate function.
 SP_FRAC is 1 => Enable sprite fraction coordinate function.
SP_CMASK is 0 => Disable sprite color mask function.
SP_CMASK is 1 => Enable sprite color mask function.

Note: The bit 3 of P_PPU_SPRITE_CTRL is reserved for PPU internal usage, so users can't change this bit setting to "1". Otherwise, the sprite function will be abnormal.

P_PPU_ENABLE								0x930201FC								PPU Control Register									
Bit	31	30	29	28	27	26	25	24									DEFEN		TFTLB						
Function	-																DEFEN		TFTLB						
Default	0 0 0 0 0 0 0 0																0		0						
																23 22 21 20 19 18 17 16		YUV_TYPE		LB		TFT_SIZE			
																0 0 0 0 0 0 0 0		0		0		0			
																15 14 13 12 11 10 9 8		SAVE_ROM FB_SEL SPR_WIN HVCMP_DIS		FB_MONO		SPR25D FB_FORMAT			
																0 0 0 0 0 0 0 0		0		0		0			
																7 6 5 4 3 2 1 0		FB_EN - VGA_NOINTL		VGA_EN TX_BOTUP		TX_DIRECT CH0_BLK		PPU_EN	
																0 0 0 0 0 0 0 0		0		0		0			

- SPR25D is 0 => Disables sprite virtual 3D function.
 SPR25D is 1 => Enables sprite virtual 3D function, the zoom and rotate functions will be disabled under this mode.
 SPR_WIN is 0 => Disables sprite window function.
 SPR_WIN is 1 => Enables sprite window function. The blending function will be reduced to 16 levels under this mode.

To enable a sprite, the SPN_CHARNUM [15:0] must not be zero. Programmer must be careful when using direct address mode. For example, if the SPN_CHARNUM [23:0] = 0x10000, this sprite will be treated as disabled. Manually moving the character address is required in this condition.

To make sure the sprites can be shown correctly on the screen, the used sprites must be enabled sequentially. For example, if 100 sprites are used, the programmer must enable sprite0~99 in order rather than enabling sprite0~49 and sprite60~109.

13.4.4 Sprite Coordinate

The following registers are able to control the sprite location on the screen.

P_PPU_SPRITEN_X_POSITION 0x93022002+(16*N) Sprite N X/Rotate Register							
Bit	15	14	13	12	11	10	9 8
Function	ANGLE						
Default	-	-	-	-	-	-	-
-							
7 6 5 4 3 2 1 0							
SpN_X [7:0]							
-							

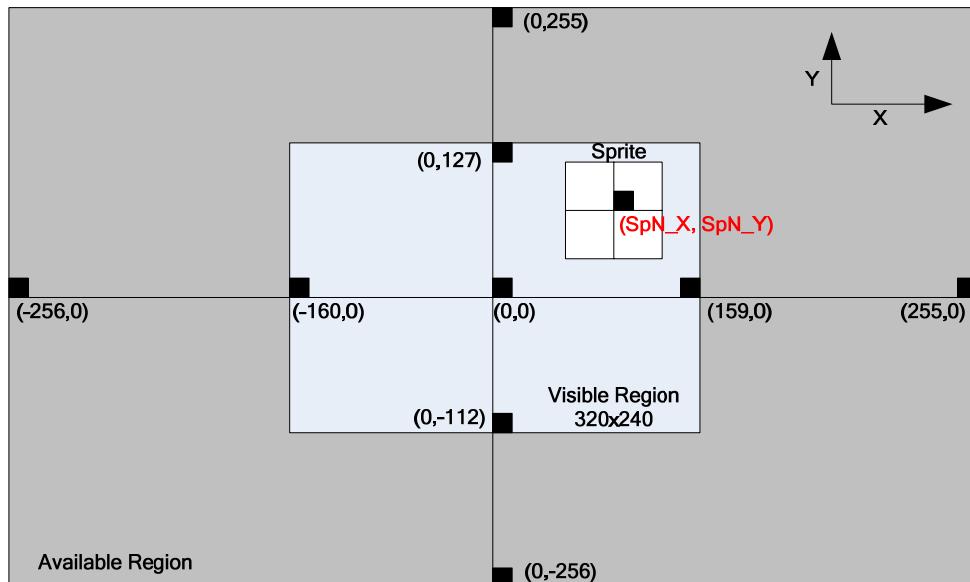
P_PPU_SPRITEN_Y_POSITION 0x93022004+(16*N) Sprite N Y/Zoom Register							
Bit	15	14	13	12	11	10	9 8
Function	ZOOM						
Default	-	-	-	-	-	-	-
-							
7 6 5 4 3 2 1 0							
SpN_Y [7:0]							
-							

Both the SpN_X and SpN_Y are 10-bit signed values; the meanings of these two parameters are depending on the coordinate system. The GPL32900A supports two sprite coordinates in QVGA mode, but only one in VGA mode. The following section describes the coordinates in QVGA mode and VGA mode.

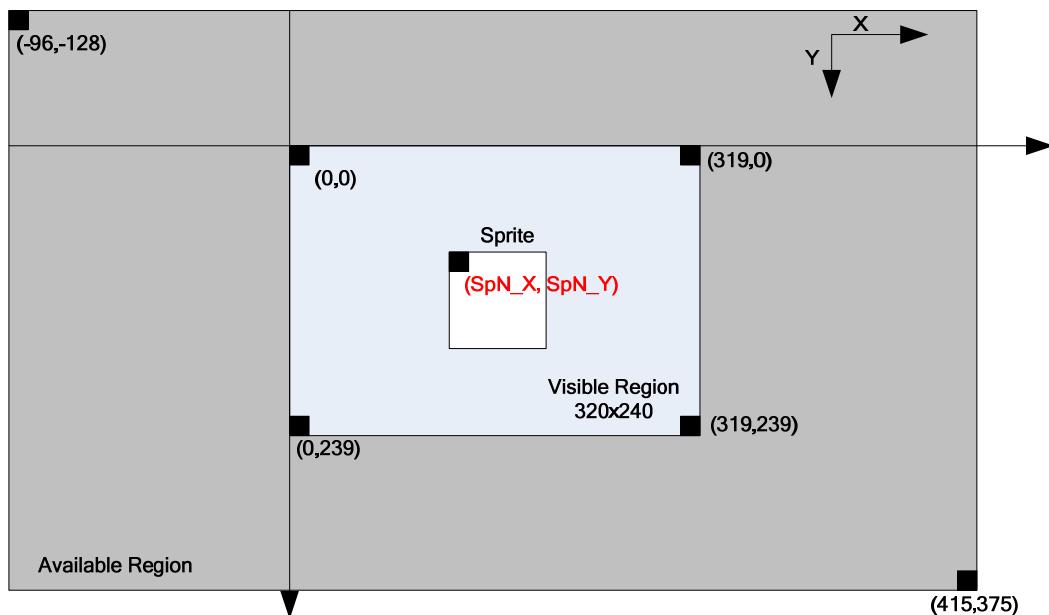
QVGA sprite coordinate mode

In QVGA mode, COORD_SEL selects between two coordinate modes - center mode and top-left mode. The following diagrams show the differences between these two modes.

Center Coordinate Mode, COORD_SEL = 0



Top-Left Coordinate Mode, COORD_SEL = 1

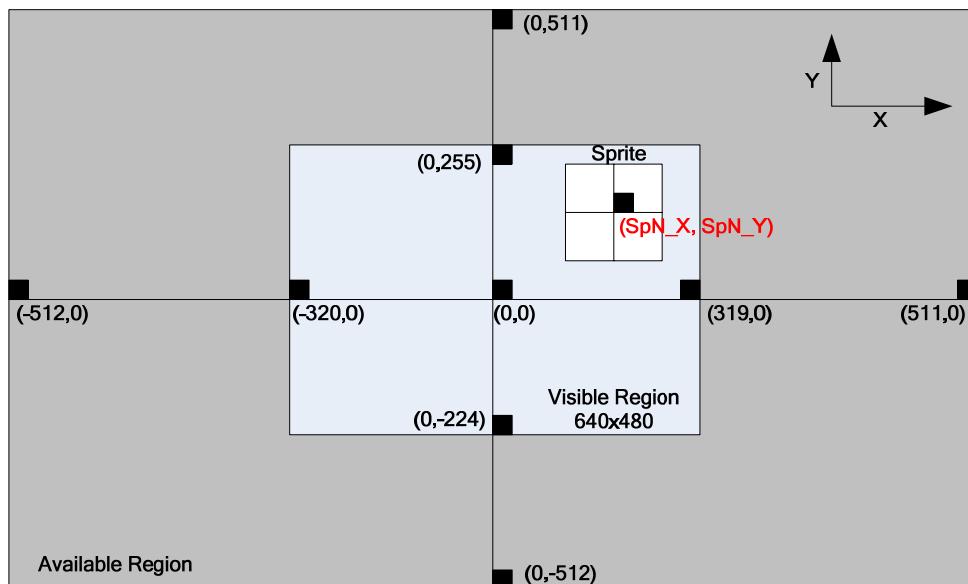


The available regions of these two modes are the same, but the bit number of SpN_X and SpN_Y is 9 bits in center mode, and 10 bits in top-left mode.

VGA sprite coordinate mode

In VGA mode, only the center coordinate mode is available. The following diagram shows the coordinate mode in VGA mode.

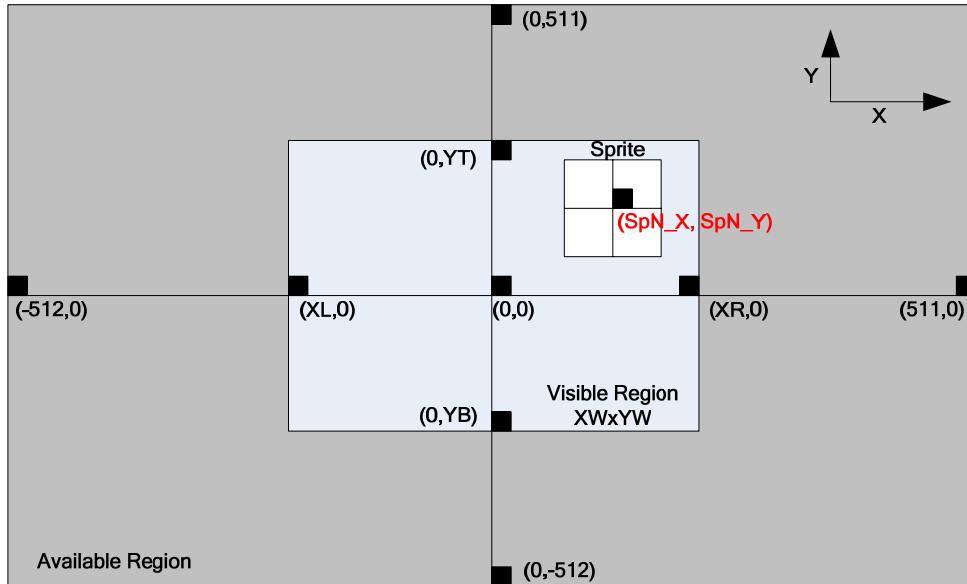
Center Coordinate Mode, COORD_SEL = 0



It is not allowed to set COORD_SEL to 1 in VGA mode.

When TFT_SIZE is not “0” and VGA_EN is “0”, the screen resolution can be different values. Under these modes, only coordinate mode 0 is supported. The following diagram shows the coordinate system under each resolution.

Center Coordinate Mode, COORD_SEL = 0



TFT_SIZE	XW	YW	XL	XR	YT	YB
0	320	240	-160	159	127	-112
1	640	480	-320	319	255	-224
2	480	234	-240	239	127	-106
3	480	272	-240	239	127	-144
4	720	480	-360	359	255	-224
5	800	480	-400	399	255	-224
6	800	600	-400	399	299	-300
7	1024	768	-512	511	383	-384

When the 1024x768 mode is selected, the sprite coordinate can't be out of the screen at the X-axis.

When free resolution mode is used, the sprite coordinate center will be dependent on the resolution setting in the FREE_HWIDTH and FREE_VWIDTH.

According to above table, you can find if you are setting the same resolution as that setting by TFT_SIZE, you will get exactly the same sprite coordinate.

FREE_HWIDTH	FREE_VWIDTH	XW	YW	XL	XR	YT	YB
X	Y(1~272)	X	Y	-1*(X/2)	X/2 - 1	127	128-Y
X	Y(273~576)	X	Y	-1*(X/2)	X/2 - 1	255	256-Y
X	Y(576~1024)	X	Y	-1*(X/2)	X/2 - 1	Y/2 - 1	-1*(Y/2)

In order to improve the coordinate accuracy on the screen, the following registers are used to control the fraction part of the coordinate.

Sprite N Extend Attribute Register							
Bit	23	22	21	20	19	18	17
Function	CMASK			INTP	LS	GROUP	
Default	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
Y3_FRAC		X3_FRAC		Y2_FRAC		X2_FRAC	

7	6	5	4	3	2	1	0
Y1_FRAC		X1_FRAC		Y0_FRAC		X0_FRAC	

X0_FRAC: Fraction part of X0 (in 2.5D mode) or X (in 2D mode)

Y0_FRAC: Fraction part of Y0 (in 2.5D mode) or Y (in 2D mode)

X1_FRAC: Fraction part of X1 (in 2.5D mode)

Y1_FRAC: Fraction part of Y1 (in 2.5D mode)

X2_FRAC: Fraction part of X2 (in 2.5D mode)

Y2_FRAC: Fraction part of Y2 (in 2.5D mode)

X3_FRAC: Fraction part of X3 (in 2.5D mode)

Y3_FRAC: Fraction part of Y3 (in 2.5D mode)

The fraction part is an extension of the original coordinate, see the following example.

X0	X0_FRAC	Real Coordinate
3	0	3.00
3	1	3.25
3	2	3.50
3	3	3.75
-3	0	-3.00
-3	1	-2.75
-3	2	-2.50
-3	3	-2.25

In order to display the correct result on the screen, the fraction function must be combined with the bilinear interpolation. If the bi-linear interpolation is not turned on, the result of coordinate with fraction will be the same as that coordinate without fraction part.

For application requires resolution higher then 1024x768, ex. 720p, the original sprite coordinate bits is not enough (only -512~511). At this programmer can choice to enable the sprite address double mode.

P_PPU_Misc
0x01F8
PPU Misc Control Register

Bit	23	22	21	20	19	18	17	16
Function			TFT_3D					SBMP_MODE
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
TV_LB		TFTVTQ	DELGO	INTP_MODE	NEW_CMP	TXT_ALPHA	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ADDR_X2	DUAL_BLD			SPR_RGB	TXT_RGB	FB_LOCK	
0	0	0	0	0	0	0	0

SP_ADDR_X2	Sprite coordinate.
0	$X = X0 + X0_FRAC/4$ $Y = Y0 + Y0_FRAC/4$
1	$X = (X0 + X0_FRAC/4)*2$ $Y = (Y0 + Y0_FRAC/4)*2$

When this function is enabled, the sprite coordinate range can reach -1024 to 1023, this make it possible to have high resolution content.

13.4.5 Sprite DMA

The sprite DMA is used to move sprite parameter from system memory to sprite memory, HVOFFSET memory, rotate memory, or palette memory. Since the memory can be updated only in the vertical blank region, programmers can update the system memory at any time and use sprite DMA to move data to these memory at vertical blank time. The following registers are used to control the sprite DMA.

P_PPU_SPRITE_DMA_SOURCE
0x930201C0
Sprite DMA Source Register

Bit	31	30	29	28	27	26	25	24
Function	SPDMA_SOURCE [31:24]							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
SPDMA_SOURCE [23:18]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SPDMA_SOURCE [15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SPDMA_SOURCE [7:0]							
0	0	0	0	0	0	0	0

P_PPU_SPRITE_DMA_TARGET **0x930201C4** **Sprite DMA Target Register**

Bit	15	14	13	12	11	10	9	8	
Function	-	SPDMA_TARGET [14:8]							
Default	0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
SPDMA_TARGET [7:0]							
0	0	0	0	0	0	0	0

P_PPU_SPRITE_DMA_NUMBER **0x930201C8** **Sprite DMA Number Register**

Bit	15	14	13	12	11	10	9	8	
Function	-	SPDMA_NUMBER [11:8]							
Default	0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
SPDMA_NUMBER [7:0]							
0	0	0	0	0	0	0	0

SPDMA_SOURCE [31:0] => DMA source address in system memory must be 4 bytes alignment.

SPDMA_TARGET [14:0] => DMA target address, range from 0x93020400 to 0x93025FFF, which map to real register address in PPU.

SPDMA_NUMBER [11:0] => DMA transfer number register.

For sprite attribute data.

Range 0x0000~0xFFFF, correspond to transfer number 1 ~ 4096 words (4 bytes).

For other data (palette/rotate data/Hvoffset)

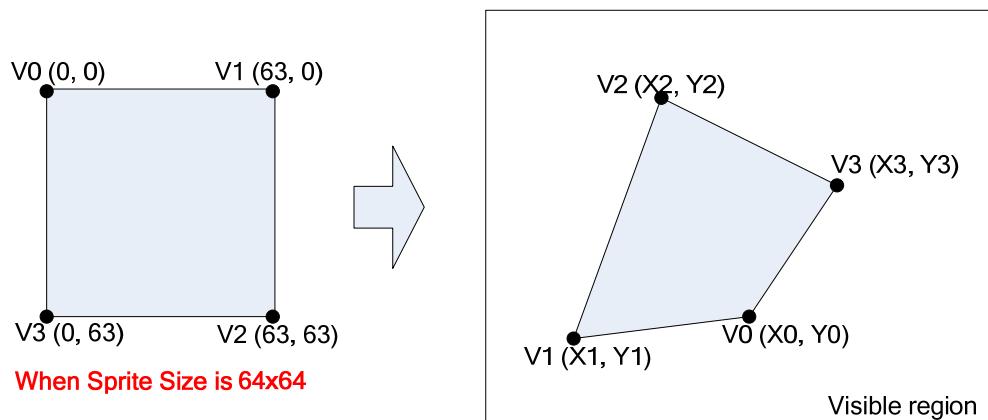
Range 0x0000~0x00FF, correspond to transfer number 1 ~ 256 half words (2 bytes)

The sprite DMA will automatically perform word to half word conversion when the SPDMA_TARGET is set as HVOFFSET/Rotate/Palette RAM. So programmers can arrange these data in system memory as half-word data structure with word aligned address. The following table shows the data format in system memory under different SPDMA_TARGET.

SPDMA_TARGET	Description	Data structure in system memory	Data structure in PPU
0x0400~0x07FF	HVOFFSET RAM	{6'h0, Hvo1[9:0], 6'h0, Hvo0[9:0]} {6'h0, Hvo3[9:0], 6'h0, Hvo2[9:0]} ...	0x0400: {6'h0, Hvo0[9:0]} 0x0404: {6'h0, Hvo1[9:0]} 0x0408: {6'h0, Hvo2[9:0]}
0x0800~0x0FFF	Rotate RAM	{3'h0, Rot1[12:0], 3'h0, Rot0[12:0]} {3'h0, Rot3[12:0], 3'h0, Rot2[12:0]} ...	0x0800: {3'h0, Rot0[12:0]} 0x0804: {3'h0, Rot1[12:0]} 0x0808: {3'h0, Rot2[12:0]}
0x1000~0x1FFF	Palette RAM	{Pal1[15:0], Pal0[15:0]} {Pal3[15:0], Pal2[15:0]} ...	0x1000: Pal0[15:0] 0x1004: Pal1[15:0] 0x1008: Pal2[15:0]
0x2000~0x5FFF	Sprite RAM	{Spr1[15:0], Spr0[15:0]} {Spr3[15:0], Spr2[15:0]} ...	0x2000: {Spr1[15:0], Spr0[15:0]} 0x2004: {Spr3[15:0], Spr2[15:0]} ...
0x6000~0x6FFF	Sprite extend RAM	{12'h0, SprExt0[19:0]} {12'h0, SprExt1[19:0]} ...	0x6000: {12'h0, SprExt0[19:0]} 0x6004: {12'h0, SprExt1[19:0]} ...

13.4.6 Sprite Virtual 3D Mode

The sprite virtual 3D mode is used to map the 2D texture at the arbitration location on the screen. This mode is useful while combining with TEXT virtual 3D mode to recognize the virtual 3D game. The following diagram shows an example of sprite virtual 3D mode.



The coordinate of V0 ~ V3 can be programmed arbitrary on the visible screen.

The following registers are used to control the virtual 3D mode.

P_PPU_ENABLE								0x930201FC								PPU Control Register								
Bit	31	30	29	28	27	26	25	24																
Function	-																TFTLB							
Default	0 0 0 0 0 0 0 0																							
23	22	21	20	19	18	17	16																	
-	YUV_TYPE								LB								TFT_SIZE							
0	0	0	0	0	0	0	0																	
15	14	13	12	11	10	9	8																	
SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO	SPR25D	FB_FORMAT																		
0	0	0	0	0	0	0	0																	
7	6	5	4	3	2	1	0																	
FB_EN	-	VGA_NOINTL	VGA_EN	TX_BOTUP	TX_DIRECT	CH0_BLK	PPU_EN																	
0	0	0	0	0	0	0	1																	

SPR25D is 0 => Disables sprite virtual 3D function.

SPR25D is 1 => Enables sprite virtual 3D function; the zoom and rotate functions will be disabled under this mode. However, if the FLIP parameter is set as HV flip (3) under sprite virtual 3D mode, the sprite can use normal zoom and rotate functions.

Each sprite needs coordinate of total four vertices. The following registers are used to control these coordinates.

P_PPU_SPRITEN_X_POSITION 0x93022002+(16*N) Sprite N X/Rotate Register							
Bit	15	14	13	12	11	10	9 8
Function	Y1 [5:0]						X [9:8]

Default	-	-	-	-	-	-	-																
<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> <tr> <td colspan="8">X [7:0]</td></tr> </table>								7	6	5	4	3	2	1	0	X [7:0]							
7	6	5	4	3	2	1	0																
X [7:0]																							

P_PPU_SPRITEN_Y_POSITION 0x93022004+(16*N) Sprite N Y/Zoom Register							
Bit	15	14	13	12	11	10	9 8
Function	Y2 [5:0]						Y [9:8]

Default	-	-	-	-	-	-	-																
<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> <tr> <td colspan="8">Y [7:0]</td></tr> </table>								7	6	5	4	3	2	1	0	Y [7:0]							
7	6	5	4	3	2	1	0																
Y [7:0]																							

P_PPU_SPRITEN_X1 0x9302200A+(16*N) Sprite N X1 Register							
Bit	15	14	13	12	11	10	9 8
Function	Y3 [5:0]						X1 [9:8]

Default	-	-	-	-	-	-	-																
<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> <tr> <td colspan="8">X1 [7:0]</td></tr> </table>								7	6	5	4	3	2	1	0	X1 [7:0]							
7	6	5	4	3	2	1	0																
X1 [7:0]																							

P_PPU_SPRITEN_X2 0x9302200C+(16*N) Sprite N X2 Register							
Bit	15	14	13	12	11	10	9 8
Function	Y3 [7:6]						X2 [9:8]

Default	-	-	-	-	-	-	-																
<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> <tr> <td colspan="8">X2 [7:0]</td></tr> </table>								7	6	5	4	3	2	1	0	X2 [7:0]							
7	6	5	4	3	2	1	0																
X2 [7:0]																							

P_PPU_SPRITEN_X3		0x93022200E+(16*N)								Sprite N X3 Register							
Bit	15	14	13	12	11	10	9	8									
Function	Y3 [9:8]				Y2 [9:6]				X3 [9:8]								
Default	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0	X3 [7:0]								
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

The order of (X0, Y0), (X1, Y1), (X2, Y2) and (X3, Y3) must be in the clockwise sequence.

In order to reduce the CPU loading while filling sprite virtual 3D coordinate, it converts the virtual 3D coordinate into the SRAM's format and executes overflow detection at the same time. The following registers are used to fill sprite virtual 3D coordinate.

P_PPU_SPRITE_X0/1/2/3		0x93020300 0x93020308								Sprite X0/1/2/3 Register							
		0x93020310 0x93020318															
Bit	15	14	13	12	11	10	9	8									
Function	SPR_X0/1/2/3 [15:8]																
Default	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0	SPR_X0/1/2/3 [7:0]								
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

P_PPU_SPRITE_Y0/1/2/3		0x93020304 0x9302030C								Sprite Y0/1/2/3 Register							
		0x93020314 0x9302031C															
Bit	15	14	13	12	11	10	9	8									
Function	SPR_Y0/1/2/3 [15:8]																
Default	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0	SPR_Y0/1/2/3 [7:0]								
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

;where the SPR_XN and SPR_YN have ranges from 32767 to -32768. Because the PPU only accepts input from 511 to -512, an automatic overflow control will be applied on SPR_XN and SPR_YN. For example, when 512 is input, output result will be 511.

The final results will be shown on the following registers.

P_PPU_SPRITE_W0
0x93020320
Sprite Word 0 Register

Bit	15	14	13	12	11	10	9	8
Function	SPR_Y1 [5:0]							
Default	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0
	SPR_X0 [7:0]							
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-

P_PPU_SPRITE_W1
0x93020324
Sprite Word 1 Register

Bit	15	14	13	12	11	10	9	8
Function	SPR_Y2 [5:0]							
Default	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0
	SPR_Y0 [7:0]							
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-

P_PPU_SPRITE_W2
0x93020328
Sprite Word 2 Register

Bit	15	14	13	12	11	10	9	8
Function	SPR_Y3 [5:0]							
Default	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0
	SPR_X1 [7:0]							
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-

P_PPU_SPRITE_W3
0x9302032C
Sprite Word 3 Register

Bit	15	14	13	12	11	10	9	8
Function	SPR_Y3 [7:6]							
Default	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0
	SPR_X2 [7:0]							
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-

P_PPU_SPRITE_W4								0x93020330								Sprite Word 4 Register								
Bit	15	14	13	12	11	10	9	8																
Function	SPR_Y3 [9:8]				SPR_Y2 [9:6]				SPR_X3 [9:8]															
Default	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0									SPR_X3 [7:0]							
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

13.4.7 Sprite Color Dither Mode

The sprite color dither mode is used to paint triangle or rectangle type sprite by specifying the coordinate and RGB color of each vertex. Under this mode, the sprite virtual 3D mode is forced to enable and the maximum sprite numbers are reduced to 512 based on various structures of sprite RAM. The following register is used to enable the color dither mode of embedded sprite.

P_PPU_SPRITE_CTRL								0x93020108								Sprite Control Register									
Bit	23	22	17	20	19	18	17	16																	
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN																	
Default	0	0	0	0	0	0	0	0																	
	15	14	13	12	11	10	9	8	SPRITE_NUMBER																
	0	0	0	0	0	0	0	0																	
	7	6	5	4	3	2	1	0	SP_ZOOMEN								SP_ROTEN								
	0	0	0	0	0	0	0	0	SP_MOSEN								SP_DIRECT								
	SP_BLDMODE	COORD_SEL	SP_EN																						

CDM = 0 => Disables embedded sprite color dither mode.

CDM = 1 => Enables embedded sprite color dither mode. At this mode, the maximum sprite numbers will become 512 and sprite virtual 3D mode will be forced to enable.

The following table shows the sprite SRAM structure under sprite color dither mode.

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_SPRITE0_CHARNUM	0x93022000	SP0_CHARNUM [15:0]															
P_PPU_SPRITE0_X_POSITION	0x93022002	SP0_ROTATE [5:0]/SP0_Y1 [5:0]						SP0_X_POSITION [9:0]									
P_PPU_SPRITE0_Y_POSITION	0x93022004	SP0_ZOOM[5:0]/SP0_Y2[5:0]						SP0_Y_POSITION [9:0]									
P_PPU_SPRITE0_ATTRIBUTE0	0x93022006	PB	BLD	DEPTH	PALETTE				VS	HS		FLIP	COLOR				

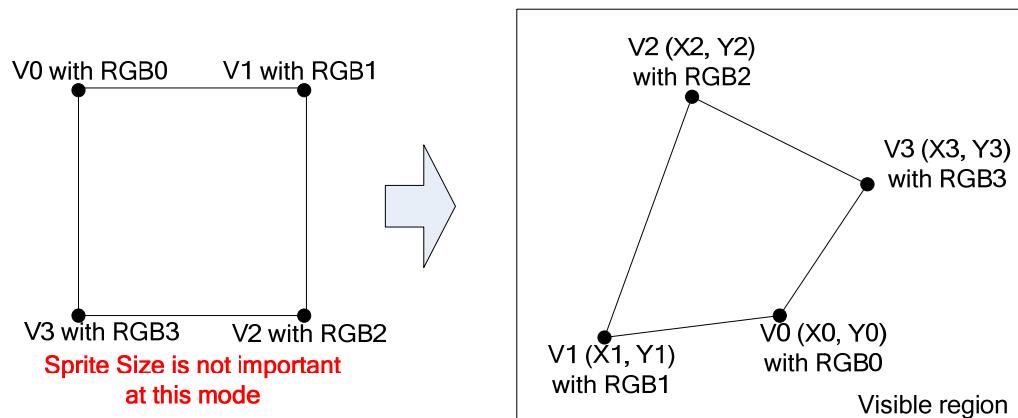
NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0					
P_PPU_SPRITE0_ATTRIBUTE1	0x93022008	MOSAIC	SP0_BLDLVL								SP0_CHARNUM [23:16]											
P_PPU_SPRITE0_X1	0x9302200A	SP0_Y3 [5:0]								SP0_X1 [9:0]												
P_PPU_SPRITE0_X2	0x9302200C	SP0_Y3 [7:6]	SP0_Y1 [9:6]								SP0_X2 [9:0]											
P_PPU_SPRITE0_X3	0x9302200E	SP0_Y3 [9:8]	SP0_Y2 [9:6]								SP0_X3 [9:0]											
P_PPU_SPRITE0_RGB0L	0x93022010	SP0_G0 [7:0]								SP0_B0 [7:0]												
P_PPU_SPRITE0_RGB0H	0x93022012	NCD	-								SP0_R0 [7:0]											
P_PPU_SPRITE0_RGB1L	0x93022014	SP0_G1 [7:0]								SP0_B1 [7:0]												
P_PPU_SPRITE0_RGB1H	0x93022016	-								SP0_R1 [7:0]												
P_PPU_SPRITE0_RGB2L	0x93022018	SP0_G2 [7:0]								SP0_B2 [7:0]												
P_PPU_SPRITE0_RGB2H	0x9302201A	-								SP0_R2 [7:0]												
P_PPU_SPRITE0_RGB3L	0x9302201C	SP0_G3 [7:0]								SP0_B3 [7:0]												
P_PPU_SPRITE0_RGB3H	0x9302201E	-								SP0_R3 [7:0]												
...												
P_PPU_SPRITE511_CHARNUM	0x93025FE0	SP511_CHARNUM [15:0]																				
P_PPU_SPRITE511_X_POSITION	0x93025FE2	SP511_ROTATE [5:0]/SP511_Y1 [5:0]								SP511_X_POSITION [9:0]												
P_PPU_SPRITE511_Y_POSITION	0x93025FE4	SP511_ZOOM [5:0]/SP511_Y2 [5:0]								SP511_Y_POSITION [9:0]												
P_PPU_SPRITE511_ATTRIBUTE0	0x93025FE6	PB	BLD	DEPTH	PALETTE				VS	HS	FLIP	COLOR										
P_PPU_SPRITE511_ATTRIBUTE1	0x93025FE8	MOSAIC	SP511_BLDLVL								SP511_CHARNUM [23:16]											
P_PPU_SPRITE511_X1	0x93025FEA	SP511_Y3 [5:0]								SP511_X1 [9:0]												
P_PPU_SPRITE511_X2	0x93025FEC	SP511_Y3 [7:6]	SP511_Y1 [9:6]								SP511_X2 [9:0]											

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0					
P_PPU_SPRITE511_X3	0x9302 5FEE	SP511_Y3 [9:8]	SP511_Y2 [9:6]										SP511_X3 [9:0]									
P_PPU_SPRITE511_RGB0L	0x9302 5FF0	SP511_G0 [7:0]										SP511_B0 [7:0]										
P_PPU_SPRITE511_RGB0H	0x9302 5FF2	NCD	-										SP511_R0 [7:0]									
P_PPU_SPRITE511_RGB1L	0x9302 5FF4	SP511_G1 [7:0]										SP511_B1 [7:0]										
P_PPU_SPRITE511_RGB1H	0x9302 5FF6	-										SP511_R1 [7:0]										
P_PPU_SPRITE511_RGB2L	0x9302 5FF8	SP511_G2 [7:0]										SP511_B2 [7:0]										
P_PPU_SPRITE511_RGB2H	0x9302 5FFA	-										SP511_R2 [7:0]										
P_PPU_SPRITE511_RGB3L	0x9302 5FFC	SP511_G3 [7:0]										SP511_B3 [7:0]										
P_PPU_SPRITE511_RGB3H	0x9302 5FFE	-										SP511_R3 [7:0]										

- SPN_NCD = 0 => This sprite is a color dither sprite.
 SPN_NCD = 1 => This sprite is a normal texture mapping sprite.
 {SPN_R0, SPN_G0, SPN_B0} => Color of vertex 0 of color dither sprite.
 {SPN_R1, SPN_G1, SPN_B1} => Color of vertex 1 of color dither sprite.
 {SPN_R2, SPN_G2, SPN_B2} => Color of vertex 2 of color dither sprite.
 {SPN_R3, SPN_G3, SPN_B3} => Color of vertex 3 of color dither sprite.

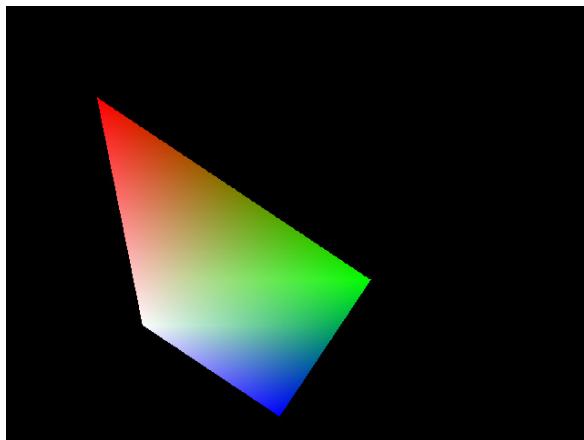
Under the sprite color dither mode, the HS/ VS/ FLIP/ PALETTE/ COLOR/ PB/ MOSAIC/ CHARNUM/ MOSAIC parameters are not used.

The following diagrams show the concept of color dither mode. User can specify the screen location and RGB color of each vertex, and the color at internal of the sprite will automatically be computed.



The coordinate of V0 ~ V3 can be programmed arbitrary on the visible screen.

The following figure shows an example of the color dither sprite.



Under the sprite color dither mode, it does not consume any bandwidth of system. The maximum sprite the PPU can plot based on the clock speed of PPU which is approximately 3000 pixels per half-line in 96MHz mode.

13.4.8 Extended Sprite General Control

The GP12 integrates a powerful sprite function, extended sprite mode, which allows programmer to display 4086 sprites at the same time using system memory as the sprite memory. The extended sprites have the same behavior as the normal sprites; the sprite memory format is also the same as normal sprite memory, but the extended sprites' performance is worse than normal sprites; therefore, we recommend using extended sprites only in frame buffer mode, and using extended sprites unless all of internal 1024 sprites have been used. Before enabling the extended sprite functions, the following register must be set at first.

P_PPU_EXTENDSPRITE_CONTROL 0x93020334 Extended Sprite Control Register

Bit	15	14	13	12	11	10	9	8
Function	SPRITE_NUMBER [8:0]							
Default	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	
-		ES_FRAC	ES_GRP	ES_LS	ES_INTP	ES_CDM	ES_EN	
0	0	0	0	0	0	0	0	0

ES_EN = 0 => Disables extended sprite function.

ES_EN = 1 => Enables extended sprite function, ES_EN should also be turned on at the same time.

ES_CDM = 0 => Disables color dither mode of extend sprite, each sprite need 16 bytes.

ES_CDM = 1 => Enables color dither mode of extend sprite, each sprite need 32 bytes.

ES_INTP is 0 => Disable extended sprite bi-linear interpolation mode.

ES_INTP is 1 => Enable extended sprite bi-linear interpolation mode.

ES_LS is 0 => Disable extended sprite large size function.

ES_LS is 1 => Enable extended sprite large size function.

ES_GRP is 0 => Disable extended sprite group function.

ES_GRP is 1 => Enable extended sprite group function.

ES_FRAC is 0 => Disable extended sprite fraction coordinate function.

ES_FRAC is 1 => Enable extended sprite fraction coordinate function

SPRITE_NUMBER = 0 => Enables extend sprite 0 ~ 4095.

SPRITE_NUMNER = 1 => Enables sprite 0~15.

SPRITE_NUMNER = 2 => Enables sprite 0~31.

...

SPRITE_NUMNER = 255 => Enables sprite 0~4079.

The following registers are used to control the start address of extended sprite.

P_PPU_EXTENDSPRITE_ADDR 0x93020338 Extended Sprite Start Address

Bit	31	30	29	28	27	26	25	24
Function	ESP_ADDR [31:24]							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
-								
0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	
Function	ESP_ADDR [15:8]							
Default	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	
Function	ESP_ADDR [7:0]							
Default	0	0	0	0	0	0	0	0

The following table shows the extended sprite SRAM structure under sprite mode.

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0																	
P_PPU_SPRITE0_CHARNUM	0x.....0	SP0_CHARNUM [15:0]																																
P_PPU_SPRITE0_X_POSITION	0x.....2	SP0_ROTATE [5:0]/SP0_Y1 [5:0]				SP0_X_POSITION [9:0]																												
P_PPU_SPRITE0_Y_POSITION	0x.....4	SP0_ZOOM[5:0]/SP0_Y2[5:0]				SP0_Y_POSITION [9:0]																												
P_PPU_SPRITE0_ATTRIBUTE0	0x.....6	PB	BLD	DEPTH	PALETTE				VS	HS	FLIP		COLOR																					
P_PPU_SPRITE0_ATTRIBUTE1	0x.....8	MOSAIC	SP0_BLDLVL				SP0_CHARNUM [23:16]																											
P_PPU_SPRITE0_X1	0x.....A	SP0_Y3 [5:0]				SP0_X1 [9:0]																												
P_PPU_SPRITE0_X2	0x.....C	SP0_Y3 [7:6]	SP0_Y1 [9:6]			SP0_X2 [9:0]																												
P_PPU_SPRITE0_X3	0x.....E	SP0_Y3 [9:8]	SP0_Y2 [9:6]			SP0_X3 [9:0]																												
P_PPU_SPRITE0_EXATTRIBUTEL	0x.....10	-												INT P _p	LS _p	GROUP_p																		
P_PPU_SPRITE0_EXATTRIBUTEH	0x.....12	FRACTION_p[15:8]						FRACTION_p[7:0]																										
Reserved.	0x.....14																																	
Reserved.	0x.....16																																	
Reserved.	0x.....18																																	
Reserved.	0x.....1A																																	
Reserved.	0x.....1C																																	
Reserved.	0x.....1E																																	
P_PPU_SPRITE1_CHARNUM	0x.....20	SP0_CHARNUM [15:0]																																
P_PPU_SPRITE1_X_POSITION	0x.....22	SP0_ROTATE [5:0]/SP0_Y1 [5:0]				SP0_X_POSITION [9:0]																												
P_PPU_SPRITE1_Y_POSITION	0x.....24	SP0_ZOOM[5:0]/SP0_Y2[5:0]				SP0_Y_POSITION [9:0]																												
P_PPU_SPRITE1_ATTRIBUTE0	0x.....26	PB	BLD	DEPTH	PALETTE				VS	HS	FLIP		COLOR																					
P_PPU_SPRITE1_ATTRIBUTE1	0x.....28	MOSAIC	SP0_BLDLVL				SP0_CHARNUM [23:16]																											
P_PPU_SPRITE1_X1	0x.....2A	SP0_Y3 [5:0]				SP0_X1 [9:0]																												

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0				
P_PPU_SPRITE1_X2	0x.....2C	SP0_Y3 [7:6]	SP0_Y1 [9:6]				SP0_Y3 [7:6]														
P_PPU_SPRITE1_X3	0x.....2E	SP0_Y3 [9:8]	SP0_Y2 [9:6]				SP0_Y3 [9:8]														
P_PPU_SPRITE1_EXATTRIBUTEL	0x.....30	-										INTP _p	LS_p	GROUP_p							
P_PPU_SPRITE1_EXATTRIBUTEH	0x.....32	FRACTION_p[15:8]																			
Reserved.	0x.....34	-																			
Reserved.	0x.....36	-																			
Reserved.	0x.....38	-																			
Reserved.	0x.....3A	-																			
Reserved.	0x.....3C	-																			
Reserved.	0x.....3E	-																			
...																			

The following table shows the extended sprite SRAM structure under sprite color dither mode.

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0							
P_PPU_SPRITE0_CHARNUM	0x.....0	SP0_CHARNUM [15:0]																						
P_PPU_SPRITE0_X_POSITION	0x.....2	SP0_ROTATE [5:0]/SP0_Y1 [5:0]						SP0_X_POSITION [9:0]																
P_PPU_SPRITE0_Y_POSITION	0x.....4	SP0_ZOOM[5:0]/SP0_Y2[5:0]						SP0_Y_POSITION [9:0]																
P_PPU_SPRITE0_ATTRIBUTE0	0x.....6	PB	BLD	DEPTH	PALETTE				VS	HS		FLIP	COLOR											
P_PPU_SPRITE0_ATTRIBUTE1	0x.....8	MOSAIC		SP0_BLDLVL				SP0_CHARNUM [23:16]																
P_PPU_SPRITE0_X1	0x.....A	SP0_Y3 [5:0]						SP0_X1 [9:0]																
P_PPU_SPRITE0_X2	0x.....C	SP0_Y3 [7:6]	SP0_Y1 [9:6]				SP0_X2 [9:0]																	
P_PPU_SPRITE0_X3	0x.....E	SP0_Y3 [9:8]	SP0_Y2 [9:6]				SP0_X3 [9:0]																	
P_PPU_SPRITE0_RGB0L	0x.....10	SP0_G0 [7:0]						SP0_B0 [7:0]																
P_PPU_SPRITE0_RGB0H	0x.....12	NCD	LS_p	GROUP_p	INTP _p	-		SP0_R0 [7:0]																



NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0																
P_PPU_SPRITE0_RGB1L	0x.....14	SP0_G1 [7:0]								SP0_B1 [7:0]																							
P_PPU_SPRITE0_RGB1H	0x.....16	FRACTION_p[15:8]								SP0_R1 [7:0]																							
P_PPU_SPRITE0_RGB2L	0x.....18	SP0_G2 [7:0]								SP0_B2 [7:0]																							
P_PPU_SPRITE0_RGB2H	0x.....1A	FRACTION_p[7:0]								SP0_R2 [7:0]																							
P_PPU_SPRITE0_RGB3L	0x.....1C	SP0_G3 [7:0]								SP0_B3 [7:0]																							
P_PPU_SPRITE0_RGB3H	0x.....1E	-								SP0_R3 [7:0]																							
P_PPU_SPRITE1_CHARNUM	0x.....20	SP0_CHARNUM [15:0]																															
P_PPU_SPRITE1_X_POSITION	0x.....22	SP0_ROTATE [5:0]/SP0_Y1 [5:0]					SP0_X_POSITION [9:0]																										
P_PPU_SPRITE1_Y_POSITION	0x.....24	SP0_ZOOM[5:0]/SP0_Y2[5:0]					SP0_Y_POSITION [9:0]																										
P_PPU_SPRITE1_ATTRIBUTE0	0x.....26	PB	BLD	DEPTH	PALETTE				VS	HS	FLIP	COLOR																					
P_PPU_SPRITE1_ATTRIBUTE1	0x.....28	MOSAIC	SP0_BLDLVL					SP0_CHARNUM [23:16]																									
P_PPU_SPRITE1_X1	0x.....2A	SP0_Y3 [5:0]					SP0_X1 [9:0]																										
P_PPU_SPRITE1_X2	0x.....2C	SP0_Y3 [7:6]		SP0_Y1 [9:6]			SP0_X2 [9:0]																										
P_PPU_SPRITE1_X3	0x.....2E	SP0_Y3 [9:8]		SP0_Y2 [9:6]			SP0_X3 [9:0]																										
P_PPU_SPRITE1_RGB0L	0x.....30	SP0_G0 [7:0]								SP0_B0 [7:0]																							
P_PPU_SPRITE1_RGB0H	0x.....32	NC D	LS_p	GROUP_p	INTP_p	-		SP0_R0 [7:0]																									
P_PPU_SPRITE1_RGB1L	0x.....34	SP0_G1 [7:0]								SP0_B1 [7:0]																							
P_PPU_SPRITE1_RGB1H	0x.....36	FRACTION_p[15:8]								SP0_R1 [7:0]																							
P_PPU_SPRITE1_RGB2L	0x.....38	SP0_G2 [7:0]								SP0_B2 [7:0]																							
P_PPU_SPRITE1_RGB2H	0x.....3A	FRACTION_p[7:0]								SP0_R2 [7:0]																							

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	
P_PPU_SPRITE1_RGB3L	0x.....3C																SP0_G3 [7:0]	SP0_B3 [7:0]
P_PPU_SPRITE1_RGB3H	0x.....3E																SP0_R3 [7:0]	
...	

When a section of memory is used as extended sprite region, the value in this region can be changed only in V-blank time, just like the normal sprite.

The extend sprite will consume the system bandwidth based on the sprite number turned on. The following equation shows the necessary bandwidth a single frame may need.

$$\text{Necessary Bandwidth} = \text{Sprite Number} * \text{lines per frame} * 2 * \text{size per sprite}.$$

For example, when 256 extend sprite is used and normal sprite mode is selected under VGA mode, the required bandwidth is $256 * 480 * 2 * 16 = 3.932\text{Mbyte/frame}$; thus, attentions must be exercised when turning on extend sprite mode.

13.4.9 Sprite Group Function

GP12 support a sprite group function, which can be used to eliminate the over-lap effect when using the sprite 2.5D function to connect multiple sprites with blending. To use the sprite group function, the following register must be set.

P_Sp_Control 0x93020108 Sprite Control Register							
Bit	23	22	21	20	19	18	17
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM
Default	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Sprite_Number							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN
0	0	0	0	0	0	0	0

SP_GRP is 0 => Disable sprite group function.

SP_GRP is 1 => Enable sprite group function.

P_SpN_EXT 0x93026000+(4*N) Sprite N Extend Attribute Register							
Bit	23	22	21	20	19	18	17
Function	CMASK				INTP	LS	GROUP
Default	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
Y3_FRAC	X3_FRAC	Y2_FRAC	X2_FRAC	-	-	-	-
-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
Y1_FRAC	X1_FRAC	Y0_FRAC	X0_FRAC	-	-	-	-
-	-	-	-	-	-	-	-

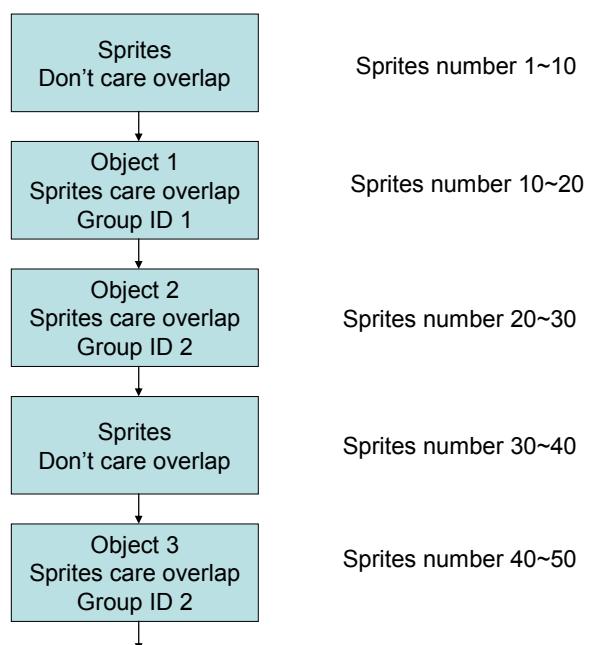
SpN_GROUP is 0 => No group function, treated as normal sprite.

SpN_GROUP is 1 => Sprite group 1, this kind of sprite will not overlap the sprite with the same group ID 1.

SpN_GROUP is 2 => Sprite group 2, this kind of sprite will not overlap the sprite with the same group ID 2.

SpN_GROUP is 3 => Sprite group 3, this kind of sprite will not overlap the sprite with the same group ID 3.

In order to use the sprite group function, sprites which have the overlap concern must have its own sprite ID which is different from its neighbor sprites. See the following example.



The group ID can be reused as long as the neighbor object has different group ID.

13.5 PPU Interrupt Control

GP12's PPU has many interrupt sources. The following registers are the interrupt control registers.

P_PPU_IRQ_EN 0x93020188								PPU IRQ Enable Register								
Bit	15	14	13	12	11	10	9	8	PPU_IRQ_EN [15:8]							
Function	0	0	0	0	0	0	0	0	PPU_IRQ_EN [7:0]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
								PPU_IRQ_EN [7:0]								
								0 0 0 0 0 0 0 0								

P_PPU_IRQ_STATUS 0x9302018C								PPU IRQ Status Register								
Bit	15	14	13	12	11	10	9	8	PPU_IRQ_STS [15:8]							
Function	0	0	0	0	0	0	0	0	PPU_IRQ_STS [7:0]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
								PPU_IRQ_STS [7:0]								
								0 0 0 0 0 0 0 0								

Bit	Function	Type	Description								Condition
17	IRQ_EN17	R/W	Illegal write enable/disable control register.								0: Disable IILEGALIRQ 1: Enable IILEGALIRQ
16	-	-	-								-
15	IRQ_EN15	R/W	Sensor receiving FIFO under-run IRQ enable/disable control register.								0: Disable SENOFIRQ 1: Enable SENOFIRQ
14	IRQ_EN14	R/W	PPU horizontal blank IRQ enable/disable control								0: Disables HBIRQ 1: Enables HBIRQ
13	IRQ_EN13	R/W	TFT-LCD frame mode vertical blank IRQ enable/disable control								0: Disables TFTVB IRQ 1: Enables TFTVB IRQ
12	IRQ_EN12	R/W	TFT-LCD frame buffer FIFO under-run IRQ enable/disable								0: Disables TFTFBUR IRQ 1: Enables TFTFBUR IRQ
11	IRQ_EN11	R/W	TV frame mode vertical blank IRQ enable/disable control								0: Disables TVVB IRQ 1: Enables TVVB IRQ
10	IRQ_EN10	R/W	TV frame buffer FIFO under-run IRQ enable/disable control								0: Disables TVFBUR IRQ 1: Enables TVFBUR IRQ
9	IRQ_EN9	R/W	Sensor motion detect FIFO under-run IRQ enable/disable control								0: Disables MDUR IRQ 1: Enables MDUR IRQ

Bit	Function	Type	Description	Condition
8	IRQ_EN8	R/W	Sensor position hit IRQ enable/disable control	0: Disables MDHIT IRQ 1: Enables MDHIT IRQ
7	IRQ_EN7	R/W	Sensor motion detect frame end IRQ enable/disable control	0: Disables MDFE IRQ 1: Enables MDFE IRQ
6	IRQ_EN6	R/W	Sensor frame end IRQ enable/disable control	0: Disables SENFE IRQ 1: Enables SENFE IRQ
5	IRQ_EN5	R/W	Sprite engine under-run IRQ enable/disable control	0: Disables SPRUR IRQ 1: Enables SPRUR IRQ
4	IRQ_EN4	R/W	TEXT engine under-run IRQ enable/disable control	0: Disables TXTUR IRQ 1: Enables TXTUR IRQ
3	IRQ_EN3	R/W	Palette RAM write error IRQ enable/disable control	0: Disables PALERR IRQ 1: Enables PALERR IRQ
2	IRQ_EN2	R/W	DMA transfer complete IRQ enable/disable control	0: Disables DMA IRQ 1: Enables DMA IRQ
1	IRQ_EN1	R/W	Video position IRQ enable/disable control	0: Disables VDO IRQ 1: Enables VDO IRQ
0	IRQ_EN0	R/W	Vertical blanking IRQ enable/disable control	0: Disables VBLANK IRQ 1: Enables VBLANK IRQ

Bit	Function	Type	Description	Condition
17	IRQ_STS17	R/W	Illegal write IRQ status	Read 0: IIILEGALIRQ not happened. Read 1: IIILEGALIRQ happened. Write 0: No effect.. Write 1: Clear this bit.
16	-	-	-	-
15	IRQ_STS15	R/W	Sensor receiving FIFO under-run IRQ status	Read 0: SENOFIRQ not happened. Read 1: SENOFIRQ happened. Write 0: No effect.. Write 1: Clear this bit.
14	IRQ_STS14	R/W	PPU horizontal blank IRQ status	Read 0: HBIRQ not occurred. Read 1: HBIRQ occurred. Write 0: No effect. Write 1: Clear this bit.
13	IRQ_STS13	R/W	TFT-LCD frame mode vertical blank IRQ status	Read 0: TFTVB IRQ not occurred. Read 1: TFTVB IRQ occurred. Write 0: No effect. Write 1: Clear this bit.

Bit	Function	Type	Description	Condition
12	IRQ_STS12	R/W	TFT frame buffer FIFO under-run IRQ status	Read 0: TFTFBUR IRQ not occurred. Read 1: TFTFBUR IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
11	IRQ_STS11	R/W	TV frame mode vertical blank IRQ status	Read 0: TVVB IRQ not occurred. Read 1: TVVB IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
10	IRQ_STS10	R/W	TV frame buffer FIFO under-run IRQ status	Read 0: TVFBUR IRQ not occurred. Read 1: TVFBUR IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
9	IRQ_STS9	R/W	Sensor motion detect FIFO under-run IRQ status	Read 0: MDUR IRQ not occurred. Read 1: MDUR IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
8	IRQ_STS8	R/W	Sensor position hit IRQ status. This bit will be set to one if the sensor engine's scan point reach the value set in P_MD_HPOS and P_MD_VPOS.	Read 0: SENHIT IRQ not occurred. Read 1: SENHIT IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
7	IRQ_STS7	R/W	Sensor motion detect frame end IRQ status. This bit will be set to 1 if the motion detect engine is completely capture a frame.	Read 0: MDFE IRQ not occurred. Read 1: MDFE IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
6	IRQ_STS6	R/W	Sensor frame end IRQ status. This bit will be set to 1 if the sensor is completely capture a frame.	Read 0: SENFE IRQ not occurred. Read 1: SENFE IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
5	IRQ_STS5	R/W	Sprite engine under-run IRQ status. This bit will be set to 1 if the sprite engine is unable to complete the calculation in one line.	Read 0: SPRUR IRQ not occurred. Read 1: SPRUR IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
4	IRQ_STS4	R/W	TEXT engine under-run IRQ status. This bit will be set to 1 if the TEXT engine is unable to complete the calculation in one line.	Read 0: TXTUR IRQ not occurred. Read 1: TXTUR IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
3	IRQ_STS3	R/W	Palette RAM write error IRQ status. This bit will be set if CPU and PPU access the palette RAM at the same time.	Read 0: PALERR IRQ not occurred. Read 1: PALERR IRQ occurred. Write 0: No effect. Write 1: Clear this bit.

Bit	Function	Type	Description	Condition
2	IRQ_STS2	R/W	DMA transfer complete IRQ status. This bit will be set if the DMA transfer is complete.	Read 0: DMA IRQ not occurred. Read 1: DMA IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
1	IRQ_STS1	R/W	Video position IRQ status. This bit will be set to 1 if the luster point is hit the value defined in P_PPU_IRQTMV and P_PPU_IRQTMH.	Read 0: VDO IRQ not occurred. Read 1: VDO IRQ occurred. Write 0: No effect. Write 1: Clear this bit.
0	IRQ_STS0	R/W	Vertical blanking IRQ status. This bit will be set to 1 if enter the VBLANK region. This bit will be clear automatically after leave the VBLANK region.	Read 0: VBLANK IRQ not occurred. Read 1: VBLANK IRQ occurred. Write 0: No effect. Write 1: Clear this bit.

In order to simplify the control of the sensor's IRQ, GP12 adds a new register to control the sensor IRQ directly. To enable this function, the following register must be set first.

P_CSI1_TG_CTRL1 0x93020240

Timing Generator Control 1

Bit	23	22	21	20	19	18	17	16
Function				OFIFOSIZE				OWNIRQ
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
INTL	FIELDINV	YUVTYPE	VRST	VADD	HRST	FGET	CCIR656
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
BSEN	YUVOUT	YUVIN	CLKIINV	RGB565	HREF	CAP	CSIEN
0	0	0	0	0	0	0	0

OWNIRQ is 0 => Original IRQ control method.

OWNIRQ is 1 => Sensor has its own IRQ. At the same time, the sensor IRQ in the PPU's register will become useless.

When OWNIRQ is 1, the following register is used to represent the sensor IRQ.

P_CSI1_TG_IRQ_EN 0x9302023C

Sensor IRQ Enable Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TG_IRQ_EN[5:0]							
0	0	0	0	0	0	0	0

P_CSI1_TG_IRQ_STS 0x93020238
Sensor IRQ Status Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TG_IRQ_STS[5:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:5]	-	-		-
5	IRQ_EN5	R/W	Sensor output FIFO mode IRQ enable/disable control register.	0: Disable SENOFIFOIRQ 1: Enable SENOFIFOIRQ
4	IRQ_EN4	R/W	Sensor receives FIFO over flow IRQ enable/disable control register.	0: Disable SENOFIRQ 1: Enable SENOFIRQ
3	IRQ_EN3	R/W	Motion detects FIFO under-run IRQ enable/disable control register.	0: Disable MDUR IRQ 1: Enable MDUR IRQ
2	IRQ_EN2	R/W	Sensor position hit IRQ enable/disable control register.	0: Disable MDHIT IRQ 1: Enable MDHIT IRQ
1	IRQ_EN1	R/W	Motion detect frame end IRQ enable/disable control register.	0: Disable MDFE IRQ 1: Enable MDFE IRQ
0	IRQ_EN0	R/W	Sensor frame end IRQ enable/disable control register.	0: Disable SENFE IRQ 1: Enable SENFE IRQ

Bit	Function	Type	Description	Condition
[31:5]	-	-		-
4	IRQ_STS5	R/W	Sensor output FIFO mode IRQ status register.	Read 0: SENOFIFOIRQ not happened. Read 1: SENOFIFOIRQ happened. Write 0: No effect.. Write 1: Clear this bit.
4	IRQ_STS4	R/W	Sensor receives FIFO over flow IRQ status register.	Read 0: SENOFIRQ not happened. Read 1: SENOFIRQ happened. Write 0: No effect.. Write 1: Clear this bit.
3	IRQ_STS3	R/W	Motion detects FIFO under-run IRQ status register.	Read 0: MDUR IRQ not happened. Read 1: MDUR IRQ happened. Write 0: No effect.. Write 1: Clear this bit.

Bit	Function	Type	Description	Condition
2	IRQ_STS2	R/W	Sensor position hit IRQ status register. This bit will be set to one if the sensor engine's scan point reach the value set in P_MD_HPOS and P_MD_VPOS.	Read 0: SENHIT IRQ not happened. Read 1: SENHIT IRQ happened. Write 0: No effect.. Write 1: Clear this bit.
1	IRQ_STS1	R/W	Motion detects frame end IRQ status register. This bit will be set to 1 if the motion detect engine is completely capture a frame.	Read 0: MDFE IRQ not happened. Read 1: MDFE IRQ happened. Write 0: No effect.. Write 1: Clear this bit.
0	IRQ_STS0	R/W	Sensor frame end IRQ status register. This bit will be set to 1 if the sensor is completely capture a frame.	Read 0: SENFE IRQ not happened. Read 1: SENFE IRQ happened. Write 0: No effect.. Write 1: Clear this bit.

For sensor 2 interface, it has its own IRQ enable and flag, which is controlled by the following register.

P_CSI1_TG2_IRQ_EN 0x930202B8								Sensor2 IRQ Enable Register								
Bit	15	14	13	12	11	10	9	8	TG2_IRQ_EN[5:0]							
Function																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
									TG2_IRQ_EN[5:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P_CSI1_TG2_IRQ_STS 0x930202BC	Sensor2 IRQ Status Register															
Bit	15	14	13	12	11	10	9	8	TG2_IRQ_STS[5:0]							
Function																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
									TG2_IRQ_STS[5:0]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:6]	-	-	Reserved	-
5	IRQ_EN5	R/W	Sensor2 output FIFO mode IRQ enable/disable control register.	0: Disable SENOFIFOIRQ 1: Enable SENOFIFOIRQ
4	IRQ_EN4	R/W	Sensor2 receive FIFO over flow IRQ enable/disable control2 register.	0: Disable SENOFIRQ 1: Enable SENOFIRQ
3			Reserved	
2			Reserved	
1			Reserved	
0	IRQ_EN0	R/W	Sensor2 frame end IRQ enable/disable control register.	0: Disable SENFE IRQ 1: Enable SENFE IRQ

Bit	Function	Type	Description	Condition
[31:6]	-	-	Reserved	-
5	IRQ_STS5	R/W	Sensor2 output FIFO mode IRQ status register.	Read 0: SENOFIFOIRQ not happened. Read 1: SENOFIFOIRQ happened. Write 0: No effect.. Write 1: Clear this bit.
4	IRQ_STS4	R/W	Sensor2 receive FIFO over flow IRQ status register.	Read 0: SENOFIRQ not happened. Read 1: SENOFIRQ happened. Write 0: No effect.. Write 1: Clear this bit.
3			Reserved	
2			Reserved	
1			Reserved	
0	IRQ_STS0	R/W	Sensor2 frame end IRQ status register. This bit will be set to 1 if the sensor is completely capture a frame.	Read 0: SENFE IRQ not happened. Read 1: SENFE IRQ happened. Write 0: No effect.. Write 1: Clear this bit.

13.6 Light Pen Interface

The primary function of a light pen interface is to latch the luster position when composite TV is connected.

By toggling IOC [10], the luster position can be latched in register to determine the light pen position.

The following registers are the light pen interface's control registers.

P_PPU_LPHPOSITION		0x930200F8								Light Pen H position Register							
Bit		15	14	13	12	11	10	9	8								
Function		-								LPHPOSITION [9:8]							
Default		0	0	0	0	0	0	0	0								

7	6	5	4	3	2	1	0
LPHPOSITION [7:0]							
0	0	0	0	0	0	0	0

P_PPU_LPVPOSITION 0x930200FC Light Pen V position Register								
Bit	15	14	13	12	11	10	9	8
Function	-							LPVPOSITION [9:8]
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
LPVPOSITION [7:0]							
0	0	0	0	0	0	0	0

P_PPU_LIGHTPEN_CTRL 0x930200E4 Light Pen Control Register								
Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
LPMODE							
0	0	0	0	0	0	0	0

LPVPOSITION [9:0] => Vertical position of light pen, read only.

LPHPOSITION [9:0] => Horizontal position of light pen, read only.

LPMODE = 0 => Latches at every pulse of light pen.

LPMODE = 1 => Latches only the first one light pen pulse, the latch value will be locked as long as the LPMODE is "1".

13.7 TV Line Counter

The TV line counter is synchronous with the TV HSYNC. The line number in NTSC system is in the range from 0 to 262. The line number in PAL system is in the range from 0 to 312. The following register is used to show current line that TV is displaying.

P_PPU_LINE_COUNTER 0x930200E0 TV Line Counter								
Bit	15	14	13	12	11	10	9	8
Function	-							TV_LINECNT [9:8]
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TV_LINECNT [7:0]							
0	0	0	0	0	0	0	0

13.8 Frame Base Mode

The PPU inside GP12 supports frame base mode for system with high speed memory, i.e. page RAM or SDRAM. When frame base mode is enabled, the PPU can be asynchronous to TV or TFT output. However, additional bandwidth is required for writing data to system memory as well as reading data from system memory. The following table shows the bandwidth requirement for frame buffer mode.

Mode	Frame buffer Output (15 fps)	Frame buffer input (60 fps @ QVGA) (30 fps @ VGA)	Total.	Percentage @ ROM base (70ns RAM, 9T access)	Percentage @ SDRAM base (96 MHz, 70% utilization)
QVGA	2.304 MB/sec	9.216 MB/sec	11.52 MB/sec	54 %	8.57 %
VGA	9.216 MB/sec	18.432 MW/sec	27.468 MB/sec	130%	20.57%

According to table shown above, it is impossible to use frame base mode on the ROM/RAM base system; thus, we strongly suggest using SDRAM for this mode. The following table shows the differences between line base mode and the frame-buffer mode.

	Advantage	Disadvantage
Line Base Mode	1. No frame buffer memory needed. 2. No frame buffer input bandwidth needed.	1. The frame rate must be synchronous to TV or TFT which becomes very critical in 2.5D/rotate mode. 2. The critical line cannot share the bandwidth of non-critical line.
Frame Base Mode	1. The frame rate does not need to be synchronous to TV or TFT output. This will reduce lots of PPU bandwidth. 2. The overall frame will be computed completely without any lose. 3. The critical line can share non-critical line's bandwidth.	1. Additional bandwidth is necessary for frame buffer's output and input. 2. Require system memory to store frame buffer data.

In order to enable the frame buffer mode, the following register must be set.

PPU Control Register							
Bit	31	30	29	28	27	26	25
Function	-					DEFEN	TFTLB
Default	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
-	YUV_TYPE			LB	TFT_SIZE		
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO	SPR25D	FB_FORMAT	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
FB_EN	-	VGA_NOINTL	VGA_EN	TX_BOT_UP	TX_DIRECT	CH0_BLK	PPU_EN
0	0	0	0	0	0	0	1

VGA_EN is 0 => QVGA mode for TV

When only TFT-LCD is connected, the frame buffer size is depended on TFT_SIZE.

VGA_EN is 1 => VGA mode

VGA_NOINTL is 0 => VGA interlace mode, requiring $640 \times 480 \times 2 = 614,400$ bytes frame buffer (FB_MONO is 0).

VGA_NOINTL is 1 => VGA non-interlace mode, requiring $640 \times 240 \times 2 = 307,200$ bytes frame buffer (FB_MONO is 0).

FB_EN is 0 => Line base mode.

FB_EN is 1 => Frame-buffer mode.

FB_FORMAT is 0 => Frame buffer uses RGB565's format or mono format.

FB_FORMAT is 1 => Frame buffer uses RGBG or YUYV format (depends on FB_MONO [0]).

FB_MONO [1:0] is 0 => RGB565 or RGBG format frame buffer (16 bits per pixel).

FB_MONO [1:0] is 1 => Mono frame buffer (1 bit per pixel)

FB_MONO [1:0] is 2 => 4 Colors frame buffer (2 bit per pixel)

FB_MONO [1:0] is 3 => 16 Colors frame buffer (4 bit per pixel)

LB is 0 => The longest burst for PPU will be 8x32bit burst.

LB is 1 => The longest burst for PPU will be 16x32bit burst. It can improve performance.

TFT_LB is 0 => TFT frame buffer uses 8x32bit burst.

TFT_LB is 1 => TFT frame buffer uses 16x32bit burst (This mode can not be used under 720x480 mode).

The following table shows the frame buffer format under different conditions.

FB_FORMAT = 0		FB_FORMAT = 1	
FB_MONO [1:0]	Frame buffer format	FB_MONO [1:0]	Frame buffer format
0	RGB565	0	RGBG
1	Mono frame buffer (1 bit per pixel)	1	YUYV
2	4 Colors frame buffer (2 bit per pixel)	2	RGBG
3	16 Colors frame buffer (4 bit per pixel)	3	YUYV

TX_BOTUP is 0 => Normal operation as in line base mode.

TX_BOTUP is 1 => When the frame base mode is turned on and TX_BOTUP is set to "1", it will perform the blending correction process. The lower layer will be processed first to assure the correction of blending effect. Programmer must place sprite to lower depth in lower sprite number. For example, sprite 0 is in depth 0, sprite 1 in depth 1, sprite 2 in depth 2 and sprite 3 in depth 3. In this mode, the frame rate will decrease because of the waiting time period between sprite and text engine.

GP12 also supports dual operation mode, which means line-base and frame-base can work at the same time. When FB_EN is "0", the FB_SEL bit is able to control which module is using the line base, and the other is using frame base. See the descriptions below for details.

FB_SEL is 0 => TV uses line base mode and TFT uses frame base mode.

FB_SEL is 1 => TV uses frame base mode and TFT uses line base mode.

GP12 also supports dual frame buffer mode, which means both TV and TFT can display different data at the same time, but the required bandwidth will be doubled under this mode. The users can easily enable this function by only specifying two different FBI_ADDR.

The pointer of frame buffer input is defined in the following registers.

P_TV_FBI_ADDR									0x930201E0									TV Frame Buffer Input Address									
Bit	31	30	29	28	27	26	25	24																			
Function	TVFBI_ADDR [31:24]																										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16	TVFBI_ADDR [23:16]																		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	TVFBI_ADDR [15:8]																		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0	TVFBI_ADDR [7:0]																		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_FBI_ADDR									0x9302033C									TFT Frame Buffer Input Address									
Bit	31	30	29	28	27	26	25	24																			
Function	TFTFBI_ADDR [31:24]																										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
TFTFBI_ADDR [23:16]							
0	0	0	0	0	0	0	0
TFTFBI_ADDR [15:8]							
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
TFTFBI_ADDR [7:0]							
0	0	0	0	0	0	0	0

The TVFBI_ADDR indicates the frame buffer address for TV display, and similarly TFTFBI_ADDR represents the frame buffer address for TFT-LCD display. The TVFBI_ADDR [31:0] and TFTFBI_ADDR [31:0] must be 32 bytes alignment and so, the last 5 bits are fixed to "0" at all time. This address will take effect only when a TV or TFT-LCD frame is completed so that programmer is able to update this register at any time.

The pointer of frame buffer output is defined in the following registers.

P_PPU_FBO_ADDR 0x930201E8								Frame Buffer Output Address	
Bit	31	30	29	28	27	26	25	24	
Function	FBO_ADDR [31:24]								
Default	0	0	0	0	0	0	0	0	
23 22 21 20 19 18 17 16									
FBO_ADDR [23:16]									
0	0	0	0	0	0	0	0	0	
15 14 13 12 11 10 9 8									
FBO_ADDR [15:8]									
0	0	0	0	0	0	0	0	0	
7 6 5 4 3 2 1 0									
FBO_ADDR [7:0]									
0	0	0	0	0	0	0	0	0	

The FBO_ADDR is used to indicate the frame buffer address for PPU calculation output. The FBO_ADDR [31:0] must be 32 bytes alignment so that the last 5 bits are fixed to "0" at all time. This address will take effect only when a PPU frame is completed; therefore, programmer can update this register at any time.

In frame buffer mode, we recommend using double buffer mode for flow control; in other word, when PPU is updating frame buffer A, the display buffer can be set to frame buffer B. In default, the frame buffer mode will stop the PPU until CPU acknowledges PPU to be activated. The following registers are used to start the PPU computation.

P_PPU_FB_GO								0x930201F0								Frame Buffer PPU Go Register											
Bit	15	14	13	12	11	10	9	8																			
Function	TVFB1_UPD	FBO_F	TFTFB1_UPD	-																							
Default	0	0	0	0	0	0	0	0																			
	7	6	5	4	3	2	1	0									PPU_GO										
	0	0	0	0	0	0	0	0																			

In frame buffer mode, the PPU stops after reset; to start PPU activity, programmer must write any data to this port. Every time the PPU completes a frame, it will assert the vertical blanking interrupt and back to the stop state until the PPU_GO is written. Thus, if PPU's register or SRAM is not updated, programmer can stop the PPU to save bandwidth.

The TVFB1_UPD / TFTFB1_UPD bit is a read only bit that indicates the TVFB1_ADDR / TFTFB1_ADDR is updated to the current pointer; programmer can use this bit to determine whether the frame buffer input engine updates the pointer. The FBO_F bit is a read only bit that represents the field of next PPU frame. This bit is used only when VGA interlace mode is selected. If this bit is "0" at the vertical blank region, it means both even and odd fields have been updated. On the other hand, if this bit is "1" at the vertical blank region, it means the PPU will start calculating the odd field after PPU_GO is written.

For two frame buffer application, if the new frame buffer is not read by TV or TFT, and programmer set the PPU_GO again at this moment, it might result in the screen been cut in the middle. In order to prevent this kind of event, the following register are used to block the PPU_GO until the FBI_ADDR/FBI_ADDR2 are been updated to real TV/TFT output pointer.

P_PPU_Misc								0x01F8								PPU Misc Control Register								
Bit	23	22	21	20	19	18	17	16																
Function				TFT_3D				SBMP_MODE																
Default	0	0	0	0	0	0	0	0																
	15	14	13	12	11	10	9	8																
	TV_LB		TFTVTQ	DELGO	INTP_MODE	NEW_CMP	TXT_ALPHA																	
	0	0	0	0	0	0	0	0																

7	6	5	4	3	2	1	0
SP_ADDR_X2	DUAL_BLD			SPR_RGB	TXT_RGB	FB_LOCK	
0	0	0	0	0	0	0	0

DELGO is 0: The PPU_GO will not be blocked by any condition.

DELGO is 1: The PPU_GO will be delayed until both TVFBI_UPD and TFTFBI_UPD is 1.

During dual-frame buffer mode, if the output resolution is different between TV and TFT, the following register must be set to realize this feature.

P_PPU_Misc		0x930201F8				PPU Misc Control Register		
Bit	15	14	13	12	11	10	9	8
Function			TFTVTQ	-		INTP_MODE	NEW_CMP	TXT_ALPHA
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	DUAL_BLD		-	SPR_RGB	TXT_RGB	FB_LOCK	
0	0	0	0	0	0	0	0

FB_LOCK is 0 => The TV and TFT use current VGA_EN/VGA_NOINTL/TFT_SIZE setting. Not possible to set TV and TFT as different resolution.

FB_LOCK is 1 => TV will use VGA_EN/VGA_NOINTL/TFT_SIZE setting before turned on the TV. TFT will use TFT_SIZE setting before turned on the TFT. And these values will be locked after TV and TFT is turned-on, so the change these values will no longer affect the TV and TFT anymore. So the program flow will become like this.

Step1: Set VGA_EN/VGA_NOINTL/TFT_SIZE for TV

Step2: Enable TV encoder

Step3: Set TFT_SIZE for TFT

Step4: Enable TFT controller

Step5: Change VGA_EN/VGA_NOINTL/TFT_SIZE for current PPU's operation.

TFTVTQ is 0 => TFT's frame buffer and display size is the same.

TFTVTQ is 1 => TFT's frame buffer must be set to 1(PPU_SIZE is 1) and the display size is QVGA.

GP12 supports FIFO output mode when programmer wish to store the PPU's output with very high resolution but the available memory is not enough. The following registers are used to control the FIFO output mode.

P_FBO_FIFO_Setup		0x01D4 Frame Buffer Output FIFO Setup.						
Bit	31	30	29	28	27	26	25	24
Function							LINE_START[9:8]	
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
LINE_START[7:0]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	
FIFO_MODE		ADDR_OFFSET[13:8]						
0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
ADDR_OFFSET[7:0]							
0	0	0	0	0	0	0	0

ADDR_OFFSET[13:0]: Address offset of each line, this register is in byte unit. This value will be add to the output address at end of each line.

FIFO_MODE is 0 => No frame buffer output mode.

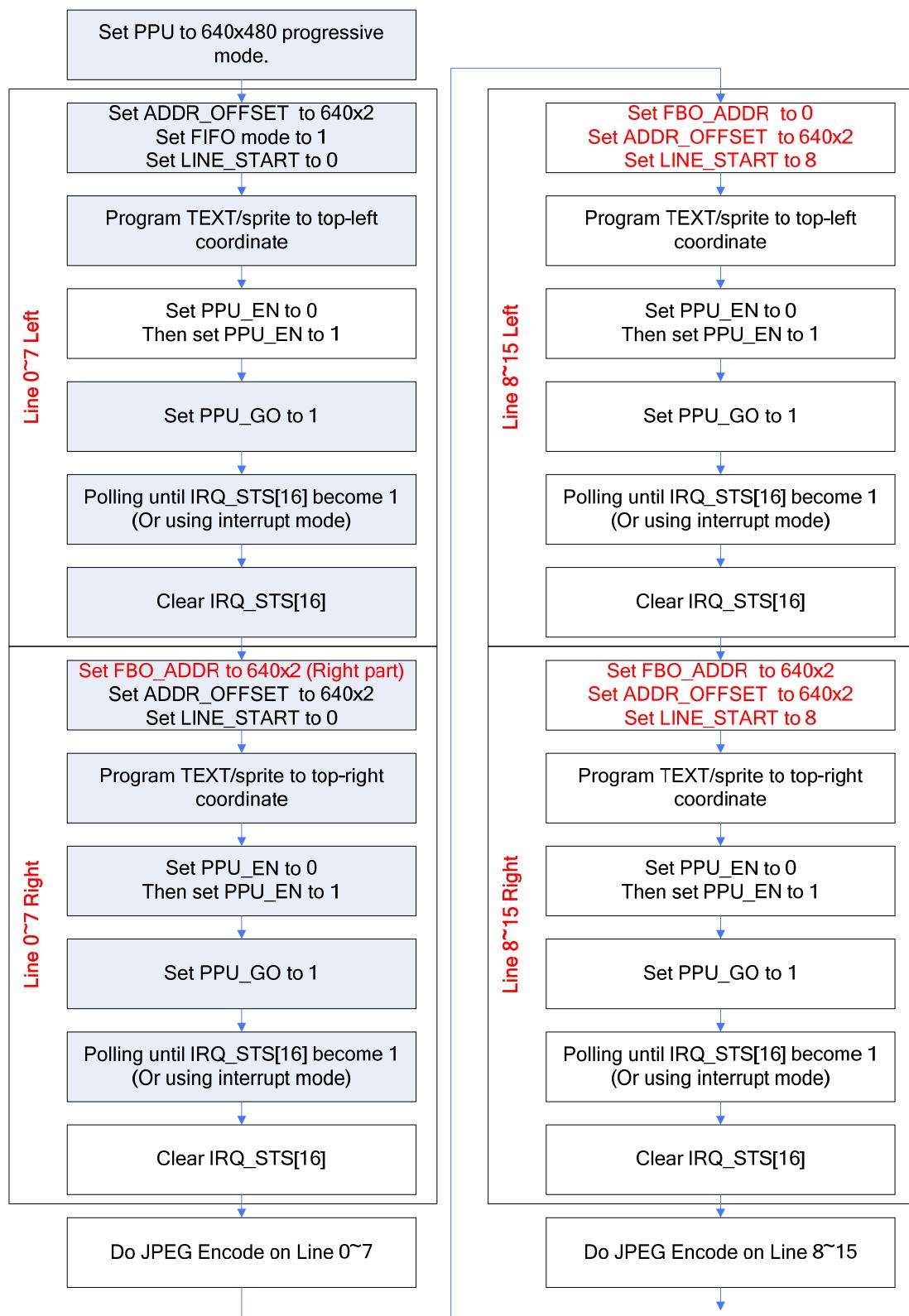
FIFO_MODE is 1 => 8 lines FIFO mode.

FIFO_MODE is 2 => 16 lines FIFO mode.

FIFO_MODE is 3 => 32 lines FIFO mode.

LINE_START[9:0] => Line start value when restart PPU for multi-time PPU process.

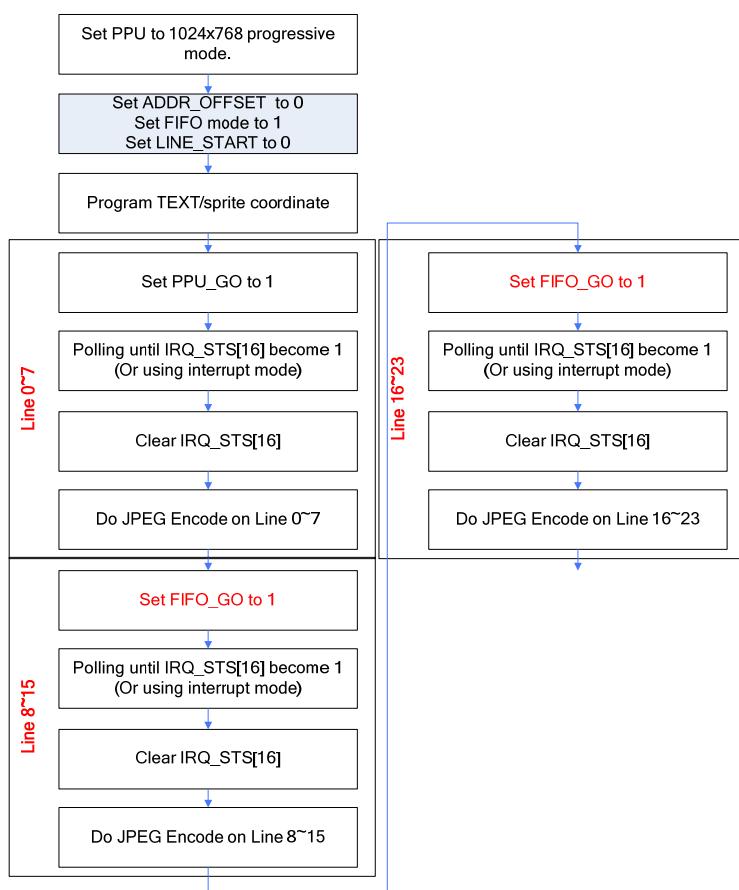
The following diagram shows an example when programmer wishes to use PPU to do a 1280x960 output with FIFO 8 mode.



When the output resolution is smaller than 1024x??, programmer can use the following register to re-start the PPU with same output address to minimize the memory usage.

P_FBO_FIFO_GO 0x01D8								Frame Buffer FIFO Go Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																FIFO_GO
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Frame buffer output will stop after the FIFO size is reach, so programmer can do the JPEG encode at this time. After the data is process completed, programmer can set FIFO_GO to 1 will re-start the PPU with the same output address. The following flow chart can be used under this condition.



When a 3D TFT panel is connected, GP12 supports a 3D display mode on the TFT panel, this feature can automatic transfer 2 individual frame buffer's data to the format 3D TFT panel is supported. The following register is used to control the 3D TFT mode.

P_PPU_Misc
0x01F8
PPU Misc Control Register

Bit	23	22	21	20	19	18	17	16
Function			TFT_3D					SBMP_MODE
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
TV_LB		TFTVTQ	DELGO	INTP_MODE	NEW_CMP	TXT_ALPHA	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ADDR_X2	DUAL_BLD			SPR_RGB	TXT_RGB	FB_LOCK	
0	0	0	0	0	0	0	0

TFT_3D is 0 => Normal 2D display mode.

TFT_3D is 1 => 3D display mode 1.

TFT_3D is 2 => 3D display mode 2.

P_TFT_3D_Offset
0x03A8 TFT Frame Buffer Offset in 3D mode.

Bit	15	14	13	12	11	10	9	8
Function						TFT_3D_OFFSET[10:8]		
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT_3D_OFFSET [7:0]							
0	0	0	0	0	0	0	0

P_TFTFBI_Addr_R
0x03AC TFT Right Frame Buffer Input Address.

Bit	31	30	29	28	27	26	25	24
Function								TFTFBI_ADDR_R[31:24]
Default	0	0	0	0	0	0	0	0

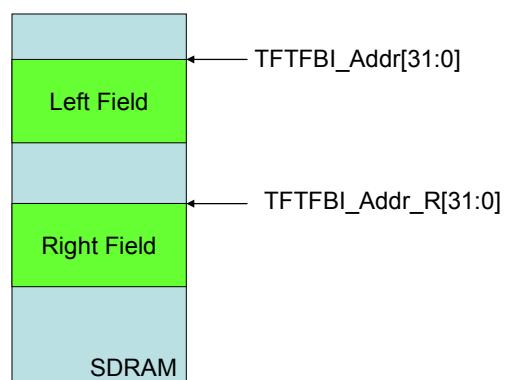
23	22	21	20	19	18	17	16
TFTFBI_ADDR_R[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
TFTFBI_ADDR_R[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFTFBI_ADDR_R[7:0]							
0	0	0	0	0	0	0	0

P_TFTFBI_Addr_R is used to specify the frame buffer address of the “right field”, at the same time, P_TFTFBI_Addr is used as the “Left field” input address. TFT_3D_OFFSET is used to specify the frame buffer width when left and right field of 3D picture is in the same frame buffer. The following figure shows two different type of 3D frame buffer.

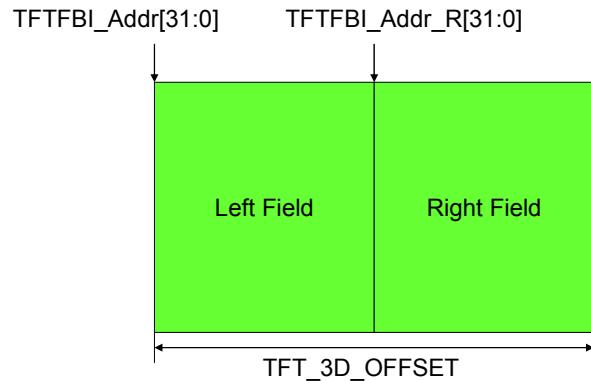
3D Frame buffer mode 1. (Individual Frame buffer mode)



TFT_3D_OFFSET must be 0 at this mode!!!

**When 3D panel has resolution 800x600 (400x600@3D mode),
Size of left field frame buffer is 400x600x2 = 480000 bytes.
Size of right field is the same.**

3D Frame buffer mode 2. (Single Frame buffer mode)

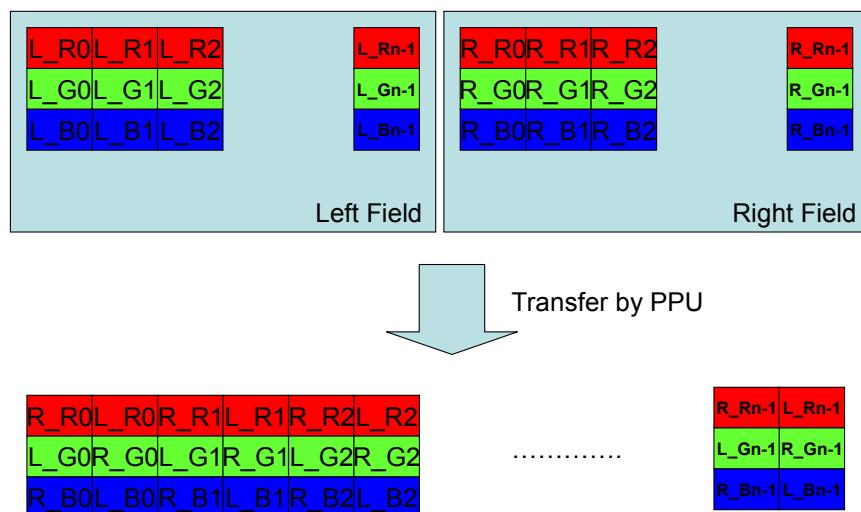


TFT_3D_OFFSET must be frame buffer width at this mode!!!

**When 3D panel has resolution 800x600 (400x600@3D mode),
Size of whole frame buffer is 800x600x2 = 960000 bytes.**

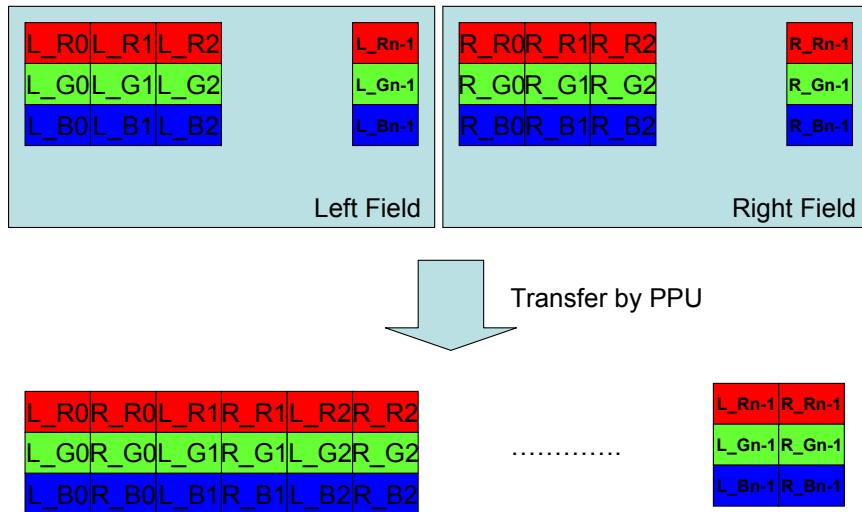
The following transform will be applied to the output data when the 3D mode is enabled.

3D mode 1 timing diagram.



When 3D panel has resolution 800x600 (400x600@3D mode) => n = 400

3D mode 2 timing diagram.



When 3D panel has resolution 800x600 (400x600@3D mode) => n = 400

13.9 Window Mode

GPL32900A supports four individual windows on the screen. Programmer can specify each TEXT sprite into one of these four windows. TEXT sprite will appear only when it is in the defined window region. These can have option to either priority difference or none by register configuration. The following registers are used to control the window selection of each TEXT sprite.

TXN_WINDOW is 0 => TXN is located at window 1 (default).

TXN WINDOW is 1 => TXN is located at window 2.

TXN WINDOW is 2 => TXN is located at window 3.

TXN WINDOW is 3 => TXN is located at window 4.

P_PPU_ENABLE								0x930201FC								PPU Control Register															
Bit	31	30	29	28	27	26	25	24									DEFEN		TFTLB												
Function	-																DEFEN		TFTLB												
Default	0	0	0	0	0	0	0	0									0		0		0		0								
23	22	21	20	19	18	17	16									YUV_TYPE		LB		TFT_SIZE											
-																	0		0		0		0								
15	14	13	12	11	10	9	8									SAVE_ROM		FB_SEL		SPR_WIN		HVCMP_DIS		FB_MONO		SPR25D		FB_FORMAT			
0	0	0	0	0	0	0	0									0		0		0		0		0							
7	6	5	4	3	2	1	0									FB_EN		-		VGA_NOINTL		VGA_EN		TX_BOTUP		TX_DIRECT		CH0_BLK		PPU_EN	
0	0	0	0	0	0	0	1									0		0		0		0		1							

SPR_WIN is 0 => Sprite does not support window function (default).

SPR_WIN is 1 => Sprite supports window function.

If SPR_WIN is "1", the window of each sprite is defined in the following registers.

P_PPU_SPRITEN_ATTRIBUTE1								0x93022008+(16*N)								Sprite N Attribute Register 1											
Bit	15	14	13	12	11	10	9	8									SPN_MOSAIC [1:0]		SPN_BLDLVL [5:2]				SPN_WIN [1:0]				
Function	SPN_MOSAIC [1:0]																SPN_MOSAIC [1:0]		SPN_BLDLVL [5:2]				SPN_WIN [1:0]				
Default	-								-								-		-				-				
7	6	5	4	3	2	1	0									SPN_PB[0]		SPN_CHARNUM [22:16]				-		-			
0	0	0	0	0	0	0	1									-		-				-		-			

Note: The sprite's blending level will be reduced to 16 levels when the window function is enabled.

To control the location and size of each window, the following register must be programmed correctly.

P_PPU_WINDOW_N_X								0x93020120 0x93020128 0x93020130								Window N X Control Register											
								0x93020138								(N = 1,2,3,4)											
Bit	31	30	29	28	27	26	25	24									MASKN		-				WINDOW_N_X [9:8]				
Function																			-				WINDOW_N_X [9:8]				
Default	0	0	0	0	0	0	0	0									0		0				0				
23	22	21	20	19	18	17	16									WINDOW_N_X [7:0]		-				0		0			
0	0	0	0	0	0	0	0									-		-				0		0			

15	14	13	12	11	10	9	8
-							WINDOWN_EX [9:8]

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
WINDOWN_EX [7:0]							

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

Window1: SX & EX & MASK1

Window2: SX & EX & MASK2

Window3: SX & EX & MASK3

Window4: SX & EX

P_PPU_WINDOW_N_Y **0x93020124 0x9302012C 0x93020134** **Window N Y Control Register**
0x9302013C **(N = 1,2,3,4)**

Bit	31	30	29	28	27	26	25	24	
Function	-								WINDOWN_SY [9:8]

Default	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---

23	22	21	20	19	18	17	16
WINDOWN_SY [7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
-							WINDOWN_EY [9:8]

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
WINDOWN_EY [7:0]							

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

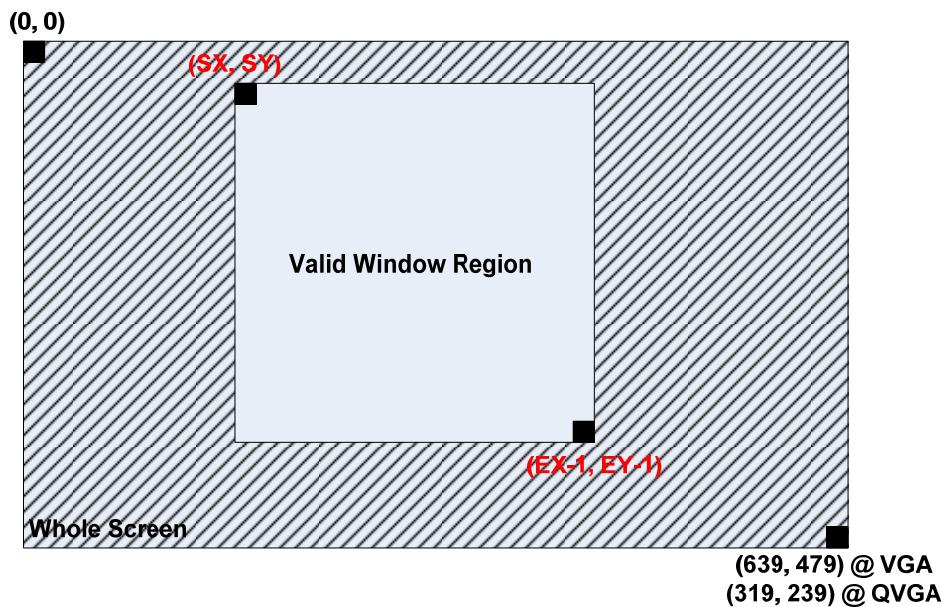
Window1: SY & EY

Window2: SY & EY

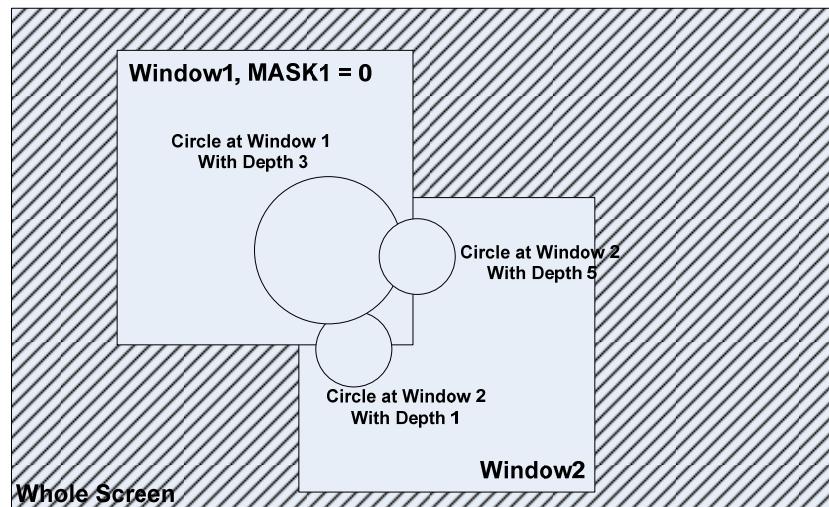
Window3: SY & EY

Window4: SY & EY

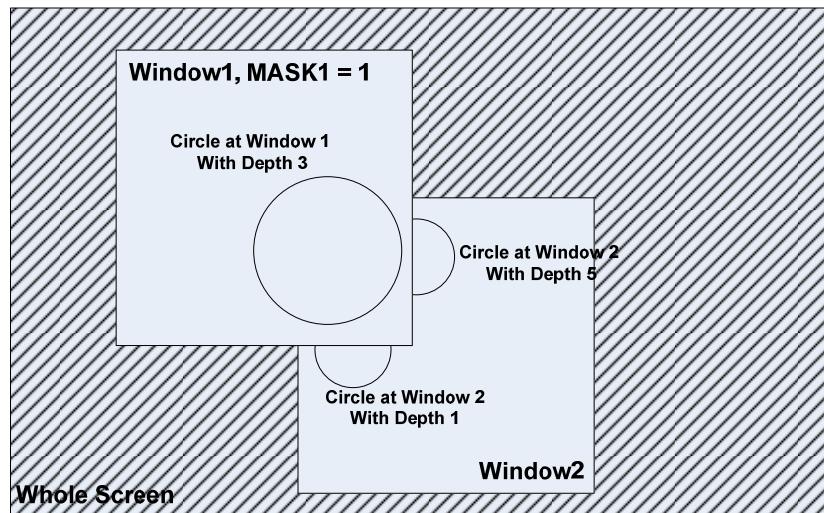
The definitions of SX/EX/SY/EY are as the following diagram.



When a TEXT or sprite is assigned to a specified window, it will only appear when it is in the valid window region. The coordinate system and depth priority are still the same under window mode. If a region is covered by multiple windows, two modes can be selected, causing display result to be affected. The following diagram shows the differences of these two modes.



When $\text{MASK1} = 0$, window1 will not block other windows



When MASK1 = 1, window1 will block window2/3/4

The priority of each window when MASK1 ~ MASK3 setting to "1" is Window1 > Window2 > Window3 > Window4.

13.10 Random Number Generator

There are two random number generators embedded in GPL32900A. The following two register are used to control these two random number generators.

P_RANDOM0 0x93020380								Random Number 0								
Bit	15	14	13	12	11	10	9	8								
Function	-								RANDOM0 [13:8]							
Default	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
								RANDOM0 [7:0]								
								0	0	0	1	1	0	0	0	0

P_RANDOM1 0x93020384								Random Number 1								
Bit	15	14	13	12	11	10	9	8								
Function	-								RANDOM1 [13:8]							
Default	0	0	0	1	0	1	1	0	0	0	1	1	0	0	0	0
								RANDOM1 [7:0]								
								0	1	0	1	1	0	0	0	0

The default values of these two random numbers are different; programmer can also write any initial values to these two registers at any time. These two random values have the same generation method as the following equation - $F(x) = x^{15} + x^{14} + 1$.

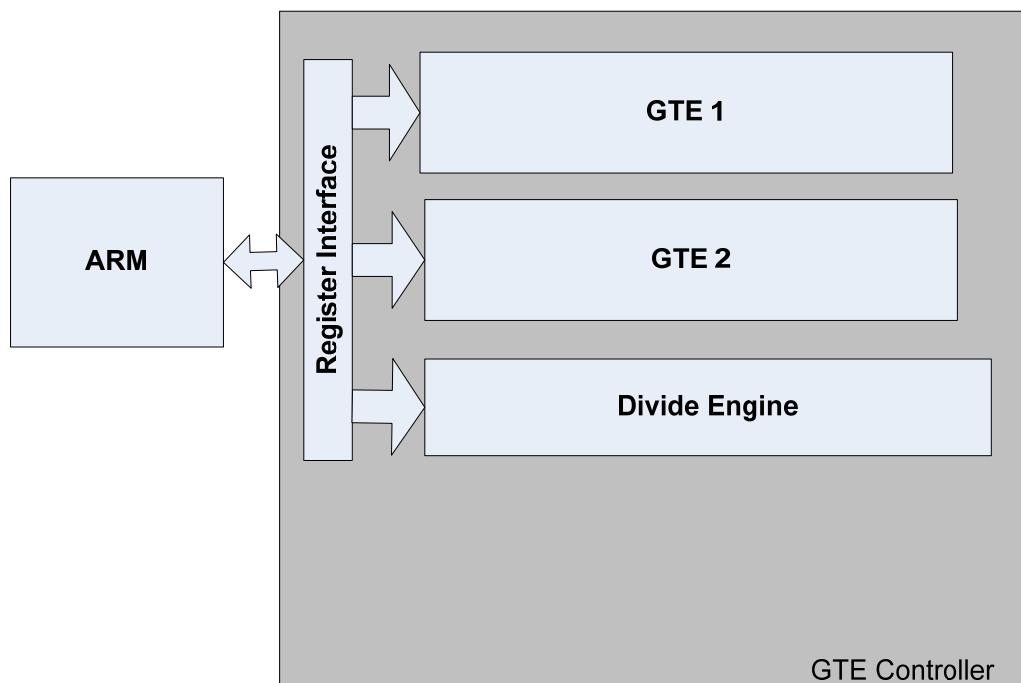
13.11 GTE Controller

The GTE (Geometry Transformation Engine) controller built-in GPL32900A is used to do matrix calculation and division operation for user's applications. The data format can be 1/15/16 or 1/31 for matrix operation and 1/15/16, 1/31, 16/16 or 32 bits for division operation.

Features:

- Support dual matrix operation unit which can do matrix operation independently.
- Support 4 kinds of matrix operations with 1/15/16 or 1/31 type of data format.
 - 4X4 X 4X1 matrix operation.
 - 3X3 X 3X1 matrix operation.
 - 1X3 X 3X1 inner product operation.
 - 3X1 X 1X3 outer product operation.
- Support overflow protection for above matrix operation.
- Support 8 clocks 32-bit division operation with residue output.
- Support 1/15/16, 1/31, 16/16 and 32 bits division operation.
- Support overflow protection under 1/15/16 and 16/16 division operation.

Block Diagram:



Register Location of GTE Controller:

Name	Address	Description
P_GTE0_ACT_M4X4	0xF6800000	GTE0 M4X4 active register.
P_GTE0_ACT_M3X3	0xF6800004	GTE0 M3X3 active register.
P_GTE0_ACT_INNER	0xF6800008	GTE0 inner product active register.

Name	Address	Description
P_GTE0_ACT_OUTER	0xF680000C	GTE0 outer product register.
P_GTE1_ACT_M4X4	0xF6800010	GTE1 M4X4 active register.
P_GTE1_ACT_M3X3	0xF6800014	GTE1 M3X3 active register.
P_GTE1_ACT_INNER	0xF6800018	GTE1 inner product active register.
P_GTE1_ACT_OUTER	0xF680001C	GTE1 outer product register.
P_GTE_A0	0xF6004000	GTE matrix A0 register.
P_GTE_A1	0xF6004004	GTE matrix A1 register.
P_GTE_A2	0xF6004008	GTE matrix A2 register.
P_GTE_A3	0xF600400C	GTE matrix A3 register.
P_GTE_A4	0xF6004010	GTE matrix A4 register.
P_GTE_A5	0xF6004014	GTE matrix A5 register.
P_GTE_A6	0xF6004018	GTE matrix A6 register.
P_GTE_A7	0xF600401C	GTE matrix A7 register.
P_GTE_A8	0xF6004020	GTE matrix A8 register.
P_GTE_A9	0xF6004024	GTE matrix A9 register.
P_GTE_AA	0xF6004028	GTE matrix AA register.
P_GTE_AB	0xF600402C	GTE matrix AB register.
P_GTE_AC	0xF6004030	GTE matrix AC register.
P_GTE_AD	0xF6004034	GTE matrix AD register.
P_GTE_AE	0xF6004038	GTE matrix AE register.
P_GTE_AF	0xF600403C	GTE matrix AF register.
P_GTE0_XI	0xF6004040	GTE0 matrix XI register.
P_GTE0_YI	0xF6004044	GTE0 matrix YI register.
P_GTE0_ZI	0xF6004048	GTE0 matrix ZI register.
P_GTE0_WI	0xF600404C	GTE0 matrix WI register.
P_GTE1_XI	0xF6004050	GTE1 matrix XI register.
P_GTE1_YI	0xF6004054	GTE1 matrix YI register.
P_GTE1_ZI	0xF6004058	GTE1 matrix ZI register.
P_GTE1_WI	0xF600405C	GTE1 matrix WI register.
P_GTE0_XO	0xF6004060	GTE0 matrix XO register.
P_GTE0_YO	0xF6004064	GTE0 matrix YO register.
P_GTE0_ZO	0xF6004068	GTE0 matrix ZO register.
P_GTE0_WO	0xF600406C	GTE0 matrix WO register.
P_GTE1_XO	0xF6004070	GTE1 matrix XO register.
P_GTE1_YO	0xF6004074	GTE1 matrix YO register.
P_GTE1_ZO	0xF6004078	GTE1 matrix ZO register.
P_GTE1_WO	0xF600407C	GTE1 matrix WO register.
P_GTE_MODE	0xF6004080	GTE current mode status register.

Name	Address	Description
P_GTE_FORMAT	0xF6004084	GTE format control register.
P_GTE0_OF	0xF600408C	GTE0 overflow detection register.
P_GTE1_OF	0xF600409C	GTE1 overflow detection register.
P_GTE_DIVA	0xF60040A0	GTE divider dividend register.
P_GTE_DIVB	0xF60040A4	GTE divider divisor register.
P_GTE_DIVOF	0xF60040A8	GTE divider overflow register.
P_GTE_DIVO	0xF60040AC	GTE divider quotient register.
P_GTE_DIVR	0xF60040B0	GTE divider residue register.

The P_GTE0_ACT_M4X4 register is used to active a 4x4 matrix operation on GTE0.

P_GTE0_ACT_M4X4 0xF6800000 GTE0 M4X4 Active Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function								ACT_M4X4
Default	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:1	-	-	Reserved.	-
0	ACT_M4X4	W	GTE 0 Active M4X4 operation control register.	Write 0 = no effect. Write 1 = Start a M4X4 operation on GTE0.

The P_GTE0_ACT_M3X3 register is used to active a 3x3 matrix operation on GTE0.

P_GTE0_ACT_M3X3 0xF6800004 GTE0 M3X3 Active Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function								ACT_M3X3
Default	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:1			Reserved.	
0	ACT_M3X3	W	GTE 0 Active M3X3 operation control register.	Write 0 = no effect. Write 1 = Start a M3X3 operation on GTE0.

The P_GTE0_ACT_INNER register is used to active an inner product operation on GTE0.

P_GTE0_ACT_INNER 0xF6800008 GTE0 Inner Product Active Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:1			Reserved.	
0	ACT_INNER	W	GTE 0 Active inner product operation control register.	Write 0 = no effect. Write 1 = Start an inner product operation on GTE0.

The P_GTE0_ACT_OUTER register is used to active an outer product operation on GTE0.

P_GTE0_ACT_OUTER 0xF680000C GTE0 Outer Product Active Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:1			Reserved.	
0	ACT_OUTER	W	GTE 0 Active outer product operation control register.	Write 0 = no effect. Write 1 = Start an outer product operation on GTE0.

The P_GTE1_ACT_M4X4 register is used to active a 4x4 matrix operation on GTE1.

P_GTE1_ACT_M4X4 0xF6800010 GTE1 M4X4 Active Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

ACT_M4X4

Bit	Function	Type	Description	Condition
31:1			Reserved.	
0	ACT_M4X4	W	GTE 1 Active M4X4 operation control register.	Write 0 = no effect. Write 1 = Start a M4X4 operation on GTE1.

The P_GTE1_ACT_M3X3 register is used to active a 3x3 matrix operation on GTE1.

		GTE1 M3X3 Active Register							
Bit	15	14	13	12	11	10	9	8	
Function									
Default	0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

ACT_M3X3

Bit	Function	Type	Description	Condition
31:1			Reserved.	
0	ACT_M3X3	W	GTE 1 Active M3X3 operation control register.	Write 0 = no effect. Write 1 = Start a M3X3 operation on GTE1.

The P_GTE1_ACT_INNER register is used to active an inner product operation on GTE1.

		GTE1 Inner Product Active Register							
Bit	15	14	13	12	11	10	9	8	
Function									
Default	0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

ACT_INNER

Bit	Function	Type	Description	Condition
31:1			Reserved.	
0	ACT_INNER	W	GTE 1 Active inner product operation control register.	Write 0 = no effect. Write 1 = Start an inner product operation on GTE1.

The P_GTE1_ACT_OUTER register is used to active an outer product operation on GTE1.

P_GTE1_ACT_OUTER 0xF680001C GTE1 Outer Product Active Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
								ACT_OUTER
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:1			Reserved.	
0	ACT_OUTER	W	GTE 1 Active outer product operation control register.	Write 0 = no effect. Write 1 = Start an outer product operation on GTE1.

The following equation shows the GTE operation result under different mode.

$$\begin{bmatrix} A0 & A4 & A8 & AC \\ A1 & A5 & A9 & AD \\ A2 & A6 & AA & AE \\ A3 & A7 & AB & AF \end{bmatrix} \times \begin{bmatrix} XI \\ YI \\ ZI \\ WI \end{bmatrix} = \begin{bmatrix} XO \\ YO \\ ZO \\ WO \end{bmatrix}$$

$$XO = A0 * XI + A4 * YI + A8 * ZI + AC * WI$$

$$YO = A1 * XI + A5 * YI + A9 * ZI + AD * WI$$

$$ZO = A2 * XI + A6 * YI + AA * ZI + AE * WI$$

$$WO = A3 * XI + A7 * YI + AB * ZI + AF * WI$$

Equation 1. Description of M4X4 operation

$$\begin{bmatrix} A0 & A4 & A8 \\ A1 & A5 & A9 \\ A2 & A6 & AA \end{bmatrix} \times \begin{bmatrix} XI \\ YI \\ ZI \end{bmatrix} = \begin{bmatrix} XO \\ YO \\ ZO \end{bmatrix}$$

$$XO = A0 * XI + A4 * YI + A8 * ZI$$

$$YO = A1 * XI + A5 * YI + A9 * ZI$$

$$ZO = A2 * XI + A6 * YI + AA * ZI$$

Equation 2. Description of M3X3 operation

$$\begin{bmatrix} A0 & A4 & A8 \end{bmatrix} \times \begin{bmatrix} XI \\ YI \\ ZI \end{bmatrix} = XO$$

$$XO = A0 * XI + A4 * YI + A8 * ZI$$



Equation 3. Description of inner product operation

$$\begin{vmatrix} 1 & 1 & 1 \\ XI & YI & ZI \\ A8 & A0 & A4 \end{vmatrix}$$

$$XO = A0 * XI + A4 * YI + A8 * ZI - A4 * XI - A8 * YI - A0 * ZI$$

Equation 4. Description of outer product operation

The following tables shows the necessary cycles under each operation mode.

Mode	Cycles
M4X4	17
M3X3	10
Inner Product	4
Outer Product	7

The P_GTE_AX register is used to program A0~AF registers for GTE0 and GTE1.

P_GTE_AX		0xF6004000~0xF600403C						GTE AX Register			
Bit	31	30	29	28	27	26	25	24			
Function	AX[31:24]										
Default	0	0	0	0	0	0	0	0			
23	22	21	20	19	18	17	16				
AX[23:16]											
0	0	0	0	0	0	0	0				
15	14	13	12	11	10	9	8				
AX[15:8]											
0	0	0	0	0	0	0	0				
7	6	5	4	3	2	1	0				
AX[7:0]											
0	0	0	0	0	0	0	0				

Bit	Function	Type	Description	Condition
31:0	AX	R/W	GTE A0~AF register. The data format of these register is determine by the P_GTE_FORMAT register.	

The P_GTE0_XI/YI/ZI/WI register is used to program XI~WI registers for GTE0.

P_GTE0_XI/YI/ZI/WI 0xF6004040~0xF600404C GTE0 XI/YI/ZI/WI Register								
Bit	31	30	29	28	27	26	25	
Function	XI/YI/ZI/WI[31:24]							
Default	0	0	0	0	0	0	0	
23	22	21	20	19	18	17	16	
XI/YI/ZI/WI[23:16]								
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
XI/YI/ZI/WI[15:8]								
0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	
XI/YI/ZI/WI[7:0]								
0	0	0	0	0	0	0	0	
Bit	Function	Type	Description					Condition
31:0	XI/YI/ZI/WI	R/W	GTE0 XI~WI register. The data format of these register is determine by the P_GTE_FORMAT register.					

The P_GTE1_XI/YI/ZI/WI register is used to program XI~WI registers for GTE1.

P_GTE1_XI/YI/ZI/WI 0x F6004050~0xF600405C GTE1 XI/YI/ZI/WI Register								
Bit	31	30	29	28	27	26	25	
Function	XI/YI/ZI/WI[31:24]							
Default	0	0	0	0	0	0	0	
23	22	21	20	19	18	17	16	
XI/YI/ZI/WI[23:16]								
0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	
XI/YI/ZI/WI[15:8]								
0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	
XI/YI/ZI/WI[7:0]								
0	0	0	0	0	0	0	0	

Bit	Function	Type	Description					Condition
31:0	XI/YI/ZI/WI	R/W	GTE1 XI~WI register. The data format of these register is determine by the P_GTE_FORMAT register.					

The P_GTE0_XO/YO/ZO/WO register is used to read XO~WO registers of GTE0.

P_GTE0_XO/YO/ZO/WO 0xF6004060~0xF600406C GTE0 XO/YO/ZO/WO Register

Bit	31	30	29	28	27	26	25	24
Function	XO/YO/ZO/WO[31:24]							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
XO/YO/ZO/WO[23:16]								
0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	
XO/YO/ZO/WO[15:8]								
0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	
XO/YO/ZO/WO[7:0]								
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
31:0	XO/YO/ZO/WO	R	GTE0 XO~WO register. The data format of these register is determine by the P_GTE_FORMAT register.					

The P_GTE1_XO/YO/ZO/WO register is used to read XO~WO registers of GTE1.

P_GTE1_XO/YO/ZO/WO 0xF6004060~0xF600406C GTE1 XO/YO/ZO/WO Register

Bit	31	30	29	28	27	26	25	24
Function	XO/YO/ZO/WO[31:24]							
Default	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	
XO/YO/ZO/WO[23:16]								
0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
XO/YO/ZO/WO[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
XO/YO/ZO/WO[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
31:0	XO/YO/ZO/WO	R	GTE1 XO~WO register. The data format of these register is determine by the P_GTE_FORMAT register.					

The P_GTE_MODE register is used to read the current GTE mode of GTE0 and GTE1.

P_GTE_MODE
0xF6004080
GTE Mode Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
7 6 5 4 3 2 1 0								
MODE1 MODE0								
0 0 0 0 0 0 0 0 0								

Bit	Function	Type	Description					Condition
31:4	-	-	Reserved.					-
3:2	MODE1	R	This register represents the current operation of GTE1.					0: M4X4 1: M3X3 2: Inner 3: Outer
1:0	MODE0	R	This register represents the current operation of GTE0.					0: M4X4 1: M3X3 2: Inner 3: Outer

The P_GTE_MODE register is used to control the data for of GTE0 and GTE1.

P_GTE_FORMAT
0xF6004084
GTE Format Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0



7	6	5	4	3	2	1	0
							FORMAT
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:2	-	-	Reserved.	-
1:0	FORMAT	R/W	This register is used to control the data format of GTE controller.	0: 1/15/16 1: 1/31 Mode 1 2: 1/31 Mode 2 3: Reserved.

The following table shows the definition of each data format.

Mode	Description	Example	Maximum	Minimum
1/15/16	1 sign bit, 15 bits integer, 16 bits fraction mode.	0x10000 = 1.00 0x18000 = 1.50 0x08000 = 0.50 0x10000*0x10000=0x10000	0x7FFFFFFF (32767.9999)	0x80000000 (-32768)
1/31 Mode 1	1 sign bit, 31 bits integer mode, output take high 32 bits result.	0x1 = 1 0x10000 = 65536 0xFFFFE0000 = -65536 0x10000*0x10000=0x1	0x7FFFFFFF (2^{31} -1)	0x80000000 (- 2^{31})
1/31 Mode 2	1 sign bit, 31 bits integer mode, output take low 32 bits result.	0x1 = 1 0x10000 = 65536 0xFFFFE0000 = -65536 0x8000*0x8000=0x40000000	0x7FFFFFFF (2^{31} -1)	0x80000000 (- 2^{31})

The P_GTE0_OF register is used to return the overflow status of GTE0.

P GTE0_OF 0xF600408C GTE0 Overflow Register

Bit	Function	Type	Description	Condition
31:5			Reserved.	
4	ERROR0	R/C	GTE0 error active flag, this flag will be set when a new active pulse is received during the GTE operation cycles.	Read 0: Not happened. Read 1: Happened. Write 0: No effect. Write 1: Clear this flag.

Bit	Function	Type	Description				Condition
3	XO0_OF	R	GTE0 XO overflow flag.				0: Not over flow 1: overflow.
2	YO0_OF	R	GTE0 YO overflow flag.				0: Not over flow 1: overflow.
1	ZO0_OF	R	GTE0 ZO overflow flag.				0: Not over flow 1: overflow.
0	WO0_OF	R	GTE0 WO overflow flag.				0: Not over flow 1: overflow.

The P_GTE1_OF register is used to return the overflow status of GTE1.

P_GTE1_OF			0xF600409C								GTE1 Overflow Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function																		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
					ERROR1	XO1_OF	YO1_OF	ZO1_OF	WO1_OF									
				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
31:5	-	-	Reserved.				-
4	ERROR1	R/C	GTE1 error active flag, this flag will be set when a new active pulse is received during the GTE operation cycles.				Read 0: Not happened. Read 1: Happened. Write 0: No effect. Write 1: Clear this flag.
3	XO1_OF	R	GTE1 XO overflow flag.				0: Not over flow 1: overflow.
2	YO1_OF	R	GTE1 YO overflow flag.				0: Not over flow 1: overflow.
1	ZO1_OF	R	GTE1 ZO overflow flag.				0: Not over flow 1: overflow.
0	WO1_OF	R	GTE1 WO overflow flag.				0: Not over flow 1: overflow.

The P_GTE_DIVA register is used to program dividend for divider.

P_GTE_DIVA			0xF60040A0								GTE DIVA Register							
Bit	31	30	29	28	27	26	25	24	DIVA[31:24]									
Function																		
Default	0	0	0	0	0	0	0	0										

23	22	21	20	19	18	17	16
DIVA[23:16]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
DIVA[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
DIVA[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
31:0	DIVA	R/W	GTE divider dividend register. The data format of this register is controller by the P_GTE_DIVOF register.					-

The P_GTE_DIVB register is used to program divisor for divider.

P_GTE_DIVB
0xF60040A4
GTE DIVB Register

Bit	31	30	29	28	27	26	25	24
Function	DIVB[31:24]							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
DIVB[23:16]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
DIVB[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
DIVB[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
31:0	DIVB	R/W	GTE divider divisor register. The data format of this register is controller by the P_GTE_DIVOF register.					

The P_GTE_DIVOF register is used to return the overflow status of divider and control the data format of divider.

P_GTE_DIVOF 0xF60040A8 Divider Overflow Register								
Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
					DIV_OF	DIV_ERROR	DIV_131	DIV_UNS
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:4	-	-	Reserved.	-
3	DIV_OF	R	Divider overflow flag, this bit will be set only when DIV_131 is 0.	0: Not over flow 1: overflow.
2	DIV_ERROR	R	Divider error active flag, this flag will be set when a new value is written to DIVB the divider operation cycles.	Read 0: Not happened. Read 1: Happened. Write 0: No effect. Write 1: Clear this flag.
1	DIV_131	R/W	Divider format control register.	0: 1/15/16 or 16/16 1: 1/31 or 32 bits
0	DIV_UNS	R/W	Divider signed/unsigned control register.	0: Signed (1/15/16 or 1/31). 1: Unsigned (16/16 or 32).

The following table shows the definition of each data format.

Mode	Description	Example	Maximum	Minimum
1/15/1 6	1 sign bit, 15 bits integer, 16 bits fraction mode. DIV_131 is 0, DIV_UNS is 0.	0x10000 = 1.00 0x18000 = 1.50 0x08000 = 0.50 0x10000/0x10000=0x10000	0x7FFFFFFF (32767.9999)	0x80000000 (-32768)
16/16	16 bits integer, 16 bits fraction mode. DIV_131 is 0, DIV_UNS is 1.	0x10000 = 1.00 0x18000 = 1.50 0x08000 = 0.50 0x10000/0x10000=0x10000	0xFFFFFFFF (65535.9999)	0x00000000 (0)
1/31	1 sign bit, 31 bits integer mode. DIV_131 is 1, DIV_UNS is 0.	0x1 = 1 0x10000 = 65536 0xFFFFE0000 = -65536 0x10000/0x10000=0x1	0x7FFFFFFF ($2^{31}-1$)	0x80000000 (- 2^{31})

Mode	Description	Example	Maximum	Minimum
32	32 bits integer mode. DIV_131 is 1, DIV_UN is 1.	0x1 = 1 0x10000 = 65536 0xFFFFE0000 = 4294836224 0x10000/0x10000=0x1	0xFFFFFFFF ($2^{32}-1$)	0x00000000 (0)

The P_GTE_DIVO register is used to read the quotient output of divider.

P_GTE_DIVO 0xF60040AC GTE DIVO Register								
Bit	31	30	29	28	27	26	25	24
Function	DIVO[31:24]							
Default	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	DIVO[23:16]							
Default	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	DIVO[15:8]							
Default	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	DIVO[7:0]							
Default	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
31:0	DIVO	R/W	GTE divider quotient output register. The data format of this register is controller by the P_GTE_DIVOF register.						

The P_GTE_DIVR register is used to read the residue output of divider.

P_GTE_DIVR 0xF60040B0 GTE DIVR Register								
Bit	31	30	29	28	27	26	25	24
Function	DIVR[31:24]							
Default	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	DIVR[23:16]							
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
DIVR[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
DIVR[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
31:0	DIVR	R/W	GTE divider residue output register. The data format of this register is controller by the P_GTE_DIVOFR register.					

13.12 ROM Saving Mode

GP12 is equipped with a ROM saving mode which can save 50% of ROM space via reducing the vertical resolution. This is achieved by PPU's hardware using even line's data as odd line's data. This mode can be turned on by implementing the following register.

P_PPU_ENABLE 0x930201FC PPU Control Register									
Bit	31	30	29	28	27	26	25	24	
Function	-							DEFEN	TFTLB
Default	0	0	0	0	0	0	0	0	
CM_EN YUV_TYPE LB TFT_SIZE									
23	22	21	20	19	18	17	16		
CM_EN	YUV_TYPE			LB	TFT_SIZE				
0	0	0	0	0	0	0	0		
15	14	13	12	11	10	9	8		
SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO		SPR25D	FB_FORMAT		
0	0	0	0	0	0	0	0		
7	6	5	4	3	2	1	0		
FB_EN	-	VGA_NOINTL	VGA_EN	TX_BOTUP	TX_DIRECT	CH0_BLK	PPU_EN		
0	0	0	0	0	0	0	1		

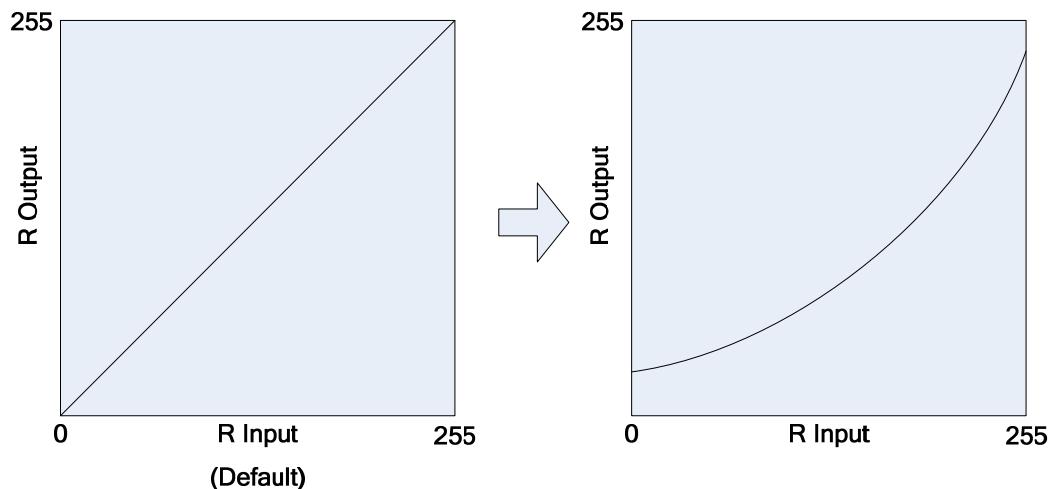
SAVE_ROM is 0 => Normal mode.

SAVE_ROM is 1 => ROM Saving mode.

The character size under the ROM saving mode is half compare to the normal mode. But this mode will not affect the bitmap mode because the start address of each line under bitmap mode is determined by programmer. But the special bitmap mode without start address will be affected by the SAVE_ROM setting.

13.13 Color Mapping

The color mapping function is used to control the overall color output for TFT-LCD display only when different output terminal is connected. For example, an image may display correctly on TV, but probably not correct on TFT-LCD panel. Programmer can adjust this table to fine tune the RGB mapping curve to get the correctly color mapping result. The following figure shows an example of color mapping.



To enable the color mapping function, the following register must be set to 1.

PPU Control Register							
Bit	31	30	29	28	27	26	25
Function						DEFEN	TFTLB
Default	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
CM_EN	YUV_TYPE			LB	TFT_SIZE		
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO		SPR25D	FB_FORMAT
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
FB_EN	-	VGA_NOINTL	VGA_EN	BLD_COR	TX_DIRECT	CH0_BLK	PPU_EN
0	0	0	0	0	0	0	1

CM_EN is 0: Disable color mapping function.

CM_EN is 1: Enable color mapping function.

Note: When users want to use the color mapping function, the bit13 of P_TFT_TS_MISC must be also set to 1.

The color mapping table is stored in the color mapping RAM. The initial value of these RAM will be linear mapping which means input and output colors are identical.

NAME	ADDR	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_PPU_TFT_COLOR_MAP_B_0	0x93027000																B_OUTPUT_0 [7:0]
P_PPU_TFT_COLOR_MAP_B_1	0x93027004																B_OUTPUT_1 [7:0]
...
P_PPU_TFT_COLOR_MAP_B_254	0x930273F8																B_OUTPUT_254 [7:0]
P_PPU_TFT_COLOR_MAP_B_255	0x930273FC																B_OUTPUT_255 [7:0]
P_PPU_TFT_COLOR_MAP_G_0	0x93027400																G_OUTPUT_0 [7:0]
P_PPU_TFT_COLOR_MAP_G_1	0x93027404																G_OUTPUT_1 [7:0]
...
P_PPU_TFT_COLOR_MAP_G_254	0x930277F8																G_OUTPUT_254 [7:0]
P_PPU_TFT_COLOR_MAP_G_255	0x930277FC																G_OUTPUT_255 [7:0]
P_PPU_TFT_COLOR_MAP_R_0	0x93027800																R_OUTPUT_0 [7:0]
P_PPU_TFT_COLOR_MAP_R_1	0x93027804																R_OUTPUT_1 [7:0]
...
P_PPU_TFT_COLOR_MAP_R_254	0x93027BF8																R_OUTPUT_254 [7:0]
P_PPU_TFT_COLOR_MAP_R_255	0x93027BFC																R_OUTPUT_255 [7:0]

A special condition is the transparent color; when a color is selected as transparent, it will display black on the screen which will not be affected by this color mapping table.

13.14 PPU Special Effect

There are three special effects supported on both TEXT and sprite: gray scale, mono color, and negative color mode. The special effect of each TEXT or sprite can be turned on/off individually and each element can have its effect. The following register shows the control register for TEXT layer.

P_PPU_TEXTN_AT 0x93020048 0x93020060 0x93020010 TEXT N Attribute Register

TRIBUTE								0x93020030								(N=1,2,3,4)									
Bit	31	30	29	28	27	26	25	24	TXN_CMASK																
Function	-	-	-	-	-	-	-	TXN_CMASK																	
Default	0	0	0	0	0	0	0	0	TXN_CMASK																
Bit	23	22	21	20	19	18	17	16	TXN_CMASK																
Function	TXN_PB[1:0]		TXN_EFF			-	TXN_WINDOW			TXN_SIZE[2]															
Default	0	0	0	0	0	0	0	0	TXN_SIZE[2]																
15								TXN_DEPTH								TXN_PALETTE									
0								TXN_DEPTH								TXN_PALETTE									
0								TXN_DEPTH								TXN_PALETTE									

7	6	5	4	3	2	1	0
TXN_VS		TXN_HS		TXN_FLIP		TXN_COLOR	
0	0	0	0	0	0	0	0

P_PPU_TEXTN_CTRL **0x9302004C** **0x93020064** **0x93020014** **TEXT N Control Register**
0x93020034 **(N=1,2,3,4)**

Bit	23	22	21	20	19	18	17	16
Function	-						-	INTP
default	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	TXN_BLDLVL						TXN_BLD MODE	TXN_BLD
	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP	
0	0	0	0	0	0	0	0

Note: TXN_RGBM is 1 and TXN_COLOR is 3: Enable Special Effect Function, which means only color YUV type can be used in special effect mode.

TXN_EFF is 0 => No special effect is applied.

TXN_EFF is 1 => Negative color effect.

TXN_EFF is 2 => Gray Scale Effect.

TXN_EFF is 3 => Mono Color Effect.

The following register shows the control register for sprites.

P_PPU_SPRITE_CTRL **0x93020108** **Sprite Control Register**

Bit	23	18	21	20	19	18	17	16
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SPRITE_NUMBER							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN
0	0	0	0	0	0	0	0

SP_EFFEN is 1 => Enable sprite special function.

P_PPU_SPRITEN_ATTRIBUTE0								0x93022006+(16*N)								Sprite N Attribute Register 0								
Bit	15	14	13	12	11	10	9	8	SPN_PALETTE															
Function	SPB_PB[1]	SPN_BLD	SPN_DEPTH									SPN_VS	SPN_HS	SPN_FLIP	SPN_COLOR									
Default	-	-	-	-	-	-	-	-																
	7	6	5	4	3	2	1	0																
	SPN_VS								SPN_HS								SPN_FLIP							
	-	-	-	-	-	-	-	-																

SPN_FLIP is 0 => No special effect function.

SPN_EFF is 1 => Negative color effect.

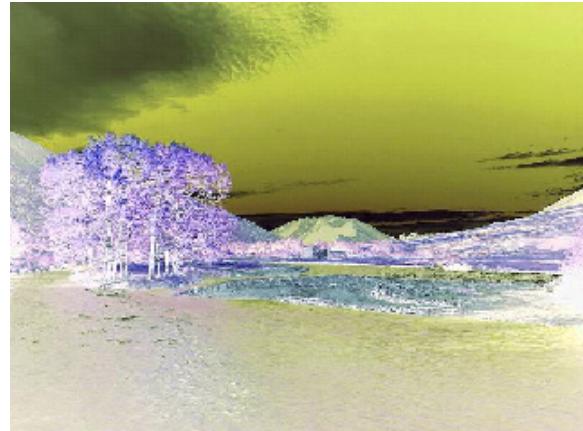
SPN_EFF is 2 => Gray Scale Effect.

SPN_EFF is 3 => Mono Color Effect.

The following figures show the result of each effect.



Original Picture



Negative Effect



Gray Level Effect



Mono Color Effect

When mono color effect is selected, the color can be adjusted by the following register.

P_PPU_RGB_OFFSET								0x9302037C								RGB Offset Register									
Bit	15	14	13	12	11	10	9	8																	
Function	-								R_OFFSET [3:0]																
Default	-	-	-	-	-	-	-	-																	
												G_OFFSET [3:0]													
												B_OFFSET [3:0]													

R_OFFSET: R offset value after transfer to gray scale, New R = (Y * R_OFFSET) / 8

G_OFFSET: G offset value after transfer to gray scale, New G = (Y * G_OFFSET) / 8

B_OFFSET: B offset value after transfer to gray scale, New B = (Y * B_OFFSET) / 8

The following figures show the result under various offset values.



R_OFFSET=0xC, G_OFFSET=0x8, B_OFFSET=0x8



R_OFFSET=0x8, G_OFFSET=0xC, B_OFFSET=0x8



R_OFFSET=0x8, G_OFFSET=0x8, B_OFFSET=0xC

13.15 Horizontal Blank Interrupt

The PPU built in GP12 supports horizontal blank interrupt which can stop PPU at specific line. This feature allows programmer to change the PPU's parameters at specific line to recognize some special effects, e.g. the dual screen mode. The following register is used to control the horizontal blank IRQ.

P_PPU_HB_CTRL 0x930201CC

Horizontal Blank Control Register

Bit	15	14	13	12	11	10	9	8
Function	HB_EN			-			HB_LINE [9:8]	
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
HB_LINE [7:0]							

0 0 0 0 0 0 0 0

HB_EN is 0 => Disable horizontal blank function.

HB_EN is 1 => Enable horizontal blank function, PPU will stop at HB_LINE and assert IRQ bit 14

HB_LINE [9:0] => Horizontal blank stop line, PPU will process from start of the frame and stop before this line.

P_PPU_HB_GO 0x930201D0 Horizontal Blank Go Register								
Bit	7	6	5	4	3	2	1	0
Function	HB_GO							
Default	0	0	0	0	0	0	0	0
HB_GO	=> Write 1 to this bit will let the PPU go and process the whole frame.							

The horizontal blank interrupt can be enabled in both line mode and frame base mode. Care must be taken at line base mode since the TV's H blank time is very short and therefore, program must active very fast to this interrupt. The HB_LINE in VGA interlace mode is defined as half of the line. For example, if programmer intends to stop at line 300, 150 must be filled into this register, and line 301 will also stop.

13.16 PPU Bi-linear Interpolation Mode

The PPU embedded in GP12 support bi-linear interpolation for sprites. The bi-linear interpolation is used to improve the picture quality when zoom/rotate/2.5D mode is used. It will not affect the output when simple 2D mode is used. To enable the bi-linear interpolation function, the following registers must be set first.

P_PPU_TEXTN_CTRL 0x9302004C 0x93020064 TEXT N Control Register 0x93020014 0x93020034 (N=1,2,3,4)																																
Bit	23	22	21	20	19	18	17	16																								
Function	-	-	-	-	-	-	-	INTP																								
Default	0	0	0	0	0	0	0	0																								
<table border="1"> <tr> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> </tr> <tr> <td colspan="6">TXN_BLDLVL</td><td>TXN_BLDMODE</td><td>TXN_BLD</td></tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>								15	14	13	12	11	10	9	8	TXN_BLDLVL						TXN_BLDMODE	TXN_BLD	0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8																									
TXN_BLDLVL						TXN_BLDMODE	TXN_BLD																									
0	0	0	0	0	0	0	0																									
<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>TXN_RGBM</td><td>TXN_MODE</td><td>TXN_MVE</td><td>TXN_EN</td><td>TXN_WALL</td><td>TXN_REGM</td><td>TXN_BMP</td><td></td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>								7	6	5	4	3	2	1	0	TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP		0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0																									
TXN_RGBM	TXN_MODE	TXN_MVE	TXN_EN	TXN_WALL	TXN_REGM	TXN_BMP																										
0	0	0	0	0	0	0	0																									

TXN_INTP is 0 => Disable TEXT N's bi-linear interpolation function.

TXN_INTP is 1 => Enable TEXT N's bi-linear interpolation function.

P_Sp_Control 0x93020108 Sprite Control Register								
Bit	23	22	21	20	19	18	17	16
Function	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN	
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Sprite_Number							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN
0	0	0	0	0	0	0	0

SP_INTP is 0 => Disable all sprites's bi-linear interpolation function.

SP_INTP is 1 => Enable all sprites's bi-linear interpolation function.

P_SpN_EXT **0x93026000+(4*N)** **Sprite N Extend Attribute Register**

Bit	23	22	21	20	19	18	17	16
					INTP	LS	GROUP	
-	-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
Y3_FRAC		X3_FRAC		Y2_FRAC		X2_FRAC	
-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
Y1_FRAC		X1_FRAC		Y0_FRAC		X0_FRAC	
-	-	-	-	-	-	-	-

SpN_INTP is useful only when SP_INTP is 1.

SpN_INTP is 0 => Disable sprite N's bi-linear interpolation function.

SpN_INTP is 1 => Enable sprite N's bi-linear interpolation function.

When the interpolation mode is enabled, the bandwidth will become very critical. Since every single pixel needs 4 pixel's data to calculation, so the bandwidth requirement will become 4 times larger than before.

13.17 TV up-scaling Function.

The GPL32900A supports VGA to D1 up-scaling function. This function can also be used on the VGA content and make it become full screen on the LCD TV. The following register is used to control the interpolation mode.

P_PPU_Misc **0x930201F8** **PPU Misc Control Register**

Bit	23	22	21	20	19	18	17	16
Function			TFT_3D					SBMP_MODE
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-		TFTVTQ	-	INTP_MODE	NEW_CMP	TXT_ALPHA	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
	DUAL_BLD			SPR_RGB	TXT_RGB	FB_LOCK	
0	0	0	0	0	0	0	0

When VGA_EN is 0 (QVGA mode)

INTP_MODE is 0 => No interpolation function.

INTP_MODE is 1 => Do QVGA to VGA up-scaling. The TV will be set to VGA interlace mode automatically.

INTP_MODE is 2 => Do QVGA to D1 up-scaling. The TV will be set to D1 interlace mode automatically.

When VGA_EN is 1 (VGA mode)

INTP_MODE is 0 => No interpolation function.

INTP_MODE is 1 => No interpolation function.

INTP_MODE is 2 => Do VGA to D1 up-scaling. The TV will be set to D1 interlace mode automatically.

The TV's output mode will be set automatically, but all other TV parameter still need to be set correctly by software if the INTP mode is change.

When doing the QVGA to VGA/D1 up-scaling, the following equation are used for doing the vertical interpolation.

Field 0 NEW_L0 = 0.75*OLD_L0 + 0.25*OLD_L1

Field 1 NEW_L0 = 0.25*OLD_L0 + 0.75*OLD_L1

Field 2 NEW_L1 = 0.75*OLD_L1 + 0.25*OLD_L2

Field 3 NEW_L1 = 0.25*OLD_L1 + 0.75*OLD_L2

13.18 De-flicker Control

The de-flicker controller built-in GP12 is used to reduce the flicker effect under the VGA interlace output mode. Because of the limit of the TV standard, if the gap difference between even field and odd field is too large under VGA output, the flicker effect will occur. The de-flicker module can reduce the flicker effect by narrowing down the difference between the even and odd filed.

P_PPU_ENABLE	0x930201FC								PPU Control Register			
Bit	31	30	29	28	27	26	25	24	Function	-	DEFEN	TFTLB
Default	0	0	0	0	0	0	0	0				

23	22	21	20	19	18	17	16
-	YUV_TYPE				LB	TFT_SIZE	
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SAVE_ROM	FB_SEL	SPR_WIN	HVCMP_DIS	FB_MONO		SPR25D	FB_FORMAT
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
FB_EN	-	VGA_NOINTL	VGA_EN	TX_BOTUP	TX_DIRECT	CH0_BLK	PPU_EN
0	0	0	0	0	0	0	1

DEFEN is 0 => Disable de-flicker function.

DEFEN is 1 => Enable de-flicker function.

P_DEFICKER_PARA 0x93020370 De-flicker Parameter Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0
PARA3 [3:0]								
7	6	5	4	3	2	1	0	
PARA2 [3:0]								
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:12	-	-	Reserved	-
11:8	PARA3	R/W	De-flicker section 3 threshold level.	-
7:4	PARA2	R/W	De-flicker section 2 threshold level.	-
3:0	PARA1	R/W	De-flicker section 1 threshold level.	-

13.19 PPU Color Mask function.

The GP12 supports a special function calls “color mask”, this function is used to mask some colors of sprite or TEXT. When one of the mask bit is set, the corresponded color of the sprite/TEXT will not be output to display/frame buffer without affect the original color of the background. For example, when programmer choice to mask the Red color of one sprite, then only Green and Blue color will be updated to the display/frame buffer, and the Red part of the output will keep the same without been modified.

To enable this feature, some corresponded register must be set.

P_TxN_Attribute 0x0048 0x0060 0x0010 0x0030 TEXT layer N Attribute Register

Bit	31	30	29	28	27	26	25	24
Function	TXN_CMASK							
Default	0	0	0	0	0	0	0	0
TXN_PB[1:0] TXN_EFF TXN_WINDOW TXN_SIZE[2]								
23	22	21	20	19	18	17	16	
0 0 0 0 0 0 0 0								

15	14	13	12	11	10	9	8
TXN_SIZE[1:0]		TXN_DEPTH		TXN_PALETTE			
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TXN_VS		TXN_HS		TXN_FLIP		TXN_COLOR	
0	0	0	0	0	0	0	0

TXN_CMASK is used to turn on the mask function of each TEXT layer.

TXN_CMASK[2] : 0 => Disable Red color mask function, 1 => Enable Red color mask function.

TXN_CMASK[1] : 0 => Disable Green color mask function, 1 => Enable Green color mask function.

TXN_CMASK[0] : 0 => Disable Blue color mask function, 1 => Enable Blue color mask function.

P_Sp_Control **0x0108** **Sprite Control Register**

Bit	23	22	21	20	19	18	17	16
Function	SP_CMASK	SP_FRAC	SP_GRP	SP_LS	SP_INTP	SP_FAR	CDM	EFFEN
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Sprite_Number							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SP_ZOOMEN	SP_ROTEN	SP_MOSEN	SP_DIRECT	-	SP_BLDMODE	COORD_SEL	SP_EN
0	0	0	0	0	0	0	0

SP_CMASK is 0 => Disable sprite color mask function.

SP_CMASK is 1 => Enable sprite color mask function.

SP_CMASK is the globe control register of sprite color mask function. For each sprite, the following register is used to control the color mask function of each sprite.

P_SpN_EXT **0x6000+(4*N)** **Sprite N Extend Attribute Register**

Bit	23	22	21	20	19	18	17	16
Function	CMASK				INTP	LS	GROUP	
Default	-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
Y3_FRAC		X3_FRAC		Y2_FRAC		X2_FRAC	
-	-	-	-	-	-	-	-

14 OVG INTRODUCTION

14.1 Genearal Description

GPL32900A has a built-in 2D graphpic hardware accelerator, which is compatible with OpenVG 1.1 standard.

15 DISPLAY CONTROLLER

15.1 Introduction

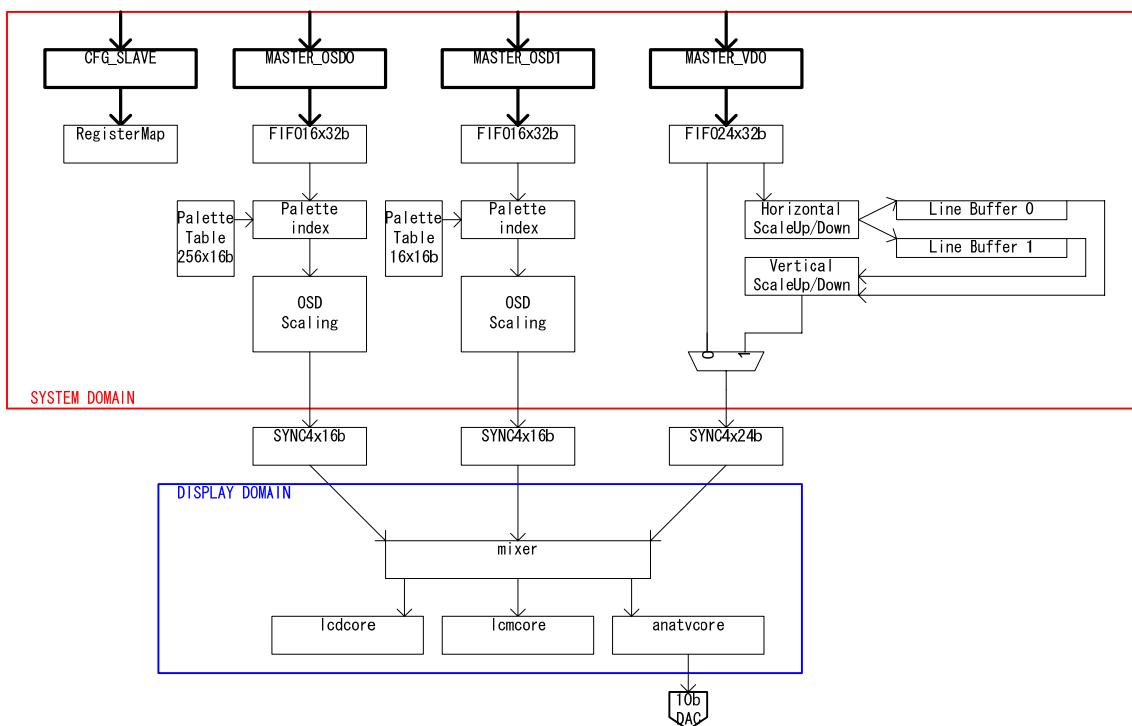
The display controller consists of logic for transferring image data from a frame buffer to an external LCD driver or LCD module. The timing generator can generate the specific timing required for various digital video modes (BT601, BT656, 8080, 6800 ..).

15.2 Features

- Progressive Scan
- Programmable Timing Generator
- Various Output Formats
- 16-235/0-255 Amplitude Selectable
- Selectable sampling time (end or middle of clock period)
- Support 2-Layer On-screen display (OSD) windows
- Each OSD window support RGB565/1555/5515 color mode
- The OSD color mode width is selectable between 1, 4, 8, or 16bits.

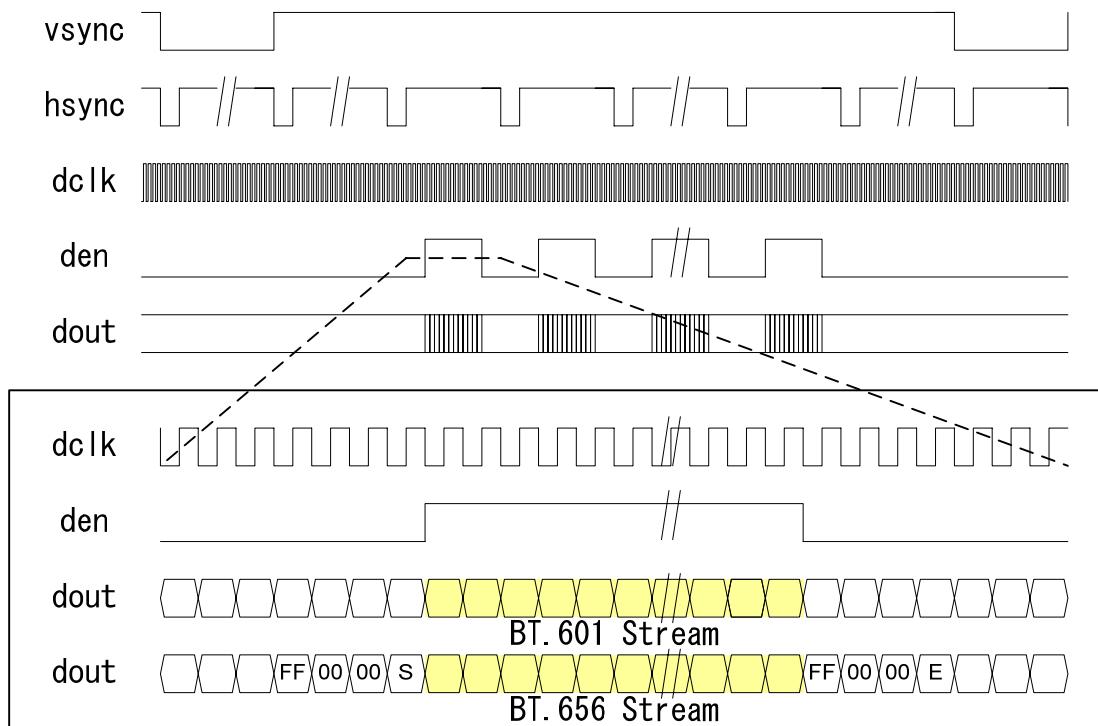
15.3 Block Diagram

The display controller takes the display frame from the external memory and formats it into desired output formats and signals, including data, clock, sync, etc.

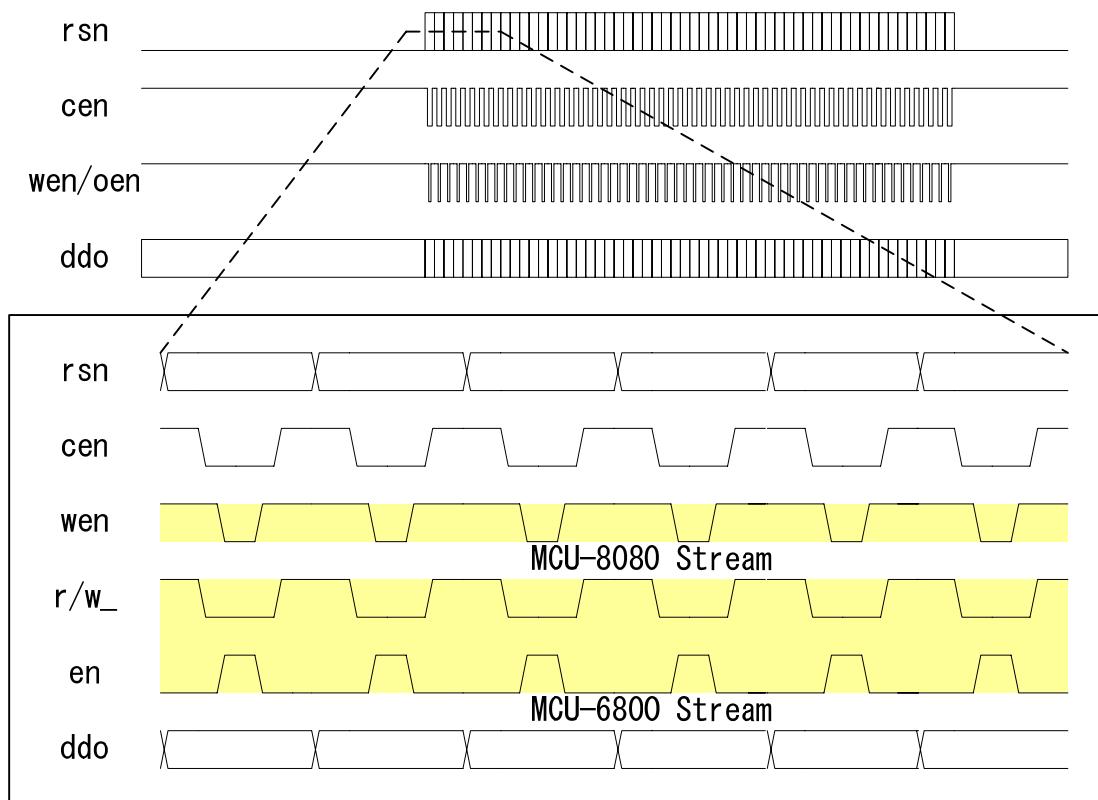


The following timing diagram shows the support of this controller.

LCD Mode



LCM Mode



15.4 Register Summary

Name	Type	Address	Description
P_TFT0_OUTPUT_RES	RW	0x93000000	Output Resolution
P_TFT0_V_BLANK	RW	0x93000004	Vertical Blanking
P_TFT0_H_BLANK	RW	0x93000008	Horizontal Blanking
P_TFT0_BLANK_DATA	RW	0x9300000C	Blanking Pattern
P_TFT0_FRAME_ADR	RW	0x93000010	Frame Buffer Address
P_TFT0_FRAME_PITCH	RW	0x93000014	Frame Buffer Pitch (Byte)
P_TFT0_FRAME_RES	RW	0x93000018	Frame Buffer Resolution
P_TFT0_INT_ENABLE	RW	0x93000080	Interrupt Enable (Mask)
P_TFT0_INT_SOURCE	RW	0x93000084	Interrupt Source
P_TFT0_CBAR_CONTROL	RW	0x93000090	Color bar Control
P_TFT0_CBAR_COLOR	RW	0x93000094	Color bar Single Color
P_TFT0_FLIP_ENABLE	RW	0x93000098	Flip function Enable
P_TFT0_CM_PARAM0	RW	0x930000A0	Color Matrix Parameter 0
P_TFT0_CM_PARAM1	RW	0x930000A4	Color Matrix Parameter 1
P_TFT0_CM_PARAM2	RW	0x930000A8	Color Matrix Parameter 2
P_TFT0_CM_PARAM3	RW	0x930000AC	Color Matrix Parameter 3
P_TFT0_CM_PARAM4	RW	0x930000B0	Color Matrix Parameter 4
P_TFT0_CM_PARAM5	RW	0x930000B4	Color Matrix Parameter 5
P_TFT0_DITHER_MAP0	RW	0x930000B8	Dithering Map 0
P_TFT0_DITHER_MAP1	RW	0x930000BC	Dithering Map 1
P_TFT0_LCM_AC_TIMING	RW	0x930000E0	LCM AC Timing
P_TFT0_LCD_VSYNC	RW	0x930000E4	LCD Vsync Timing
P_TFT0_LCD_HSYNC	RW	0x930000E8	LCD Hsync Timing
P_TFT0_DISP_FORMAT	RW	0x930000F0	Display Format
P_TFT0_DISP_CONTROL	RW	0x930000FC	Display Control
P_TFT0_GAMMA_PTR	RW	0x93000400	Gamma Pointer
P_TFT0_LCM_PROGRAM_IO	RW	0x93000F00	LCM Programming In-out

15.5 Register Definition

P_TFT0_OUTPUT_RES		0x93000000								Output Resolution															
Bit	31	30	29	28	27	26	25	24	RES_WIDTH																
Function	-	-	-	-	RES_WIDTH																				
Default	0	0	0	0	0																				
RES_WIDTH																									
0																									

Bit	15	14	13	12	11	10	9	8	
Function	-	-	-	-	RES_HEIGHT				

Default 0 0 0 0 0

7	6	5	4	3	2	1	0
RES_HEIGHT							

0

Bit	Function	Type	Description					Condition
[31:28]	-	-	Reserved					-
[27:16]	RES_WIDTH	RW	The width of output screen.					-
[15:12]	-	-	Reserved					-
[11:0]	RES_HEIGHT	RW	The height of output screen.					-

LIMIT:

** RES_WIDTH = BLANK_LEFT + BLANK_RIGHT + ACT_WIDTH (or SCL_WIDTH)

** RES_HEIGHT = BLANK_TOP + BLANK_BOTTOM + ACT_HEIGHT (or SCL_HEIGHT)

P_TFT0_V_BLANK

0x93000004

Vertical Blanking

Bit	31	30	29	28	27	26	25	24	
Function	-	-	-	-	BLANK_TOP				

Default 0 0 0 0 0

23	22	21	20	19	18	17	16
BLANK_TOP							

0

Bit	15	14	13	12	11	10	9	8	
Function	-	-	-	-	BLANK_BOTTOM				

Default 0 0 0 0 0

7	6	5	4	3	2	1	0
BLANK_BOTTOM							

0

Bit	Function	Type	Description					Condition
[31:28]	-	-	Reserved					-
[27:16]	BLANK_TOP	RW	Blanking from the top of the screen.					-
[15:12]	-	-	Reserved					-

Bit	Function	Type	Description					Condition
[11:0]	BLANK_BOTTOM	RW	Blanking from the bottom of the screen.					-

P_TFT0_H_BLANK
0x93000008
Horizontal Blanking

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	BLANK_LEFT
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
BLANK_LEFT							
0							

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	BLANK_RIGHT
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
BLANK_RIGHT							
0							

Bit	Function	Type	Description					Condition
[31:28]	-	-	Reserved					-
[27:16]	BLANK_LEFT	RW	Blanking from the left of the screen.					-
[15:12]	-	-	Reserved					-
[11:0]	BLANK_RIGHT	RW	Blanking from the right of the screen.					-

P_TFT0_BLANK_DATA
0x9300000C
Blanking Pattern

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
BLANK_DATA_HI							
0							

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	BLANK_DATA_MI
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
BLANK_DATA_LO							
0							

Bit	Function	Type	Description	Condition
[31:24]	-	-	Reserved	-
[23:16]	BLANK_DATA_HI	RW	Blanking data of R/Cr.	-
[15:8]	BLANK_DATA_MI	RW	Blanking data of R/Cb.	-
[7:0]	BLANK_DATA_LO	RW	Blanking data of R/Y.	-

P_TFT0_FRAME_ADR 0x93000010 Frame Buffer Address							
Bit	31	30	29	28	27	26	25
FRAME_BUF_ADR							
Default	0						

23	22	21	20	19	18	17	16
FRAME_BUF_ADR							
0							

Bit	15	14	13	12	11	10	9	8
FRAME_BUF_ADR								
Default	0							

7	6	5	4	3	2	1	0
FRAME_BUF_ADR							
0							

Bit	Function	Type	Description	Condition
[31:0]	FRAME_BUF_ADR	RW	Frame buffer address.	-

P_TFT0_FRAME_PITCH 0x93000014 Frame Buffer Pitch (Byte)							
Bit	31	30	29	28	27	26	25
SRC_PITCH							
Default	0						

23	22	21	20	19	18	17	16
SRC_PITCH							
0							

Bit	15	14	13	12	11	10	9	8
Function	ACT_PITCH							
Default	0							
	7	6	5	4	3	2	1	0
	ACT_PITCH							
	0							

Bit	Function	Type	Description					Condition
[31:16]	SRC_PITCH	RW	Source frame pitch.					-
[15:0]	ACT_PITCH	RW	Active frame pitch.					-

LIMIT:

** PITCH = WIDTH * BYTE_PER_PIXEL

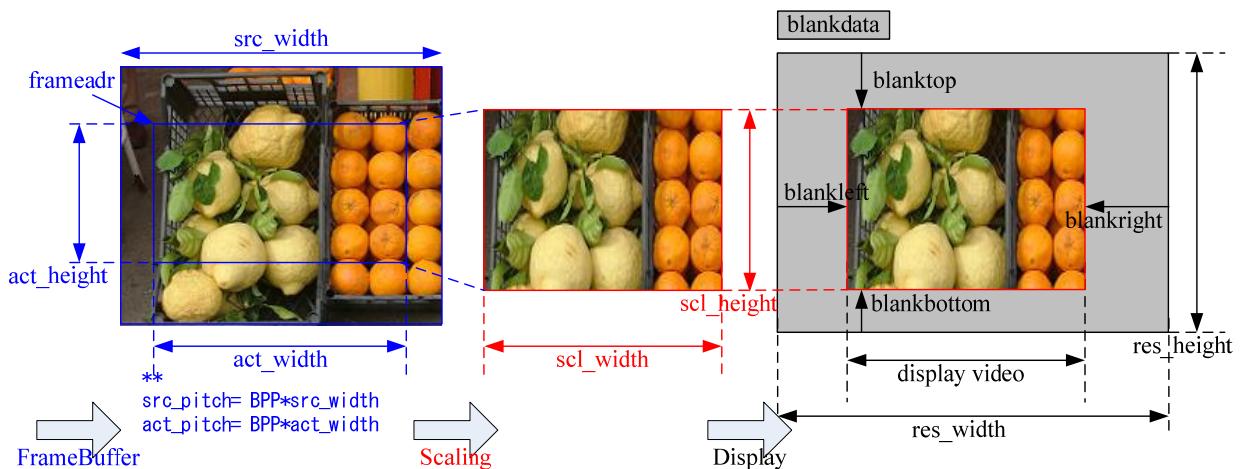
** PITCH % 4 == 0

P_TFT0_FRAME_RES 0x93000018 Frame Buffer Resolution

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	ACT_WIDTH			
Default	0	0	0	0	0			
	23	22	21	20	19	18	17	16
	ACT_WIDTH							
	0							

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	ACT_HEIGHT			
Default	0	0	0	0	0			
	7	6	5	4	3	2	1	0
	ACT_HEIGHT							
	0							

Bit	Function	Type	Description					Condition
[31:28]	-	-	Reserved					-
[27:16]	ACT_WIDTH	RW	Active frame width.					-
[15:12]	-	-	Reserved					-
[11:0]	ACT_HEIGHT	RW	Active frame height.					-


P_TFT0_INT_ENABLE 0x93000080

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	-	-	UPDATE_FAIL_INT_EN	FIELD_END_INT_EN	FRAME_END_INT_EN	UPDATE_PARAM_INT_EN	DISPLAY_OFF_INT_EN
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:5]	-	-	Reserved	-
[4]	UPDATE_FAIL_INT_EN	RW	MMRs update failed Interrupt enable	-
[3]	FIELD_END_INT_EN	RW	Each field end Interrupt enable	-
[2]	FRAME_END_INT_EN	RW	Each frame end Interrupt enable	-
[1]	UPDATE_PARAM_INT_EN	RW	Update all the modified registers Interrupt enable	-
[0]	DISPLAY_OFF_INT_EN	RW	Controller is disabled Interrupt enable	-

P_TFT0_INT_SOURCE
0x93000084
Interrupt Source

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	-	-	-	OSD_BANK	GAMMA_BANK			
	0	0	0	0		0		
Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	-	-	-	UPDATE_FAIL	FIELD_END	FRAME_END	UPDATE_PARAM	DISPLAY_OFF
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:21]	-	-	Reserved	-
[20]	OSD_BANK	RW	Palette pointer	0: OSD0 1: OSD1
[19:18]	-	-	Reserved	-
[17:16]	GAMMA_BANK	RW	Gamma table	0: R 1: G 2: B
[15:5]	-	-	Reserved	-
[4]	UPDATE_FAIL	RWC	MMRs update failed	-
[3]	FIELD_END	RWC	Each field end	-
[2]	FRAME_END	RWC	Each frame end	-
[1]	UPDATE_PARAM	RWC	Update all the modified registers	-
[0]	DISPLAY_OFF	RWC	Controller is disabled	-

NOTE:

"Frame End" Interrupt will be sent only when bit [2] of P_TFT0_INT_ENABLE and

P_TFT0_INT_SOURCE are both set to 1.

P_TFT0_CBAR_CONTROL 0x93000090
Color-bar Control

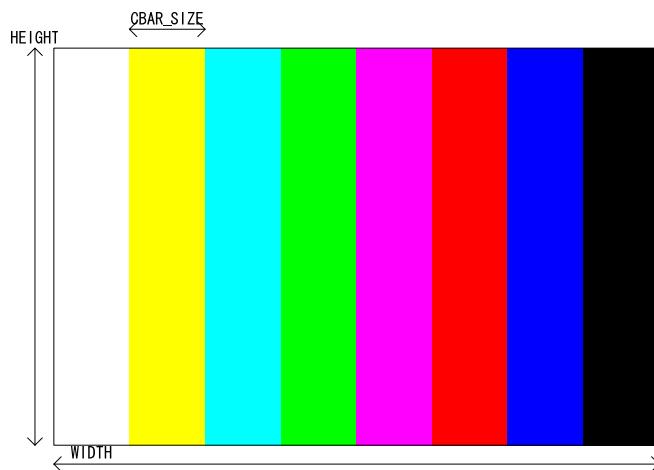
Bit	31	30	29	28	27	26	25	24
Function	CBAR_GEN	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	CBAR_TYPE			
Default	0	0	0	0			0	

7	6	5	4	3	2	1	0
CBAR_SIZE							
0							

Bit	Function	Type	Description	Condition
[31]	CBAR_GEN	RW	Interrupt Enable	-
[30:12]	-	-	Reserved	-
[11:8]	CBAR_TYPE	RW	Color bar type	0: Single color 1: 75/0/75/0 (gamma=1/0.45) 2: 75/0/75/0 (RGB) 3: 75/0/75/0 (YCbCr) 4: 100/0/75/0 (gamma=1/0.45) 5: 100/0/75/0 (RGB) 6: 100/0/75/0 (YCbCr) 7: 100/0/100/0 (RGB) 8: 100/0/100/0 (YCbCr) 9: 10step staircase (RGB) A: 10step staircase (YCbCr)
[7:0]	CBAR_SIZE	RW	Column per bar	


P_TFT0_CBAR_COLOR 0x93000094
Color-bar single color

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
COLORHI							

0

Bit	15	14	13	12	11	10	9	8
Function	COLORMI							

Default

0

7	6	5	4	3	2	1	0
COLORLO							

0

Bit	Function	Type	Description				Condition
[31:24]	-	-	Reserved				-
[23:16]	COLORHI	RW	Single color R/Cr data				-
[15:8]	COLORMI	RW	Single color G/Cb data				-
[7:0]	COLORLO	RW	Single color B/Y data				-

P_TFT0_FLIP_ENABLE
0x93000098
Flip function control

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	-	-	OSD_SCALE_1	OSD_SCALE_0	OSD_1_EN	OSD_0_EN	DISP_FLIP_EN
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:5]	-	-	Reserved	-
[4]	OSD_SCALE_1	RW	Flip function control of OSD1 (Using after scaling effect)	0: Disable 1: Enable
[3]	OSD_SCALE_0	RW	Flip function control of OSD0 (Using after scaling effect)	0: Disable 1: Enable
[2]	OSD_1_EN	RW	Flip function control of OSD1 (Using before scaling effect)	0: Disable 1: Enable
[1]	OSD_0_EN	RW	Flip function control of OSD0 (Using before scaling effect)	0: Disable 1: Enable
[0]	DISP_FLIP_EN	RW	Flip function control of display unit	0: Disable 1: Enable

P_TFT0_CM_PARAM0	0x930000A0	Color Matrix Parameter 0						
Bit	31	30	29	28	27	26	25	24
Function	CMTX_a01							

Default 0

23	22	21	20	19	18	17	16
CMTX_a01							
0							

Bit	15	14	13	12	11	10	9	8
Function	CMTX_a00							
Default	0							

7	6	5	4	3	2	1	0
CMTX_a00							
0							

Bit	Function	Type	Description	Condition
[31:16]	CMTX_a01	RW	Color Matrix Parameter a01	-
[15:0]	CMTX_a00	RW	Color Matrix Parameter a00	-

P_TFT0_CM_PARAM1 0x930000A4 Color Matrix Parameter 1

Bit	31	30	29	28	27	26	25	24
CMTX_a10								

Default 0

23	22	21	20	19	18	17	16
CMTX_a10							
0							

Bit	15	14	13	12	11	10	9	8
CMTX_a02								

Default 0

7	6	5	4	3	2	1	0
CMTX_a02							
0							

Bit	Function	Type	Description	Condition
[31:16]	CMTX_a10	RW	Color Matrix Parameter a10	-
[15:0]	CMTX_a02	RW	Color Matrix Parameter a02	-

P_TFT0_CM_PARAM2 0x930000A8 Color Matrix Parameter 2

Bit	31	30	29	28	27	26	25	24
CMTX_a12								

Default 0

23	22	21	20	19	18	17	16
CMTX_a12							
0							

Bit	15	14	13	12	11	10	9	8
Function	CMTX_a11							
Default	0							
	7	6	5	4	3	2	1	0
	CMTX_a11							
	0							
Bit	Function	Type	Description				Condition	
[31:16]	CMTX_a12	RW	Color Matrix Parameter a12				-	
[15:0]	CMTX_a11	RW	Color Matrix Parameter a11				-	

P_TFT0_CM_PARAM3 0x930000AC Color Matrix Parameter 3								
Bit	31	30	29	28	27	26	25	24
Function	CMTX_a21							
Default	0							
	23	22	21	20	19	18	17	16
	CMTX_a21							
	0							
Bit	15	14	13	12	11	10	9	8
Function	CMTX_a20							
Default	0							
	7	6	5	4	3	2	1	0
	CMTX_a20							
	0							

Bit	Function	Type	Description				Condition	
[31:16]	CMTX_a21	RW	Color Matrix Parameter a21				-	
[15:0]	CMTX_a20	RW	Color Matrix Parameter a20				-	

P_TFT0_CM_PARAM4 0x930000B0 Color Matrix Parameter 4								
Bit	31	30	29	28	27	26	25	24
Function	CMTX_b0							
Default	0							
	31	30	29	28	27	26	25	24
	CMTX_b0							
	0							

23	22	21	20	19	18	17	16
CMTX_b0							

0

Bit	15	14	13	12	11	10	9	8
Function	CMTX_a22							
Default	0							

7	6	5	4	3	2	1	0
CMTX_a22							

0

Bit	Function	Type	Description				Condition
[31:16]	CMTX_b0	RW	Color Matrix Parameter b0				-
[15:0]	CMTX_a22	RW	Color Matrix Parameter a22				-

P_TFT0_CM_PARAM5 0x930000B4 Color Matrix Parameter 5

Bit	31	30	29	28	27	26	25	24
Function	CMTX_b2							
Default	0							

23	22	21	20	19	18	17	16
CMTX_b2							

0

Bit	15	14	13	12	11	10	9	8
Function	CMTX_b1							

Default 0

7	6	5	4	3	2	1	0
CMTX_b1							

0

Bit	Function	Type	Description				Condition
[31:16]	CMTX_b2	RW	Color Matrix Parameter b2				-
[15:0]	CMTX_b1	RW	Color Matrix Parameter b1				-

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

NOTICE:

If the LCD output is YUV or YCbCr color space, color matrix is necessary for transforming from RGB to YUV/YCbCr. Notice that P0, P1, P2 means V/Cr, U/Cb, and Y respectively.

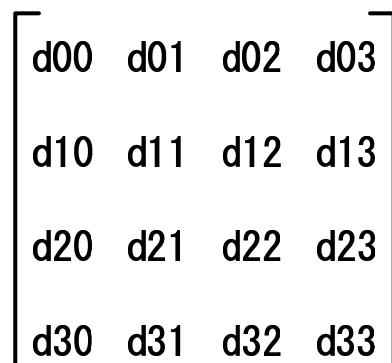
P_TFT0_DITHER_MAP0 0x930000B8 Dithering Map 0									
Bit	31	30	29	28	27	26	25	24	
Function	d00					d01			
Default	0					0			
	23	22	21	20	19	18	17	16	
	d02					d03			
	0					0			
Bit	15	14	13	12	11	10	9	8	
Function	d10					d11			
Default	0					0			
	7	6	5	4	3	2	1	0	
	d12					d13			
	0					0			

Bit	Function	Type	Description	Condition
[31:28]	d00	RW	Dithering Map parameter d00	-
[27:24]	d01	RW	Dithering Map parameter d01	-
[23:20]	d02	RW	Dithering Map parameter d02	-
[19:16]	d03	RW	Dithering Map parameter d03	-
[15:12]	d10	RW	Dithering Map parameter d10	-
[11:8]	d11	RW	Dithering Map parameter d11	-
[7:4]	d12	RW	Dithering Map parameter d12	-
[3:0]	d13	RW	Dithering Map parameter d13	-

P_TFT0_DITHER_MAP1
0x930000BC
Dithering Map 1

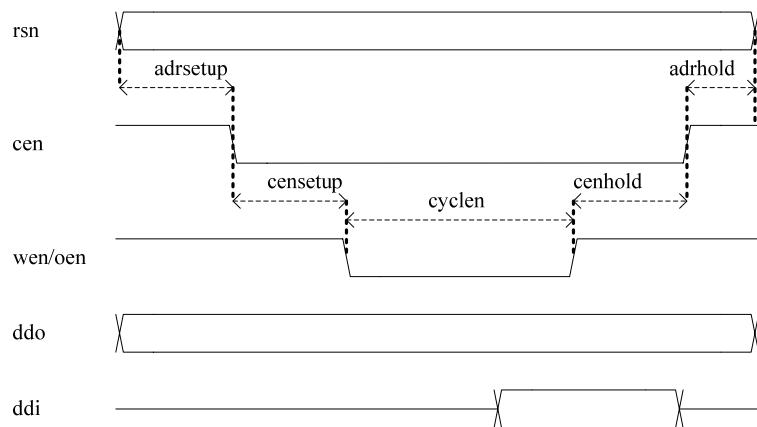
Bit	31	30	29	28	27	26	25	24	
Function	d20					d21			
Default	0					0			
Bit	23	22	21	20	19	18	17	16	
Function	d22					d23			
Default	0					0			
Bit	15	14	13	12	11	10	9	8	
Function	d30					d31			
Default	0					0			
Bit	7	6	5	4	3	2	1	0	
Function	d32					d33			
Default	0					0			

Bit	Function	Type	Description	Condition
[31:28]	d20	RW	Dithering Map parameter d20	-
[27:24]	d21	RW	Dithering Map parameter d21	-
[23:20]	d22	RW	Dithering Map parameter d22	-
[19:16]	d23	RW	Dithering Map parameter d23	-
[15:12]	d30	RW	Dithering Map parameter d30	-
[11:8]	d31	RW	Dithering Map parameter d31	-
[7:4]	d32	RW	Dithering Map parameter d32	-
[3:0]	d33	RW	Dithering Map parameter d33	-

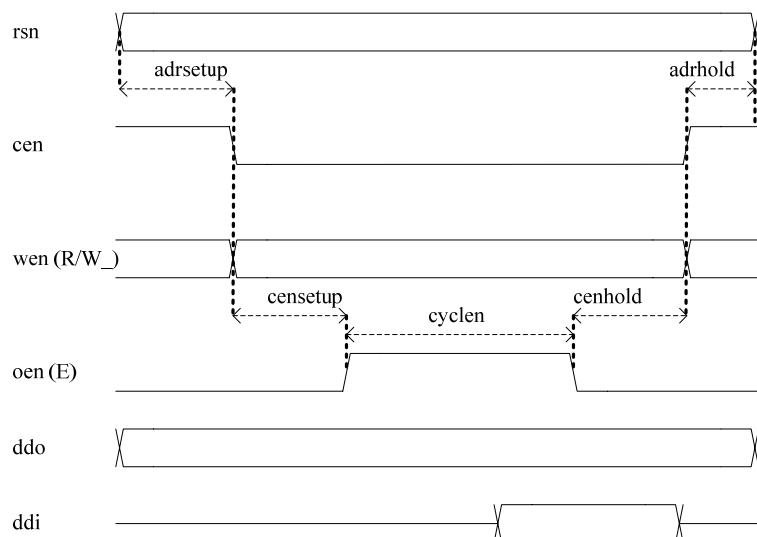


P_TFT0_LCM_AC_TIMING 0x930000E0								LCM AC Timing								
Bit	31	30	29	28	27	26	25	24								
Function	ADRSETUP				ADRHOLD											
Default	0								0							
	23	22	21	20	19	18	17	16								
	ADRHOLD								0							
Bit	15	14	13	12	11	10	9	8								
Function	CENSETUP				CENHOLD											
Default	0								0							
	7	6	5	4	3	2	1	0								
	CYCLEN								0							
Bit	Function		Type		Description								Condition			
[31:28]	ADRSETUP		RW		Address Setup Time								-			
[27:16]	ADRHOLD		RW		Address Hold Time								-			
[15:12]	CENSETUP		RW		Chip Select Setup Time								-			
[11:8]	CENHOLD		RW		Chip Select Hold Time								-			
[7:0]	CYCLEN		RW		Strobe Length								-			

MCU-8080 :



MCU-6800 :



P TFT0 LCD VSYNC

0x930000E4

LCD Vsync Timing

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	VS_POLAR	VS_FRONT			
Default	0	0	0	0	0			

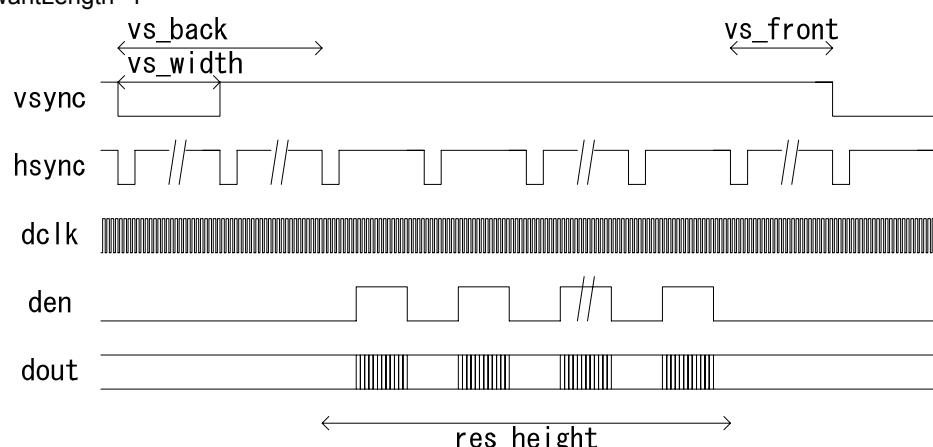
23	22	21	20	19	18	17	16
VS FRONT						VS BACK	

7	6	5	4	3	2	1	0
VS_WIDTH							
0							

Bit	Function	Type	Description	Condition
[31:29]	-	-	Reserved	-
[28]	VS_POLAR	RW	Vertical Sync Polarity	-
[27:18]	VS_FRONT	RW	Vertical Sync Front Porch	-
[17:8]	VS_BACK	RW	Vertical Sync Back Porch	-
[7:0]	VS_WIDTH	RW	Vertical Sync Width	-

LIMIT:

Register = WantLength -1



P_TFT0_LCD_HSYNC 0x930000E8

LCD Hsync Timing

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	HS_POLAR	HS_FRONT			
Default	0	0	0	0				

23	22	21	20	19	18	17	16
HS_FRONT						HS_BACK	
						0	

Bit	15	14	13	12	11	10	9	8
Function	HS_BACK							
Default	0							

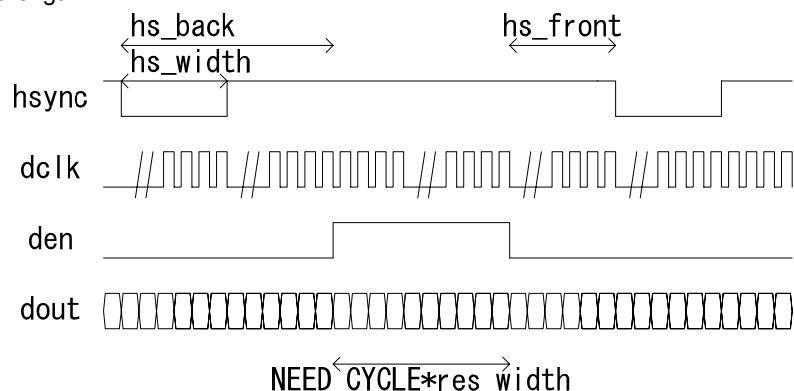
7	6	5	4	3	2	1	0
HS_WIDTH							
0							

Bit	Function	Type	Description	Condition
[31:29]	-	-	Reserved	-
[28]	HS_POLAR	RW	Horizontal Sync Polarity	-
[27:18]	HS_FRONT	RW	Horizontal Sync Front Porch	-
[17:8]	HS_BACK	RW	Horizontal Sync Back Porch	-
[7:0]	HS_WIDTH	RW	Horizontal Sync Width	-

LIMIT:

WantLength % CLKDIV == 0

Register = WantLength -1



P_TFT0_DISP_FORMAT 0x930000F0 Display Format							
Bit	31	30	29	28	27	26	25
Function	-	-	OFORMAT			-	-
Default	0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	OSEQ0				-	OSEQ1	
Default	0	0				0	0	

Bit	15	14	13	12	11	10	9	8
Function	DITHEREN	-	DITHSEL		LCM_OSEQSEL	-	GAMMAEN	-
Default	0	0	0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	AMPSEL	IFORMAT	ITYPE	
Default	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:30]	-	-	Reserved	-

Bit	Function	Type	Description	Condition																																																	
[29:28]	OFORMAT	RW	Output Format	<p>LCD Output Format:</p> <table border="1"> <tr><td>0</td><td>RGB Color Space</td></tr> <tr><td>1</td><td>YCbCr Color Space</td></tr> <tr><td>2</td><td>YUV Color Space</td></tr> </table> <p>LCM Output Format:</p> <table border="1"> <tr><td>0</td><td>RGB Color Space</td></tr> </table>	0	RGB Color Space	1	YCbCr Color Space	2	YUV Color Space	0	RGB Color Space																																									
0	RGB Color Space																																																				
1	YCbCr Color Space																																																				
2	YUV Color Space																																																				
0	RGB Color Space																																																				
[27:26]	-	-	Reserved	-																																																	
[25:24]	OTYPE	RW	Output Type	<p>LCD Output Type:</p> <table border="1"> <tr><td></td><td>RGB</td><td>YCbCr</td><td>YUV</td></tr> <tr><td>0</td><td>pRGB888</td><td>YCbCr24</td><td>YUV24</td></tr> <tr><td>1</td><td>pRGB565</td><td>YCbCr16</td><td>YUV16</td></tr> <tr><td>2</td><td>sRGB888</td><td>YCbCr8</td><td>YUV8</td></tr> <tr><td>3</td><td>sRGBM8 88</td><td></td><td></td></tr> </table> <p>LCM Output Type:</p> <table border="1"> <tr><td>0</td><td>RGB666</td></tr> <tr><td>1</td><td>RGB565</td></tr> <tr><td>2</td><td>RGB444</td></tr> <tr><td>3</td><td>RGB332</td></tr> </table>		RGB	YCbCr	YUV	0	pRGB888	YCbCr24	YUV24	1	pRGB565	YCbCr16	YUV16	2	sRGB888	YCbCr8	YUV8	3	sRGBM8 88			0	RGB666	1	RGB565	2	RGB444	3	RGB332																					
	RGB	YCbCr	YUV																																																		
0	pRGB888	YCbCr24	YUV24																																																		
1	pRGB565	YCbCr16	YUV16																																																		
2	sRGB888	YCbCr8	YUV8																																																		
3	sRGBM8 88																																																				
0	RGB666																																																				
1	RGB565																																																				
2	RGB444																																																				
3	RGB332																																																				
[23]	-	-	Reserved	-																																																	
[22:20]	OSEQ0	RW	Output Sequence for Even Line	<p>LCD Output Sequence:</p> <table border="1"> <tr><td rowspan="2">pRGB888</td><td>0</td><td>RGB</td></tr> <tr><td>1</td><td>BGR</td></tr> <tr><td rowspan="2">pRGB565</td><td>0</td><td>RGB</td></tr> <tr><td>1</td><td>BGR</td></tr> <tr><td rowspan="6">sRGB888</td><td>0</td><td>R -> G -> B</td></tr> <tr><td>1</td><td>G -> B -> R</td></tr> <tr><td>2</td><td>B -> R -> G</td></tr> <tr><td>3</td><td>R -> B -> G</td></tr> <tr><td>4</td><td>B -> G -> R</td></tr> <tr><td>5</td><td>G -> R -> B</td></tr> <tr><td rowspan="6">sRGBM888</td><td>0</td><td>R -> G -> B -> M</td></tr> <tr><td>1</td><td>G -> B -> R -> M</td></tr> <tr><td>2</td><td>B -> R -> G -> M</td></tr> <tr><td>3</td><td>R -> B -> G -> M</td></tr> <tr><td>4</td><td>B -> G -> R -> M</td></tr> <tr><td>5</td><td>G -> R -> B -> M</td></tr> <tr><td rowspan="6">YCbCr24</td><td>0</td><td>YCbCr</td></tr> <tr><td>1</td><td>YcrCb</td></tr> <tr><td>2</td><td>CbYCr</td></tr> <tr><td>3</td><td>CrYCb</td></tr> <tr><td>4</td><td>CbCrY</td></tr> <tr><td>5</td><td>CrCbY</td></tr> </table>	pRGB888	0	RGB	1	BGR	pRGB565	0	RGB	1	BGR	sRGB888	0	R -> G -> B	1	G -> B -> R	2	B -> R -> G	3	R -> B -> G	4	B -> G -> R	5	G -> R -> B	sRGBM888	0	R -> G -> B -> M	1	G -> B -> R -> M	2	B -> R -> G -> M	3	R -> B -> G -> M	4	B -> G -> R -> M	5	G -> R -> B -> M	YCbCr24	0	YCbCr	1	YcrCb	2	CbYCr	3	CrYCb	4	CbCrY	5	CrCbY
pRGB888	0	RGB																																																			
	1	BGR																																																			
pRGB565	0	RGB																																																			
	1	BGR																																																			
sRGB888	0	R -> G -> B																																																			
	1	G -> B -> R																																																			
	2	B -> R -> G																																																			
	3	R -> B -> G																																																			
	4	B -> G -> R																																																			
	5	G -> R -> B																																																			
sRGBM888	0	R -> G -> B -> M																																																			
	1	G -> B -> R -> M																																																			
	2	B -> R -> G -> M																																																			
	3	R -> B -> G -> M																																																			
	4	B -> G -> R -> M																																																			
	5	G -> R -> B -> M																																																			
YCbCr24	0	YCbCr																																																			
	1	YcrCb																																																			
	2	CbYCr																																																			
	3	CrYCb																																																			
	4	CbCrY																																																			
	5	CrCbY																																																			

Bit	Function	Type	Description	Condition		
[22:20]	OSEQ0	RW	Output Sequence for Even Line	YCbCr16	0	Y0Cb -> Y1Cr
					1	Y0Cr -> Y1Cb
					2	CbY0 -> CrY1
					3	CrY0 -> CbY1
				YCbCr8	0	Y0 -> Cb -> Y1 -> Cr
					1	Y0 -> Cr -> Y1 -> Cb
					2	Cb -> Y0 -> Cr -> Y1
					3	Cr -> Y0 -> Cb -> Y1
				YUV24	0	YUV
					1	YVU
					2	UYV
					3	VYU
					4	UVY
					5	VUY
				YUV16	0	Y0U -> Y1V
					1	Y0V -> Y1U
					2	UY0 -> VY1
					3	VY0 -> UY1
				YUV8	0	Y0 -> U -> Y1 -> V
					1	Y0 -> V -> Y1 -> U
					2	U -> Y0 -> V -> Y1
					3	V -> Y0 -> U -> Y1
				LCM Output Sequence:		
				See the appendix for more details		
[19]	-	-	Reserved	-		
[18:16]	OSEQ1	RW	Output Sequence for Odd Line	Table is the same as OSEQ0 (for LCD Mode only).		
[15]	DITHEREN	RW	Dithering Enable	0: Disable 1: Enable		
[14]	-	-	Reserved	-		
[13:12]	DITHSEL	RW	Dithering Selection	0	Ordered Dithering (Fixed)	
				1	Ordered Dithering (Wheel)	
				2	HorizontalErrorDiffusion	

Bit	Function	Type	Description	Condition	
[11]	LCM_OSEQSEL	RW	LCM Output Sequence Selection	See the appendix for more details	
[10]	-	-	Reserved	-	
[9]	GAMMAEN	RW	Gamma Enable	0: Disable 1: Enable	
[8:4]	-	-	Reserved	-	
[3]	AMPSEL	RW	Amplitude Selection	0	0-255 RGB <-> 0-255 YCbCr
				1	0-255 RGB <-> 16-235 YCbCr
[2]	IFORMAT	RW	Input Format	0: RGB color space 1: YCbCr color space	
[1:0]	ITYPE	RW	Input Type	0	RBG
				0	YCbCr
				1	RGB565
				2	4Y4Cb4Y4Cr
				3	RGB888
					YCbCr

P_TFT0_DISP_CONTROL 0x930000FC
Display Control

Bit	31	30	29	28	27	26	25	24
Function	DISTYPE	-	DMATYPE	-	-	-	BTYPE	
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
CLKPOLAR	-	CLKSEL	-	OBT656EN	-	OBLANK0	
0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	LCMDATATYPE	LCMMODE	LCMDATASEL	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0

Bit	Function	Type	Description			Condition	
[31:30]	DISTYPE	RW	Display device type		0	Disable	
					1	LCM DMA	
					2	LCD Enable	
					3	TV Enable	
[29]	-	-	Reserved			-	

Bit	Function	Type	Description	Condition	
[28]	DMATYPE	RW	DMA type	0	Progressive DMA
				1	Interlaced DMA
[27:26]	-	-	Reserved	-	
[25:24]	BTYPE	RW	Burst type	0	SINGLE
				1	INCR4
				2	INCR8
				3	INCR16
[23]	CLKPOLAR	RW	LCD Sample Clock Polarity	-	
[22]	-	-	Reserved	-	
[21:20]	CLKSEL	RW	Clock Selection	lcd_dclk = clk/CLKDIV	
					CLKDIV
				0	1
				1	2
				2	4
				3	8
[19]	-	-	Reserved	-	
[18]	OBT656EN	RW	BT656 mode Enable (Insert SAV,EAV into video stream)	0: Disable	1: Enable
[17]	-	-	Reserved	-	
[16]	OBLANK0	RW	Fill 0 during blanking interval.	-	
[15]	-	-	Reserved	-	
[14]	LCMDATATYPE	RW	LCM mode data type	0	LCM 16bit Interface
				1	LCM 8bit Interface
[13]	LCMMODE	RW	LCM 8080/6800 mode selection	0	8080 Mode
				1	6800 Mode
[12]	LCMDATASEL	RW	LCM data selection	0	Data Select is logic-0
				1	Data Select is logic-1
[11:0]	-	-	Reserved	-	

- **(0x93000F00) LCM Programming In-out (16-bit)**

haddr				
Bit	Name	Type	Description	
11:8	RegMapOffset	RW	LCM ProgramIO Mode	1
7		RW		0
6		RW		0
5	RegisterSelect1	RW	RegisterSelect for Data1	X
4	RegisterSelect0	RW	RegisterSelect for Data0	X
3:0		RW	0	0
hwdata/hrdata				
Bit	Name	Type	Description	Reset Value
31:16	Data1	RW	Data channel1	0
15:0	Data0	RW	Data channel0	0

- **(0x93000F00) LCM Programming In-out (8-bit)**

haddr				
Bit	Name	Type	Description	
11:8	RegMapOffset	RW	LCM ProgramIO Mode	1
7	RegisterSelect3	RW	RegisterSelect for Data3	X
6	RegisterSelect2	RW	RegisterSelect for Data2	X
5	RegisterSelect1	RW	RegisterSelect for Data1	X
4	RegisterSelect0	RW	RegisterSelect for Data0	X
3:0		RW	0	0

hwdata/hrdata				
Bit	Name	Type	Description	Reset Value
31:24	Data3	RW	Data channel3	0
23:16	Data2	RW	Data channel2	0
15:8	Data1	RW	Data channel1	0
7:0	Data0	RW	Data channel0	0

Programming example (Based on 8-bit):

```
#define LCM_write_CMD8(addr)      HAL_WRITE_UINT8(0x93000F00, (INT8U)addr);
```

For giving command, writing address into 0x93000F00. (bit[7:4] = 0 means this is for address writing)

```
#define LCM_write_DATA8(data)      HAL_WRITE_UINT8(0x93000Fx0, (INT8U)data);
```

For giving data, writing data into 0x93000Fx0. (bit[7:4] != 0 means this is for data writing)

- ◆ 0x93000F10 : write data into bit [7:0].
- ◆ 0x93000F20 : write data into bit [15:8].
- ◆ 0x93000F40 : write data into bit [23:16].
- ◆ 0x93000F80 : write data into bit [31:24].
- ◆ 0x93000F30 : write data into bit [15:0].
- ◆ 0x93000FC0 : write data into bit [31:16].
- ◆ There are more combinations...

15.6 APPENDIX

iNPUT FORMATS

Input Format

OUTPUT FORMATS

OTYPE	TIME	[23:16]								[15:8]								[7:0]							
YUV16	n									Y	Y	Y	Y	Y	Y	Y	Y	U	U	U	U	U	U	U	U
	n+1									Y	Y	Y	Y	Y	Y	Y	Y	V	V	V	V	V	V	V	V
YUV8	n									Y	Y	Y	Y	Y	Y	Y	Y								
	n+1									U	U	U	U	U	U	U	U								
	n+2									Y	Y	Y	Y	Y	Y	Y	Y								
	n+3									V	V	V	V	V	V	V	V								

LCD Output Format @ SeqN = 0

OTYPE	OSEQ0	TIME	[15:8]												[7:0]											
RGB666	0	n																							R	R
		n+1	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B	B	B	B	
	1	n																							B	B
		n+1	B	B	B	B	G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R	R	R	R	
	2	n	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B	B	B	B	
		n+1																						R	R	
	3	n	B	B	B	B	G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R	R	R	R	
		n+1																						B	B	
	4	n	R	R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B	B	B	B	B	B	B	
		n+1	B	B																						
	5	n	B	B	B	B	B	B	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R	R	R	
		n+1	R	R																						
	6	n					G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B	B	B	B	
		n+1					B	B	B	B	B	B	B	B	R	R	R	R	R	R	R	R	R	R	R	
		n+2					R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	G	G	G	
	7	n					G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R	R	R	R	
		n+1					R	R	R	R	R	R	R	R	B	B	B	B	B	B	B	B	B	B	B	
		n+2					B	B	B	B	B	B	B	B	G	G	G	G	G	G	G	G	G	G	G	
RGB565	0	n	R	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B	B	B	
	1	n	B	B	B	B	B	G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R	R	R	
	6	n	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B	B	B	B	
	7	n	B	B	B	B	G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R	R	R	R	
RGB444	0	n					R	R	R	R	R	G	G	G	G	B	B	B	B	B	B	B	B	B	B	
	1	n					B	B	B	B	B	G	G	G	G	R	R	R	R	R	R	R	R	R	R	
RGB332	0	n									R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	
	1	n									B	B	G	G	G	R	R	R	R	R	R	R	R	R	R	

LCM Output Sequence @16bit Mode (LCM_OSEQSEL = 0)

OTYPE	OSEQ0	TIME	[15:8]	[7:0]
RGB666	0	n		B B B B B B
		n+1		G G G G G G
		n+2		R R R R R R
	1	n		R R R R R R
		n+1		G G G G G G
		n+2		B B B B B B
RGB565	0	n		G G G B B B B
		n+1		R R R R R G G G
	1	n		G G G R R R R R
		n+1		B B B B B G G G
RGB444	0	n		G G G G B B B B
		n+1		B B B B R R R R
		n+2		R R R R G G G G
	1	n		G G G G R R R R
		n+1		R R R R B B B B
		n+2		B B B B G G G G
RGB332	0	n		R R R G G G B B
	1	n		B B G G G R R R

LCM Output Sequence @8bit Mode (LCM_OSEQSEL = 0)

OTYPE	OSEQ0	TIME	[15:8]	[7:0]
RGB666	0	n	R R R R R G G G G G G B B B B B B	
		n+1		R R
	1	n	B B B B B G G G G G G R R R R R R	
		n+1		B B
	2	n		R R
		n+1	R R R R R G G G G G G B B B B B B	
	3	n		B B
		n+1	B B B B B G G G G G G R R R R R R	
	4	n	B B	
		n+1	R R R R R R G G G G G G G G B B B B B B	
	5	n	R R	
		n+1	B B B B B B G G G G G G G G R R R R R R	
	6	n		
		n+1	B B B B B B R R R R R R G G G G G G	
		n+2	G G G G G G B B B B B B	
	7	n	B B B B B B	
		n+1	R R R R R R B B B B B B	
		n+2	G G G G G G R R R R R R	

OTYPE	OSEQ0	TIME	[15:8]	[7:0]
RGB565	0	n	R R R R R G G G G G	B B B B B
	1	n	B B B B B G G G G G	G R R R R R
	6	n	R R R R R G G G G G	B B B B B B
	7	n	B B B B G G G G G	R R R R R R
RGB444	0	n		R R R R G G G G B B B B
	1	n	B B B B G G G G R R R R	
RGB332	0	n		R R R G G G B B
	1	n		B B G G G R R R R

LCM Output Sequence @16bit Mode (LCM_OSEQSEL = 1)

OTYPE	OSEQ0	TIME	[15:8]	[7:0]
RGB666	0	n		B B B B B B
		n+1		G G G G G G
		n+2		R R R R R R R
	1	n		R R R R R R R R
		n+1		G G G G G G G
		n+2		B B B B B B
RGB565	0	n		R R R R R G G G
		n+1		G G G B B B B B
	1	n		B B B B B G G G
		n+1		G G G R R R R R R
RGB444	0	n		R R R R G G G G
		n+1		B B B B R R R R
		n+2		G G G G B B B B
	1	n		B B B B G G G G
		n+1		R R R R B B B B
		n+2		G G G G R R R R
RGB332	0	n		R R R G G G B B
	1	n		B B G G G R R R R

LCM Output Sequence @8bit Mode (LCM_OSEQSEL = 1)

16 ON SCREEN DISPLAY (OSD) CONTROLLER

16.1 Introduction

The display controller support 2-Layer On-Screen display (OSD) windows, each OSD window can support RGB565/1555/5515 color mode. Notice that user only can use OSD function after enabling Display controller (TFT0).

16.2 Register Summary

Name	Type	Address	Description
P_OSD0_BASE_ADR	RW	0x93000030	OSD0 Base Address
P_OSD0_PITCH	RW	0x93000034	OSD0 Pitch (Byte)
P_OSD0_RES	RW	0x93000038	OSD0 Resolution
P_OSD0_START_COOR	RW	0x9300003C	OSD0 Starting Coordinate
P_OSD0_FORMAT	RW	0x93000040	OSD0 Format
P_OSD0_CONTROL	RW	0x93000044	OSD0 Control
P_OSD1_FORMAT	RW	0x93000048	OSD1 Format
P_OSD1_CONTROL	RW	0x9300004C	OSD1 Control
P_OSD1_BASE_ADR	RW	0x93000050	OSD1 Base Address
P_OSD1_PITCH	RW	0x93000054	OSD1 Pitch (Byte)
P_OSD1_RES	RW	0x93000058	OSD1 Resolution
P_OSD1_START_COOR	RW	0x9300005C	OSD1 Starting Coordinate
P_OSD0_H_FACTOR	RW	0x93000060	OSD0 Scaling Parameter(Horizontal)
P_OSD0_V_FACTOR	RW	0x93000064	OSD0 Scaling Parameter(Vertical)
P_OSD0_PARAM	RW	0x93000068	OSD0 Scaling Parameter(field1)
P_OSD0_SCALER_RES	RW	0x9300006C	OSD0 Scaling Resolution
P_OSD1_H_FACTOR	RW	0x93000070	OSD1 Scaling Parameter(Horizontal)
P_OSD1_V_FACTOR	RW	0x93000074	OSD1 Scaling Parameter(Vertical)
P_OSD1_PARAM	RW	0x93000078	OSD1 Scaling Parameter(field1)
P_OSD1_SCALER_RES	RW	0x9300007C	OSD1 Scaling Resolution
P_OSDx_PALETTE_PTR	W	0x93000800	OSDx Palette Pointer

16.3 Register Definition

P_OSDx_BASE_ADR		0x93000030/0x93000050						OSDx Base Address	
Bit	31	30	29	28	27	26	25	24	
Function	OSD_BASEADR								
Default	0								



C

Default

0

C

Bit	Function	Type	Description	Condition
[31:0]	OSD_BASEADR	RW	OSD layer base address	-

P_OSDx_PITCH

0x93000034/0x93000054

OSDx Pitch (Byte)

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	OSD_SRCPITCH			

Default

6

6

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	OSD ACTPITCH			

Default

6

C

Bit	Function	Type	Description	Condition
[31:28]	-	-	Reserved	-
[27:16]	OSD_SRCPITCH	RW	Source OSD layer pitch	-
[15:12]	-	-	Reserved	-
[11:0]	OSD_ACTPITCH	RW	Active OSD layer pitch	-

P OSDx_RES
0x93000038/0x93000058
OSDx Resolution

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	OSD_ACTWIDTH			
Default	0	0	0	0	0			

23	22	21	20	19	18	17	16
OSD_ACTWIDTH							

0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	OSD_ACTHEIGHT			
Default	0	0	0	0	0			

7	6	5	4	3	2	1	0
OSD_ACTHEIGHT							

0

Bit	Function	Type	Description				Condition
[31:28]	-	-	Reserved				-
[27:16]	OSD_ACTWIDTH	RW	Active OSD layer width				-
[15:12]	-	-	Reserved				-
[11:0]	OSD_ACTHEIGHT	RW	Active OSD layer height				-

P OSD0_START_COOR
0x9300003C/0x9300005C
OSDx Starting Coordinate

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	OSD_STARTX			
Default	0	0	0	0	0			

23	22	21	20	19	18	17	16
OSD_STARTX							

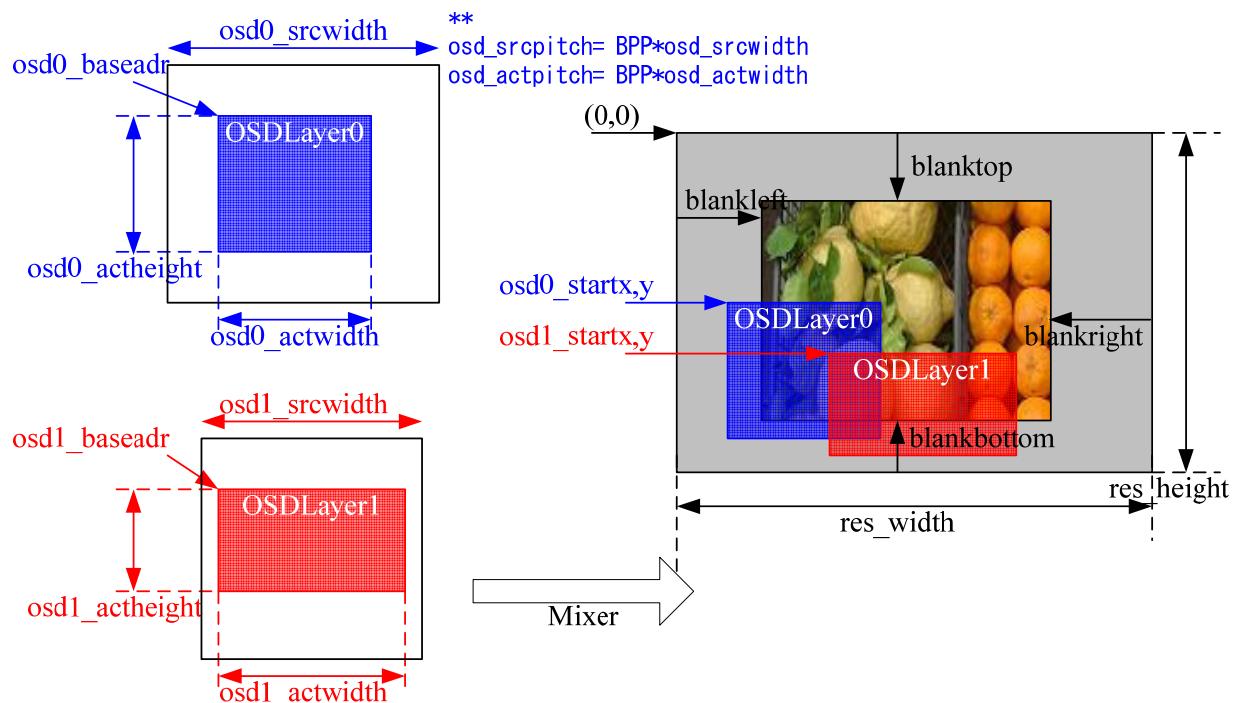
0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	OSD_STARTY			
Default	0	0	0	0	0			

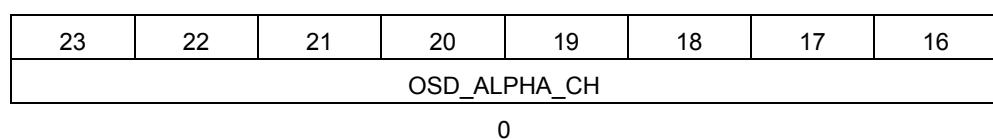
7	6	5	4	3	2	1	0
OSD_STARTY							

0

Bit	Function	Type	Description	Condition
[31:28]	-	-	Reserved	-
[27:16]	OSD_STARTX	RW	Starting Coordinate X	-
[15:12]	-	-	Reserved	-
[11:0]	OSD_STARTY	RW	Starting Coordinate Y	-



P_OSDx_FORMAT		0x93000040/0x93000048								OSDx Format			
Bit	Function	31	30	29	28	27	26	25	24				
Default	OSD_FORMAT	-	-	OSD_ALPHA_SET	-	OSD_PPAMD	-	-	-	0	0	0	0



Bit	Function	15	14	13	12	11	10	9	8
OSD_CKEY									
Default		0							

7	6	5	4	3	2	1	0
OSD_CKEY							
0							

Bit	Function	Type	Description	Condition			
[31:30]	OSD_FORMAT	RW	OSDx input format	0 RGB565			
				1 RGB5515			
				2 RGB1555			
[29:28]	-	-	Reserved	-			
[27]	OSD_ALPHA_SET	RW	OSDx alpha setting	0 Per-pixel-Alpha			
				1 Constant Alpha			
[26]	-	-	Reserved	-			
[25:24]	OSD_PPAMD		OSDx Per-pixel mode	0 PerPixelAlpha Only			
				1 ColorKey Only			
				2 PerPixelAlpha+ColorKey			
[23:16]	OSD_ALPHA_CH	RW	OSDx Alpha channel	0~64			
[15:0]	OSD_CKEY	RW	OSDx Transparent color key	-			

P OSDx_CONTROL 0x93000044/0x9300004C OSDx Control

Bit	31	30	29	28	27	26	25	24
Function	OSD_ENABLE	OSD_DMATYPE	OSD_BTYPEn	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	-	-	-	-	OSD_H_SCALE_EN	OSD_V_SCALE_EN
0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	OSD_TYPE	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
OSD_OFFSET							
0							

Bit	Function	Type	Description	Condition	
[31]	OSD_ENABLE	RW	OSDx layer enable	0	Disable
				1	Enable



Bit	Function	Type	Description	Condition	
[30]	OSD_DMATYPE	RW	OSDx DMA type	0	Progressive DMA
				1	Interlaced DMA
[29:28]	OSD_BTYPEn	RW	OSDx burst type	0	SINGLE
				1	INCR4
				2	INCR8
				3	INCR16
				-	
[27:18]	-	-	Reserved	-	
[17]	OSD_H_SCALE_EN	RW	OSDx Horizontal scaling enable	0	Disable
				1	Enable
[16]	OSD_V_SCALE_EN	RW	OSDx Vertical scaling enable	0	Disable
				1	Enable
[15:14]	-	-	Reserved	-	
[13:12]	OSD_TYPE		OSDx data type	0	16bit Per Pixel
				1	8bit Per Pixel
				2	4bit Per Pixel
				3	1bit Per Pixel
				-	
[11:8]	-	-	Reserved	-	
[7:0]	OSD_OFFSET	RW	OSDx Palette Table Offset Value	-	

P OSDx H FACTOR 0x93000060/0x93000070 OSDx Scaling Parameter(Horizontal)

Default 0

0

Default 0

0

Bit	Function	Type	Description	Condition
[31:16]	OSD_H_INITIAL	RW	Horizontal scaling initial value	-

Bit	Function	Type	Description				Condition
[15:0]	OSD_H_FACTOR	RW	Horizontal scaling factor				-

NOTE:

$$\text{OSD_H_FACTOR} = \text{Floor}(2048 * (\text{OSD_ACTWIDTH}-1) / (\text{OSD_SCLWIDTH}-1))$$

P OSDx_V_FACTOR 0x93000064/0x93000074 OSDx Scaling Parameter(Vertical)

Bit	31	30	29	28	27	26	25	24
Function	OSD_V_INITIAL0							

Default 0

23	22	21	20	19	18	17	16
OSD_V_INITIAL0							

0

Bit	15	14	13	12	11	10	9	8
Function	OSD_V_FACTOR							

Default 0

7	6	5	4	3	2	1	0
OSD_V_FACTOR							

0

Bit	Function	Type	Description				Condition
[31:16]	OSD_V_INITIAL0	RW	Vertical scaling initial value0				-
[15:0]	OSD_V_FACTOR	RW	Vertical scaling factor				-

NOTE:

$$\text{OSD_V_FACTOR} = \text{Floor}(2048 * (\text{OSD_ACTHEIGHT}-1) / (\text{OSD_SCLHEIGHT}-1))$$

P OSDx_PARAM 0x93000068/0x93000078 OSDx Scaling Parameter(field1)

Bit	31	30	29	28	27	26	25	24
Function	OSD_V_INITIAL1							

Default 0

23	22	21	20	19	18	17	16
OSD_V_INITIAL1							

0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:16]	OSD_V_INITIAL1	RW	Vertical scaling initial value1				-
[15:0]	-	-	Reserved				-

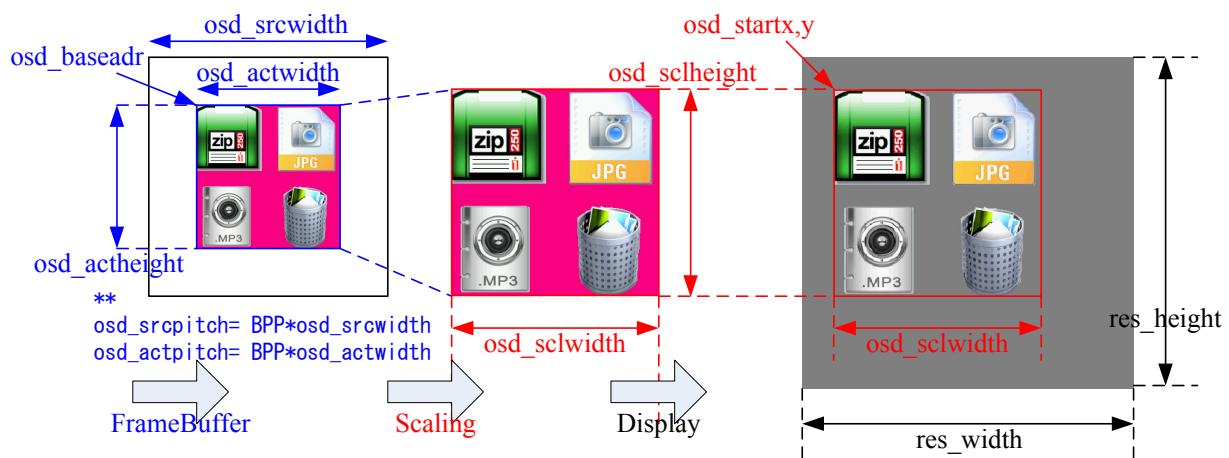
P OSD0_SCALER_RES 0x9300006C/0x9300007C OSDx Scaling Resolution							
Bit	31	30	29	28	27	26	25
Function OSD_SCLWIDTH							
Default 0							

23	22	21	20	19	18	17	16
OSD_SCLWIDTH							
0							

Bit	15	14	13	12	11	10	9	8
Function	OSD_SCLHEIGHT							
Default	0							

7	6	5	4	3	2	1	0
OSD_SCLHEIGHT							
0							

Bit	Function	Type	Description				Condition
[31:16]	OSD_SCLWIDTH	RW	OSD layer scaling width				-
[15:0]	OSD_SCLHEIGHT	RW	OSD layer scaling height				-


P_OSDx_PALETTE_PTR 0x93000800

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
PALETTEn								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PALETTEn							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	-	Reserved	-
[15:0]	PALETTEn	W	OSD Palette pattern	-

NOTE: ADDRESS(PALETTEn) = PALETTEPTR + n * 4

For example, the address of PALETTE3 is 0x9300080C

17 INDEPENDENT SCALING CONTROLLER

17.1 Introduction

This scaling controller (Scaler0) let user can Zoom-in or Zoom-out OSD independently besides the built-in scaling controller (Scaler1).

17.2 Register Summary

Name	Type	Address	Description
P_SCALER0_FRAME_ADR	RW	0x92007000	Scaler0 Frame Buffer Address
P_SCALER0_CB_STR_ADR	RW	0x92007004	Scaler0 Cb Start Address
P_SCALER0_CR_STR_ADR	RW	0x92007008	Scaler0 Cr Start Address
P_SCALER0_WB_STR_ADR	RW	0x9200700C	Scaler0 Write Back Address
P_SCALER0_IMG_PITCH	RW	0x92007010	Scaler0 Image Pitch (byte)
P_SCALER0_IMG_RES	RW	0x92007014	Scaler0 Image Resolution
P_SCALER0 MCU_PITCH	RW	0x92007018	Scaler0 Macro-block Pitch (byte)
P_SCALER0_OUTPUT_PITCH	RW	0x92007020	Scaler0 Output Pitch (byte)
P_SCALER0_OUTPUT_RES	RW	0x92007024	Scaler0 Output Resolution
P_SCALER0_H_PARAM	RW	0x92007028	Scaler0 Scaling Parameter(Horizontal)
P_SCALER0_V_PARAM	RW	0x9200702C	Scaler0 Scaling Parameter(Vertical)
P_SCALER0_DITHER_MAP0	RW	0x92007030	Scaler0 Dither Map 0
P_SCALER0_DITHER_MAP1	RW	0x92007034	Scaler0 Dither Map 1
P_SCALER0_CONTROL	RW	0x92007040	Scaler0 Scaling Control
P_SCALER0_INT_ENABLE	RW	0x92007080	Scaler0 Interrupt Enable
P_SCALER0_INT_SOURCE	RW	0x92007084	Scaler0 Interrupt Source

17.3 Register Definition

P_SCALER0_FRAME_ADR 0x92007000 Scaler0 Frame Buffer Address							
Bit	31	30	29	28	27	26	25
Function	SCL0_FRAMEADR						
Default	0						
	23	22	21	20	19	18	17
	SCL0_FRAMEADR						
	0						

Bit	15	14	13	12	11	10	9	8
Function	SCL0_FRAMEADR							
Default	0							
	7	6	5	4	3	2	1	0
	SCL0_FRAMEADR							
	0							

Bit	Function	Type	Description					Condition
[31:0]	SCL0_FRAMEADR	RW	Scaler0 Frame buffer address					-

P_SCALER0_CB_STR_ADR 0x92007004 Scaler0 Cb Start Address								
Bit	31	30	29	28	27	26	25	24
Function	SCL0_CB_STR_ADR							
Default	0							
	23	22	21	20	19	18	17	16
	SCL0_CB_STR_ADR							
	0							
Bit	15	14	13	12	11	10	9	8
Function	SCL0_CB_STR_ADR							
Default	0							
	7	6	5	4	3	2	1	0
	SCL0_CB_STR_ADR							
	0							

Bit	Function	Type	Description					Condition
[31:0]	SCL0_CB_STR_ADR	RW	Scaler0 Cb start address					-

P_SCALER0_CR_STR_ADR 0x92007008 Scaler0 Cr Start Address								
Bit	31	30	29	28	27	26	25	24
Function	SCL0_CR_STR_ADR							
Default	0							
	31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16
SCL0_CR_STR_ADDR							

0

Bit	15	14	13	12	11	10	9	8
SCL0_CR_STR_ADDR								

Default 0

7	6	5	4	3	2	1	0
SCL0_CR_STR_ADDR							

0

Bit	Function	Type	Description				Condition
[31:0]	SCL0_CR_STR_ADDR	RW	Scaler0 Cr start address				-

P_SCALER0_WB_STR_ADDR 0x9200700C Scaler0 Write Back Address

Bit	31	30	29	28	27	26	25	24
SCL0_WB_STR_ADDR								

Default 0

23	22	21	20	19	18	17	16
SCL0_WB_STR_ADDR							

0

Bit	15	14	13	12	11	10	9	8
SCL0_WB_STR_ADDR								

Default 0

7	6	5	4	3	2	1	0
SCL0_WB_STR_ADDR							

0

Bit	Function	Type	Description				Condition
[31:0]	SCL0_WB_STR_ADDR	RW	Scaler0 Write Back Address				-

P_SCALER0_IMG_PITCH 0x92007010
Scaler0 Image Pitch (byte)

Bit	31	30	29	28	27	26	25	24
-----	----	----	----	----	----	----	----	----

Function SCL0_SRC_PITCH

Default 0

23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----

SCL0_SRC_PITCH

0

Bit	15	14	13	12	11	10	9	8
-----	----	----	----	----	----	----	---	---

Function SCL0_ACT_PITCH

Default 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

SCL0_ACT_PITCH

0

Bit	Function	Type	Description	Condition
[31:16]	SCL0_SRC_PITCH	RW	Scaler0 Image Source Pitch (byte)	-
[15:0]	SCL0_ACT_PITCH	RW	Scaler0 Image Active Pitch (byte)	-

P_SCALER0_IMG_RES 0x92007014
Scaler0 Image Resolution

Bit	31	30	29	28	27	26	25	24
-----	----	----	----	----	----	----	----	----

Function SCL0_ACT_WIDTH

Default 0

23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----

SCL0_ACT_WIDTH

0

Bit	15	14	13	12	11	10	9	8
-----	----	----	----	----	----	----	---	---

Function SCL0_ACT_HEIGHT

Default 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

SCL0_ACT_HEIGHT

0

Bit	Function	Type	Description				Condition
[31:16]	SCL0_ACT_WIDTH	RW	Scaler0 Active Image Width				-
[15:0]	SCL0_ACT_HEIGHT	RW	Scaler0 Active Image Height				-

P_SCALER0 MCU PITCH 0x92007018 Macro-block Pitch (byte)

Bit	31	30	29	28	27	26	25	24
Function	SCL0_MCUOPITCH							
Default	0							
Bit	23	22	21	20	19	18	17	16
Function	SCL0_MCUOPITCH							
Default	0							
Bit	15	14	13	12	11	10	9	8
Function	SCL0_MCUIPITCH							
Default	0							
Bit	7	6	5	4	3	2	1	0
Function	SCL0_MCUIPITCH							
Default	0							

Bit	Function	Type	Description				Condition
[31:16]	SCL0_MCUOPITCH	RW	Scaler0 Outout MCU pitch (byte)				-
[15:0]	SCL0_MCUIPITCH	RW	Scaler0 Input MCU pitch (byte)				-

Limit:

2 passes are needed. One is for horizontal scalingUp/Down and another is for vertical scalingUp/Down.

SCL0_BTYPEn must be 1 (INCR4).

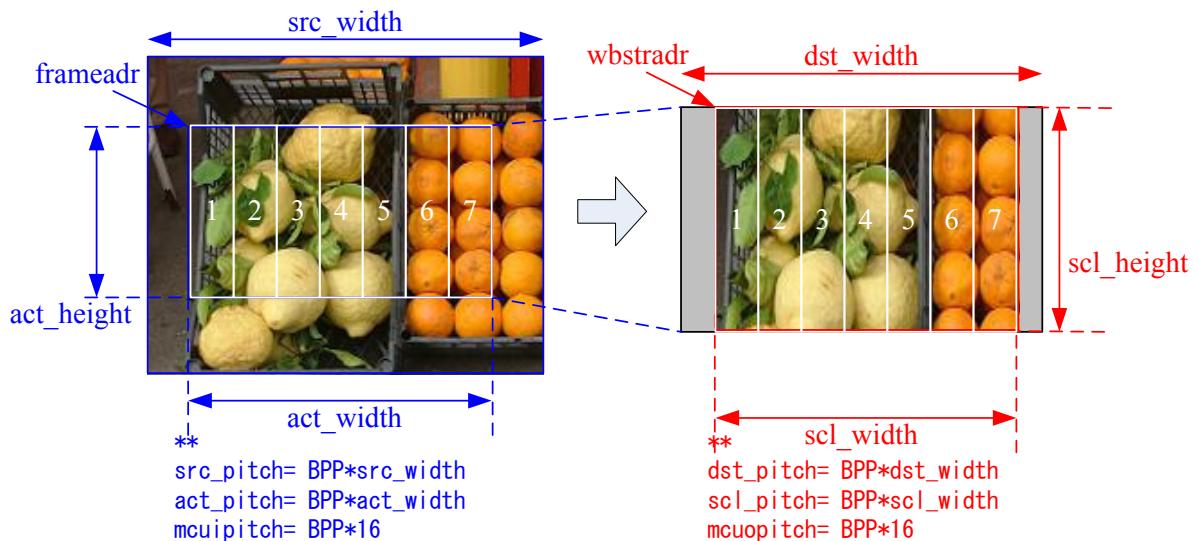
SCL0_ACT_WIDTH must be equal to SCL0_SCL_WIDTH

SCL0_ACT_WIDTH%16 == 0

Advantage:

Large Size: Scaling resolution will not limited by the line buffer.

Good Quality: Vertical scaling up uses the hermite algorithm (4 point interpolation) instead of bilinear algorithm


P_SCALER0_OUTPUT_PITCH 0x92007020
Scaler0 Output Pitch (byte)

Bit	31	30	29	28	27	26	25	24
Function								
Default	SCL0_DST_PITCH							

0

23	22	21	20	19	18	17	16
SCL0_DST_PITCH							

0

Bit
Function
Default

15	14	13	12	11	10	9	8
SCL0_SCL_PITCH							

0

7	6	5	4	3	2	1	0
SCL0_SCL_PITCH							

0

Bit	Function	Type	Description	Condition
[31:16]	SCL0_DST_PITCH	RW	Scaler0 Output Target Image (byte)	-
[15:0]	SCL0_SCL_PITCH	RW	Scaler0 Output Scaling Image (byte)	-

P_SCALER0_OUTPUT_RES 0x92007024
Scaler0 Output Resolution

Bit	31	30	29	28	27	26	25	24
Function								
Default	SCL0_SCL_WIDTH							

0

23	22	21	20	19	18	17	16
SCL0_SCL_WIDTH							

0

Bit	15	14	13	12	11	10	9	8
SCL0_SCL_HEIGHT								

Default 0

7	6	5	4	3	2	1	0
SCL0_SCL_HEIGHT							

0

Bit	Function	Type	Description					Condition
[31:16]	SCL0_SCL_WIDTH	RW	Scaler0 Scaling Output Image width					-
[15:0]	SCL0_SCL_HEIGHT	RW	Scaler0 Scaling Output Image height					-

P_SCALER0_H_PARAM 0x92007028 Scaler0 Scaling Parameter(Horizontal)

Bit	31	30	29	28	27	26	25	24
SCL0_H_INITIAL								

Default 0

23	22	21	20	19	18	17	16
SCL0_H_INITIAL							

0

Bit	15	14	13	12	11	10	9	8
SCL0_H_FACTOR								

Default 0

7	6	5	4	3	2	1	0
SCL0_H_FACTOR							

0

Bit	Function	Type	Description					Condition
[31:16]	SCL0_H_INITIAL	RW	Scaler0 Horizontal scaling initial value					-
[15:0]	SCL0_H_FACTOR	RW	Scaler0 Horizontal scaling factor					-

NOTE:

HscaleFactor = Floor(2048*(ACT_WIDTH-1)/(SCL_WIDTH-1)) // SCL_FUNCTION = 1

HScaleUpFactor = Floor((65536*ACT_WIDTH)/SCL_WIDTH) // SCL_FUNCTION = 0

HScaleDownFactor = ceil ((65536*SCL_WIDTH)/ACT_WIDTH)

P_SCALER0_V_PARAM 0x9200702C Scaler0 Scaling Parameter(Vertical)								
Bit	31	30	29	28	27	26	25	
Function	SCL0_V_INITIAL							
Default	0							

Bit	23	22	21	20	19	18	17	16
SCL0_V_INITIAL								
0								

Bit	15	14	13	12	11	10	9	8
Function	SCL0_V_FACTOR							
Default	0							

Bit	7	6	5	4	3	2	1	0
SCL0_V_FACTOR								
0								

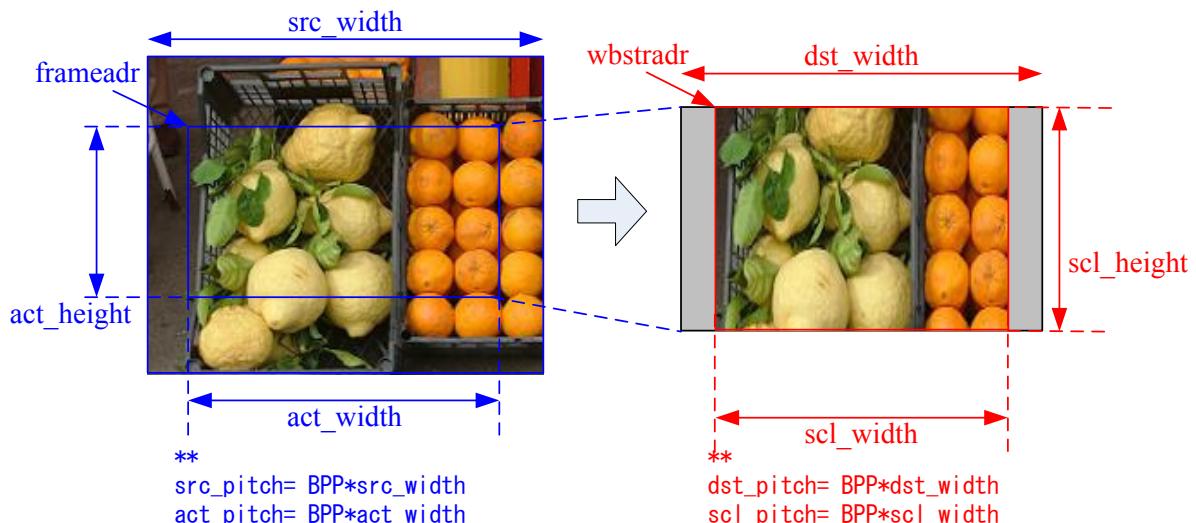
Bit	Function	Type	Description	Condition
[31:16]	SCL0_V_INITIAL	RW	Scaler0 Vertical scaling initial value	-
[15:0]	SCL0_V_FACTOR	RW	Scaler0 Vertical scaling factor	-

NOTE:

VscaleFactor = Floor(2048*(ACT_HEIGHT-1)/(SCL_HEIGHT-1)) // SCL_FUNCTION = 1

VScaleUpFactor = Floor((65536*ACT_HEIGHT)/SCL_HEIGHT) // SCL_FUNCTION = 0

VScaleDownFactor = ceil ((65536*SCL_HEIGHT)/ACT_HEIGHT)



P_SCALER0_DITHER_MAP0 0x920007030
Scaler0 Dithering Map 0

Bit	31	30	29	28	27	26	25	24	
Function	d00					d01			
Default	0					0			
Bit	23	22	21	20	19	18	17	16	
Function	d02					d03			
Default	0					0			
Bit	15	14	13	12	11	10	9	8	
Function	d10					d11			
Default	0					0			
Bit	7	6	5	4	3	2	1	0	
Function	d12					d13			
Default	0					0			

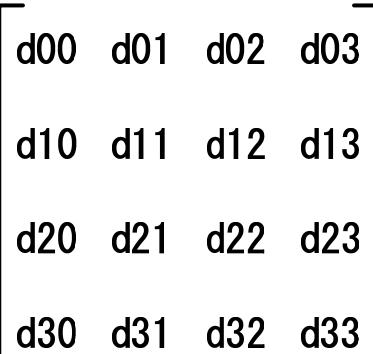
Bit	Function	Type	Description	Condition
[31:28]	d00	RW	Dithering Map parameter d00	-
[27:24]	d01	RW	Dithering Map parameter d01	-
[23:20]	d02	RW	Dithering Map parameter d02	-
[19:16]	d03	RW	Dithering Map parameter d03	-
[15:12]	d10	RW	Dithering Map parameter d10	-
[11:8]	d11	RW	Dithering Map parameter d11	-
[7:4]	d12	RW	Dithering Map parameter d12	-
[3:0]	d13	RW	Dithering Map parameter d13	-

P_SCALER0_DITHER_MAP1 0x920007030
Scaler0 Dithering Map 1

Bit	31	30	29	28	27	26	25	24	
Function	d20					d21			
Default	0					0			
Bit	23	22	21	20	19	18	17	16	
Function	d22					d23			
Default	0					0			
Bit	15	14	13	12	11	10	9	8	
Function	d30					d31			
Default	0					0			

7	6	5	4	3	2	1	0
d32				d33			
0				0			

Bit	Function	Type	Description	Condition
[31:28]	d20	RW	Dithering Map parameter d20	-
[27:24]	d21	RW	Dithering Map parameter d21	-
[23:20]	d22	RW	Dithering Map parameter d22	-
[19:16]	d23	RW	Dithering Map parameter d23	-
[15:12]	d30	RW	Dithering Map parameter d30	-
[11:8]	d31	RW	Dithering Map parameter d31	-
[7:4]	d32	RW	Dithering Map parameter d32	-
[3:0]	d33	RW	Dithering Map parameter d33	-


P_SCALER0_CONTROL 0x92007040
Scaler0 Control

Bit	31	30	29	28	27	26	25	24
Function	-	-	SCL0_BTYPEn	SCL0_V_EN	SCL0_V_SEL	SCL0_H_EN	SCL0_H_SEL	
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
SCL0_FUNCTION	-	-	SCL0_VBBON	SCL0_ITYPE	SCL0_IFORMAT		
0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	SCL0_DITHSEQ	SCL0_DITHSEL	SCL0_DITHEREN	
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SCL0_ODDLINE_EN	-	-	-	SCL0_AMPSEL	SCL0_OTYPE	SCL0_OFORMAT	
0	0	0	0	0	0	0	0

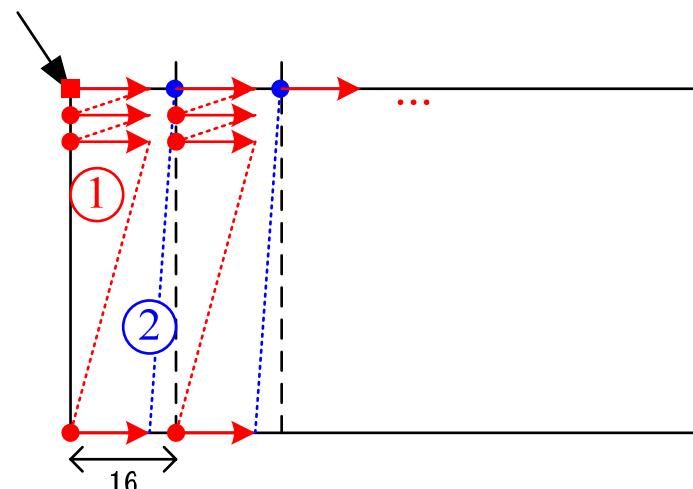
Bit	Function	Type	Description	Condition						
[31:30]	-	-	Reserved	-						
[29:28]	SCL0_BTYPE	RW	Scaler0 burst type	0	SINGLE					
				1	INCR4					
				2	INCR8					
				3	INCR16					
[27]	SCL0_V_EN	RW	Scaler0 vertical scaling enable	-						
[26]	SCL0_V_SEL	RW	Scaler0 vertical scaling select	0	Vertical Scaling Up					
				1	Vertical Scaling Down					
[25]	SCL0_H_EN	RW	Scaler0 horizontal scaling enable	-						
[24]	SCL0_H_SEL	RW	Scaler0 horizontal scaling select	0	Horizontal Scaling Up					
				1	Horizontal Scaling Down					
[23]	SCL0_FUNCTION	RW	Scaler0 scaling algorithm	0	Interpolation					
				1	Duplication/Drop					
[22:21]	-	-	Reserved	-						
[20]	SCL0_VBBON	RW	Scaler0 vertical scaling Block-by-Block	-						
[19:18]	SCL0_ITYPE	RW	Scaler0 input type		RBG	YCbCr	YCbCrSep			
				0	RGB565	YCbCr	Y400			
				1	RGB1555	4Y4Cb4Y4Cr	Y420			
				2	RGB888	YCbCr	Y422			
				3	RGB5515		Y444			
[17:16]	SCL0_IFORMAT	RW	Scaler0 input format	0	RGB Color Space					
				1	YCbCr Color Space					
				2	YCbCr separate					
[15:12]	-	-	Reserved	-						
[11:10]	SCL0_DITHSEQ	RW	Scaler0 Dithering sequence	-						
[9]	SCL0_DITHSEL	RW	Scaler0 Dithering selection	0	Ordered Dithering					
				1	HorizontalErrorDiffusion					
[8]	SCL0_DITHEREN	RW	Scaler0 Dithering enable	-						
[7]	SCL0_ODDLINE_EN	RW	Scaler0 Oddline function enable	Cb, Cr Start at Odd Line when enable.						
[6:4]	-	-	Reserved	-						

Bit	Function	Type	Description	Condition	
[3]	SCL0_AMPSEL	RW	Scaler0 Amplitude Selection	0	0-255 RGB <-> 0-255 YCbCr
				1	0-255 RGB <-> 16-235 YCbCr
[2:1]	SCL0_OTYPE	RW	Scaler0 output type		RBG
				0	RGB565
				1	RGB1555
				2	RGB888
				3	RGB5515
[0]	SCL0_OFORMAT	RW	Scaler0 output format	0	RGB Color Space
				1	YCbCr Color Space

LIMIT:

SCL0_VBBON	SCL0_FUNCTION	SCL0_V_EN	SCL_H_EN	SCL_V_SEL	SCL_H_SEL
1	0	0	0	0/1	0
0	1	0/1	0/1	0	0
0	0	0/1	0/1	0/1	0/1

YCbCrSep Format is valid only when SCL0_VBBON= 0 and SCL0_FUNCTION= 0

Vertical ScalingUp/Down Block-by-Block

P_SCALER0_INT_ENABLE 0x92007080
Scaler0 Interrupt Enable

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SCL0_SCL_DONE_INT_EN	SCL0_SCL_START_INT_EN
0	0	0	0	0	0	0	0

Bit	Function		Type	Description				Condition
[31:2]	-		-	Reserved				-
[1]	SCL0_SCL_DONE_INT_EN		RW	Scaler0 Scaling Done Interrupt Enable				-
[0]	SCL0_SCL_START_INT_EN		RW	Scaler0 Scaling Start Interrupt Enable				-

P_SCALER0_INT_SOURCE 0x92007084
Scaler0 Interrupt Source

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SCL0_SCL_DONE	SCL0_SCL_START
0	0	0	0	0	0	0	0

Bit	Function		Type	Description				Condition
[31:2]	-		-	Reserved				-
[1]	SCL0_SCL_DONE		RWC	Scaler0 Scaling Done				-
[0]	SCL0_SCL_START		RWC	Scaler0 Scaling Start				Trigger to start scaling engine

NOTE:

"Scaling Done" Interrupt will be sent only when bit [1] of P_SCALER0_INT_ENABLE and P_SCALER0_INT_SOURCE are both set to 1.

18 BUILT-IN SCALING CONTROLLER

18.1 Introduction

Note that users can only use the built-in scaling controller (Scaler1) after enabling display controller (TFT0).

18.2 Register Summary

Name	Type	Address	Description
P_SCALER1_RES	RW	0x9300001C	Scaler1 Scaling Resolution
P_SCALER1_H_FACTOR	RW	0x93000020	Scaler1 Scaling Parameter (Horizontal)
P_SCALER1_V_FACTOR	RW	0x93000024	Scaler1 Scaling Parameter (Vertical)
P_SCALER1_CONTROL	RW	0x93000028	Scaler1 Scaling Control

18.3 Register Definition

P_SCALER1_RES 0x9300001C Scaler1 Scaling Resolution											
Bit	31	30	29	28	27	26	25				
Function	-	-	-	-	SCL1_SCL_WIDTH						
Default	0	0	0	0	0						
SCL1_SCL_WIDTH											
0											
Bit	15	14	13	12	11	10	9	8			
Function	-	-	-	-	SCL1_SCL_HEIGHT						
Default	0	0	0	0	0						
SCL1_SCL_HEIGHT											
0											

Bit	Function	Type	Description	Condition
[31:28]	-	-	Reserved	-
[27:16]	SCL1_SCL_WIDTH	RW	Scaler1 scaling width.	-
[15:12]	-	-	Reserved	-
[11:0]	SCL1_SCL_HEIGHT	RW	Scaler1 scaling height.	-



P_SCALER1_H_FACTOR		0x93000020		Scaler1 Scaling Parameter (Horizontal)			
Bit	31	30	29	28	27	26	25
Function	SCL1_SCL_HINITVAL						
Default	0						
	23	22	21	20	19	18	17
	SCL1_SCL_HINITVAL						
	0						
Bit	15	14	13	12	11	10	9
Function	SCL1_SCL_HFACTOR						
Default	0						
	7	6	5	4	3	2	1
	SCL1_SCL_HFACTOR						
	0						
Bit	Function	Type	Description			Condition	
[31:16]	SCL1_SCL_HINITVAL	RW	Scaler1 Horizontal scaling initial value			-	
[15:0]	SCL1_SCL_HFACTOR	RW	Scaler1 Horizontal scaling factor			-	

NOTE:

HScaleUpFactor = Floor((65536*ACT WIDTH)/ SCL1 SCL WIDTH)

HScaleDownFactor = ceil ((65536* SCL1 SCL WIDTH)/ACT WIDTH)

Bit	Function	Type	Description				Condition
[31:16]	SCL1_SCL_VINITVAL0	RW	Scaler1 Vertical scaling initial value0				-
[15:0]	SCL1_SCL_VFACTOR	RW	Scaler1 Vertical scaling factor				-

NOTE:

$$\text{VScaleUpFactor} = \text{Floor}((65536 * \text{ACT_HEIGHT}) / \text{SCL1_SCL_HEIGHT})$$

$$\text{VScaleDownFactor} = \text{ceil}((65536 * \text{SCL1_SCL_HEIGHT}) / \text{ACT_HEIGHT})$$

P_SCALER1_CONTROL 0x93000028

Scaler1 Scaling Control

Bit	31	30	29	28	27	26	25	24
-----	----	----	----	----	----	----	----	----

Function SCL1_SCL_VINITVAL1

Default 0

23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----

SCL1_SCL_VINITVAL1

0

Bit	15	14	13	12	11	10	9	8
-----	----	----	----	----	----	----	---	---

Function - - - - - - - -

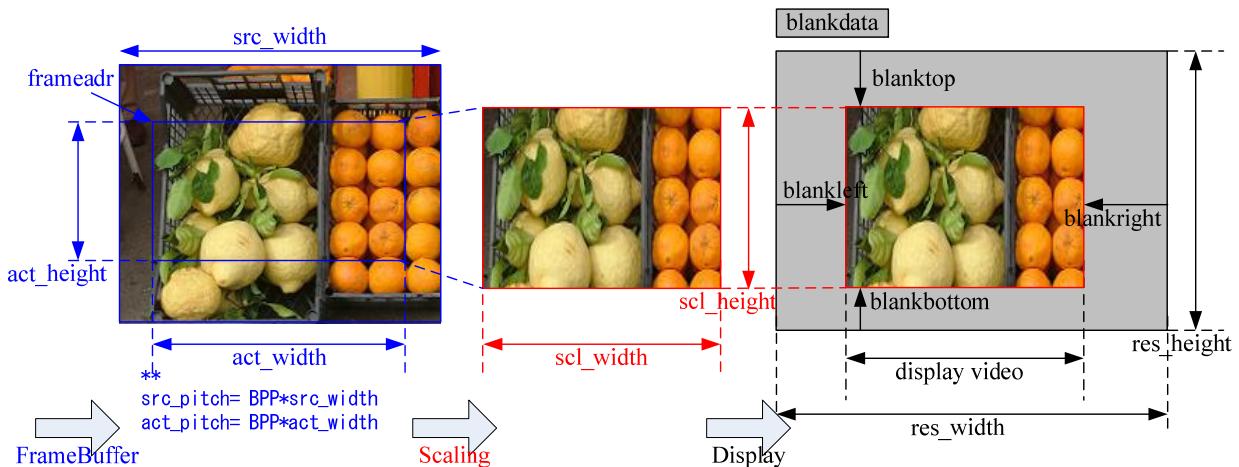
Default 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

SCL1_V_EN SCL1_V_SEL SCL1_H_EN SCL1_H_SEL

0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition	
[31:16]	SCL1_SCL_VINITVAL0	RW	Scaler1 Vertical scaling initial value1	-	
[15:4]	-	-	Reserved	-	
[3]	SCL1_V_EN	RW	Scaler1 Vertical Scaling Enable.	0	Disable
				1	Enable
[2]	SCL1_V_SEL	RW	Scaler1 Vertical Scaling Selection.	0	Vertical Scaling Up
				1	Vertical Scaling Down
[1]	SCL1_H_EN	RW	Scaler1 Horizontal Scaling Enable.	0	Disable
				1	Enable
[0]	SCL1_H_SEL	RW	Scaler1 Horizontal Scaling Selection.	0	Horizontal Scaling Up
				1	Horizontal Scaling Down



19 Scaler2

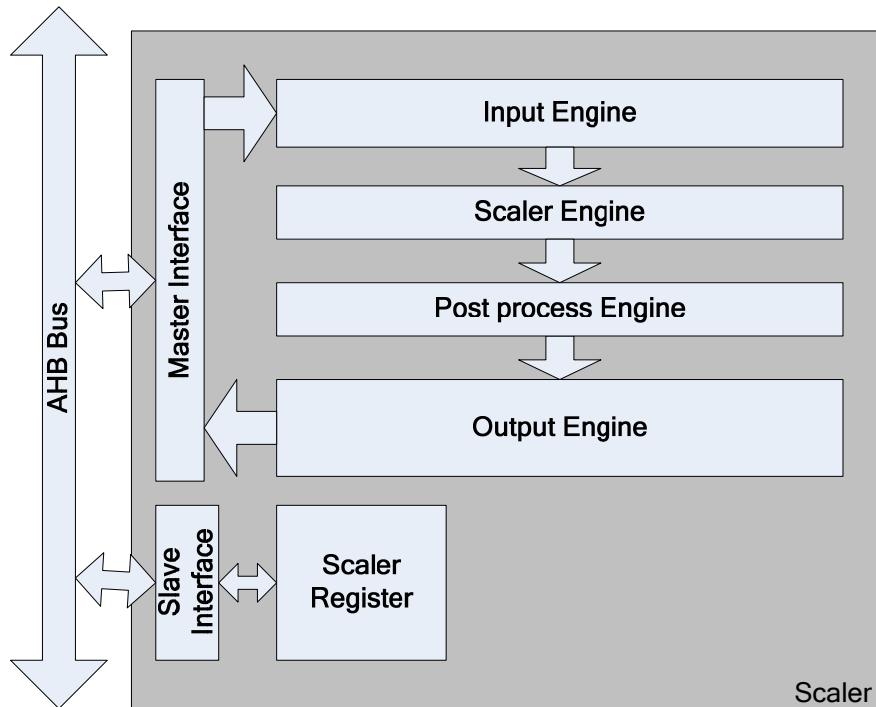
19.1 Introduction

The scaler2 controller built-in GPL32900A is used to transform output data from JPEG decode engine to the format which is supported by PPU. It also supports transform data to format supported by JPEG encode engine for image capture or video record application. Total 15 types of input format and 13 types of output format are supported. Typical AMBA interface with internal FIFO is supported for better performance.

19.2 Features

- Supports input size up-to 8191x8191
- Supports output size up-to 8191x8191
- Supports YUV separate type input format include YUV422, YUV422v, YUV420, YUV411, YUV444 and Y only, which is suitable for JPEG decode.
- Supports frame type input format include RGB1555, RGB565, RGBG, GRGB, YUYV and UYVY, which is suitable for JPEG encode.
- Support six special input format YUYV8X32/16X32/32X32 and YUYV8X64/16X64/64X64 which is suitable for PPU's sprite display mode
- Support YUV separate type output format include YUV422, YUV420, YUV411, YUV444 and Y only which is suitable for JPEG encode.
- Support frame type output format include RGB1555, RGB565, RGBG, GRGB, YUYV and UYVY which is suitable for PPU's display.
- Support six special output format YUYV8X32/16X32/32X32 and YUYV8X64/16X64/64X64 which is suitable for PPU's sprite display mode.
- Support horizontal interpolation
- Support vertical interpolation for output width less than 1024.
- Support arbitrary scale-down and scale-up with 24 bits accuracy
- Support AB FIFO input mode which is suitable for system with small external RAM
- Support AB FIFO output mode which is suitable for multi-time scale down process
- Support time sharing multi-scale task mode, which is useful to do multiple scale job concurrently
- Support external line buffer mode which is suitable to do high quality up-sampling which large size
- Support both polling mode and interrupt mode.
- Support Y gamma table function.
- Support Y 256 section histogram.
- Support Y/U/V individual max/min/summation value calculation.
- Support Color matrix for Hue/Saturation control.

19.3 Block Diagram



19.4 Register Summary

Name	Address	Description
P_SCALER2_CTRL	0x93002000	Scaler control register.
P_SCALER2_OB_COLOR	0x93002004	Scaler out-of-boundary color register,
P_SCALER2_OUTX_WIDTH	0x93002008	Scaler output X width register.
P_SCALER2_OUTY_WIDTH	0x9300200C	Scaler output Y width register.
P_SCALER2_X_FACTOR	0x93002010	Scaler X scale factor register.
P_SCALER2_Y_FACTOR	0x93002014	Scaler Y scale factor register.
P_SCALER2_X_START	0x93002018	Scaler X Start register.
P_SCALER2_Y_START	0x9300201C	Scaler Y Start register.
P_SCALER2_IN_X_Width	0x93002020	Scaler input X width register.
P_SCALER2_IN_Y_Width	0x93002024	Scaler input Y width register.
P_SCALER2_Y_INPTR	0x93002028	Scaler Y input pointer register
P_SCALER2_U_INPTR	0x9300202C	Scaler U input pointer register
P_SCALER2_V_INPTR	0x93002030	Scaler V input pointer register
P_SCALER2_Y_OUTPTR	0x93002034	Scaler Y output pointer register
P_SCALER2_U_OUTPTR	0x93002038	Scaler U output pointer register
P_SCALER2_V_OUTPTR	0x9300203C	Scaler V output pointer register
P_SCALER2_CUR_LINE	0x93002040	Scaler current line register.

Name	Address	Description
P_SCALER2_A11	0x93002044	Scaler color matrix A11 parameter.
P_SCALER2_A12	0x93002048	Scaler color matrix A12 parameter.
P_SCALER2_A13	0x9300204C	Scaler color matrix A13 parameter.
P_SCALER2_A21	0x93002050	Scaler color matrix A21 parameter.
P_SCALER2_A22	0x93002054	Scaler color matrix A22 parameter.
P_SCALER2_A23	0x93002058	Scaler color matrix A23 parameter.
P_SCALER2_A31	0x9300205C	Scaler color matrix A31 parameter.
P_SCALER2_A32	0x93002060	Scaler color matrix A32 parameter.
P_SCALER2_A33	0x93002064	Scaler color matrix A33 parameter.
P_SCALER2_IN_RX_WIDTH	0x93002068	Scaler input real X width register.
P_SCALER2_IN_RY_WIDTH	0x9300206C	Scaler input real Y width register.
P_SCALER2_OUT_OFFSET	0x93002070	Scaler output offset register.
P_SCALER2_LB_PTR	0x93002074	Scaler external line buffer pointer.
P_SCALER2_INT	0x9300207C	Scaler interrupt register.
P_SCALER2_POST	0x93002080	Scaler post control module.
P_SCALER2_MAX_Y	0x93002084	Scaler maximum Y register.
P_SCALER2_MIN_Y	0x93002088	Scaler minimum Y register.
P_SCALER2_SUM_Y	0x9300208C	Scaler Y summary register.
P_SCALER2_MAX_U	0x93002090	Scaler maximum U register.
P_SCALER2_MIN_U	0x93002094	Scaler minimum U register.
P_SCALER2_SUM_U	0x93002098	Scaler U summary register.
P_SCALER2_MAX_V	0x9300209C	Scaler maximum V register.
P_SCALER2_MIN_V	0x930020A0	Scaler minimum V register.
P_SCALER2_SUM_V	0x930020A4	Scaler V summary register.
P_SCALER2_Y_HIS0	0x930020C0	Scaler Y histogram 0 register. (0~15)
P_SCALER2_Y_HIS1	0x930020C4	Scaler Y histogram 1 register. (16~31)
P_SCALER2_Y_HIS2	0x930020C8	Scaler Y histogram 2 register. (32~47)
P_SCALER2_Y_HIS3	0x930020CC	Scaler Y histogram 3 register. (48~63)
P_SCALER2_Y_HIS4	0x930020D0	Scaler Y histogram 4 register. (64~79)
P_SCALER2_Y_HIS5	0x930020D4	Scaler Y histogram 5 register. (80~95)
P_SCALER2_Y_HIS6	0x930020D8	Scaler Y histogram 6 register. (96~111)
P_SCALER2_Y_HIS7	0x930020DC	Scaler Y histogram 7 register. (112~127)
P_SCALER2_Y_HIS8	0x930020E0	Scaler Y histogram 8 register. (128~143)
P_SCALER2_Y_HIS9	0x930020E4	Scaler Y histogram 9 register. (144~159)
P_SCALER2_Y_HIS10	0x930020E8	Scaler Y histogram 10 register. (159~175)
P_SCALER2_Y_HIS11	0x930020EC	Scaler Y histogram 11 register. (176~191)
P_SCALER2_Y_HIS12	0x930020F0	Scaler Y histogram 12 register. (192~207)

Name	Address	Description
P_SCALER2_Y_HIS13	0x930020F4	Scaler Y histogram 13 register. (208~223)
P_SCALER2_Y_HIS14	0x930020F8	Scaler Y histogram 14 register. (224~239)
P_SCALER2_Y_HIS15	0x930020FC	Scaler Y histogram 15 register. (240~255)
P_SCALER2_CONTI_W1	0x93002100	Continuous mode write register 1.
P_SCALER2_CONTI_R1	0x93002104	Continuous mode read register 1.
P_SCALER2_CONTI_R2	0x9300108	Continuous mode read register 2.
P_SCALER2_Y_GAMMA0	0x93002400	Scaler Y gamma table 0.
P_SCALER2_Y_GAMMA1	0x93002404	Scaler Y gamma table 1.
		...
P_SCALER2_Y_GAMMA254	0x930027F8	Scaler Y gamma table 254.
P_SCALER2_Y_GAMMA255	0x930027FC	Scaler Y gamma table 255.
P_SCALER2_Y_HIS0	0x93002800	Scaler Y histogram 0 register.
P_SCALER2_Y_HIS1	0x93002804	Scaler Y histogram 1 register.
		...
P_SCALER2_Y_HIS254	0x93002FF8	Scaler Y histogram 254 register.
P_SCALER2_Y_HIS255	0x93002FFC	Scaler Y histogram 255 register.

19.5 Register Definition

19.5.1 P_SCALER2_CTRL

The P_SCALER2_CTRL register is used to control scaler operation mode.

P_SCALER2_CTRL 0x93002000								Scaler Control Register								
Bit	31	30	29	28	27	26	25	24								
Function	ARGB8888_OUT	ARGB8888_IN	VYUY_MODE				RGB1555_BLD	OINTEN								
Default								0								
23	22	21	20	19	18	17	16									
OFIFOSIZE	IN_XDIV_SEL	XDIV_SEL		CONTI	USERXRY			0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8									
OUTMODE	FIFOSIZE		INTEN	YUVTYPE	RESET	START		0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0									
OUTF				INF				0	0	0	0	0	0	0	0	0

Table 2 Description of P_Sca_Con Register

Bit	Function	Type	Description	Condition
31			Reserved.	
30	ARGB8888_OUT	R/W	When this bit is 1 and OUTF is 0xF, the output format will become ARGB8888, otherwise, it is ARGB4444.	0: Disbale ARGB8888 output mode. 1: Enable ARGB8888 output mode.
29	ARGB8888_IN	R/W	When this bit is 1 and INF is 0xF, the input format will become ARGB8888, otherwise, it is ARGB4444.	0: Disbale ARGB8888 input mode. 1: Enable ARGB8888 input mode.
28	VYUY_MODE	R/W	VYUY_MODE enable register, when this bit is 1, the following three modes will be changed to VYUY_MODE. Input mode: 0x5 = UYVY => VYUY 0x6 = YUYV Xx32 => VYUYx32 0x7 = YUYV Xx64 => VYUYx64 Output mode: 0x5 = UYVY => VYUY 0x6 = YUYV Xx32 => VYUYx32 0x7 = YUYV Xx64 => VYUYx64	0: Disable VYUY_MODE 1: Enable VYUY_MODE.
27:26			Reserved.	
25	RGB1555_BLD	R/W	RGB1555 transparent bit enable register. This bit is useful only when RGB1555 mode is selected as input mode.	0: Disable RGB1555 transparent function (bit 15 will always been discarded). 1: Enable RGB1555 transparent function.
24	OINTEN	R/W	Scaler output fifo reach interrupt enable register.	0: Disable interrupt 1: Enable interrupt.
23:22	OFIFOSIZE	R/W	Scaler output data FIFO mode selection. The output FIFO mode can not combine with output format 0x6/0x7/0xE/0xF. The output FIFO does not have the AB FIFO mode, this is for single buffer usage, programmer can done the AB FIFO mode by change the output pointer with software.	0: No output FIFO. 1: 16 lines output FIFO. 2: 32 lines output FIFO. 3: 64 lines output FIFO.

Bit	Function	Type	Description	Condition
21:20	IN_XDIV_SEL	R/W	Scaler X divide input selection. This register is used only when YUYVXX32 and YUYVXX64 mode is selected.	00: YUYV32X32 or YUYV64X64 mode 01: YUYV8x32 or YUYV8x64 mode 10: YUYV16x32 or YUYV16x64 mode. 11: Reserved.
19:18	XDIV_SEL	R/W	Scaler X divide output selection. This register is used only when YUYVXX32 and YUYVXX64 mode is selected.	00: YUYV32X32 or YUYV64X64 mode 01: YUYV8x32 or YUYV8x64 mode 10: YUYV16x32 or YUYV16x64 mode. 11: Reserved.
17	CONTI	R/W	Continuous mode. When user wish to start a previous scaler task, this bit must be set to 1 before set the RESET flag.	0: Normal mode. 1: Previous scaler task mode.
16	USERXY	R/W	Use IN_RX_Width, IN_RY_Width control register.	0: Don't use IN_RX_Width, IN_RY_Width, use IN_X_Width and IN_Y_Width as real input width. 1: Use IN_RX_Width, IN_RY_Width as real input width.
15	OUTMODE	R/W	Scaler out-of-boundary processing mode.	0: Use the boundary data of the input picture as the out-of-boundary. 1: Use color defined in P_Sca_OutBndColor.
14:12	FIFOSIZE	R/W	Scaler input data FIFO mode selection.	0: No AB FIFO, input data is valid for all frame. 1: 2x16 lines AB FIFO. 2: 2x32 lines AB FIFO. 3: 2x64 lines AB FIFO. 4: 2x128 lines AB FIFO. 5: 2x256 lines AB FIFO.
11	INTEN	R/W	Scaler interrupt enable register.	0: Disable interrupt 1: Enable interrupt.



Bit	Function	Type	Description	Condition
10	YUVTYP	R/W	YUV type selection when YUYV or UYVY format is selected.	0: Unsigned (YCbCr) 1: Signed (YUV)
9	RESET	W	Scaler software reset register.	Write 0: No effect. Write 1: Reset Scaler controller.
8	START	R/W	Scaler Start control register, write 1 to this register will start the scaler function.	Write 0: No effect. Write 1: Start Scaler function. Read 0: Scaler is idle. Read 1: Scaler is busy.
7:4	OUTF	R/W	Output format selection.	0x0 = RGB1555 0x1 = RGB565. 0x2 = RGBG. 0x3 = GRGB. 0x4 = YUYV. 0x5 = UYVY. 0x6 = YUYV Xx32 0x7 = YUYV Xx64 0x8 = YUV422 (YUV Separate type) 0x9 = YUV420 (YUV Separate type) 0xA = YUV411 (YUV Separate type) 0xB = YUV444 (YUV Separate type) 0xC = Y only (YUV Separate type) 0xF = ARGB4444 or ARGB8888 mode. Other = Reserved.

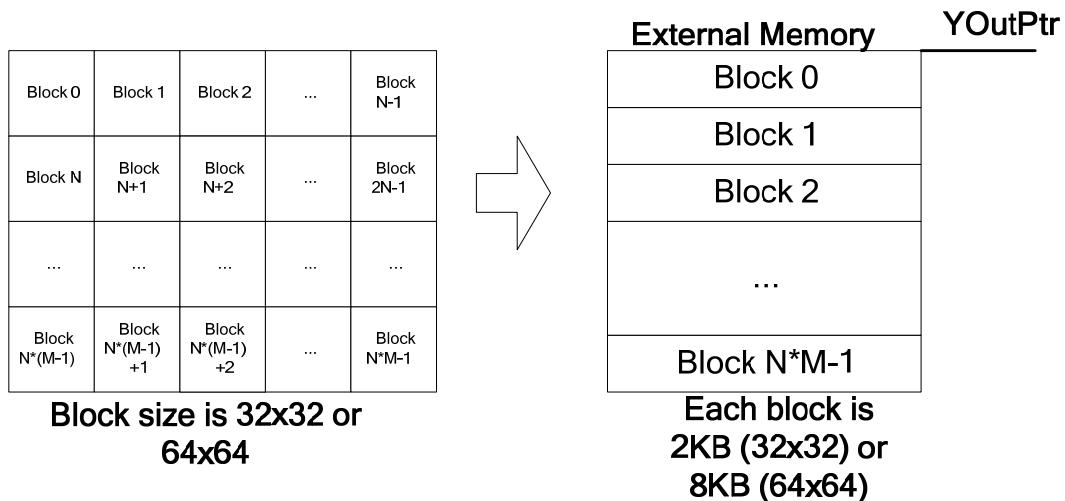
Bit	Function	Type	Description	Condition
3:0	INF	R/W	Input format selection.	0x0 = RGB1555 0x1 = RGB565. 0x2 = RGBG. 0x3 = GRGB. 0x4 = YUYV. 0x5 = UYVY 0x6 = YUYV Xx32 0x7 = YUYV Xx64 0x8 = YUV422 (YUV Separate type) 0x9 = YUV420 (YUV Separate type) 0xA = YUV411 (YUV Separate type) 0xB = YUV444 (YUV Separate type) 0xC = Y only (YUV Separate type) 0xD = YUV422 Vertical Type (YUV Separate type). 0xE = YUV411 Vertical Type (YUV Separate type). 0xF = ARGB4444 or ARGB8888.

The following diagram shows the actual format of frame type on the external RAM.

RGB1555	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 X R _{N+1} G _{N+1} B _{N+1} X R _N G _N B _N
RGB565	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 R _{N+1} G _{N+1} B _{N+1} R _N G _N B _N
RGBG	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 R _{avg} G _{N+1} B _{avg} G _N
GRGB	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 G _{N+1} R _{avg} G _N B _{avg}
YUYV	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Y _{N+1} U _{avg} Y _N V _{avg}
UYVY	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 U _{avg} Y _{N+1} V _{avg} Y _N
VYUY	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 V _{avg} Y _{N+1} U _{avg} Y _N
ARGB4444	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 A _{N+1} R _{N+1} G _{N+1} B _{N+1} A _N R _N G _N B _N
ARGB8888	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 A _N R _N G _N B _N

The following diagram shows the actual format on the external memory of special output format YUYVXX32 or YUYVXX64. **The output width or height does “not” need to be multiply of 32 or 64.**

YUYV32X32	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Y _{N+1} U _{avg} Y _N V _{avg}
YUYV64X64	



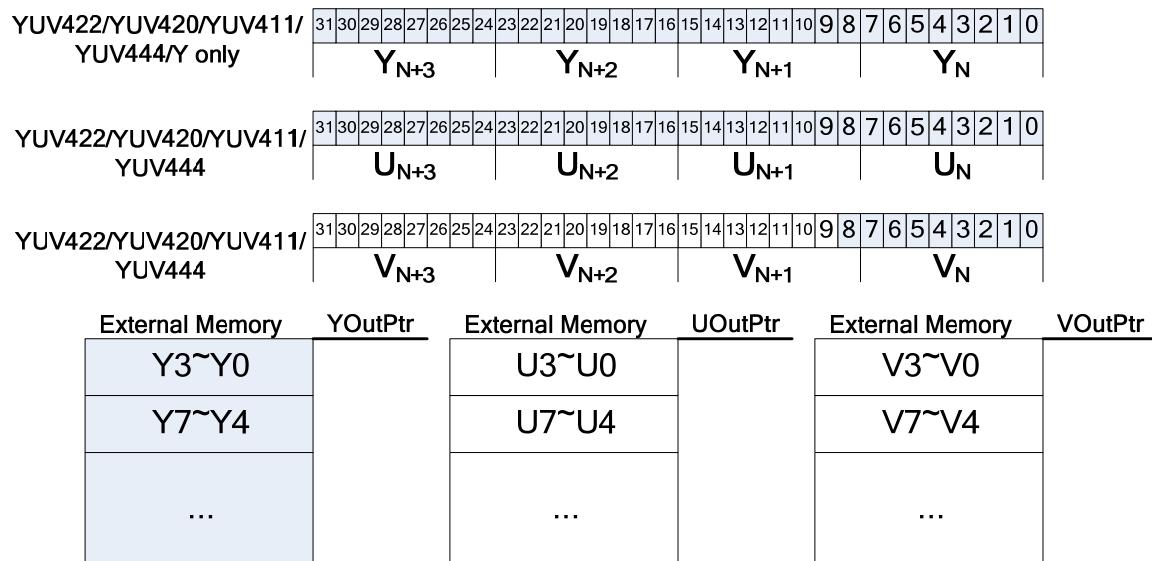
The following table shows the X width selection under YUYVXX32 and YUYVXX64 output mode.

XDIV_SEL[1:0]	OUTF = YUYVXX32	OUTF = YUYVXX64	Maximum output X width.
0b00	32x32	64x64	8191
0b01	8x32	8x64	2047
0b10	16x32	16x64	4095
0b11	Reserved.	Reserved.	Reserved.

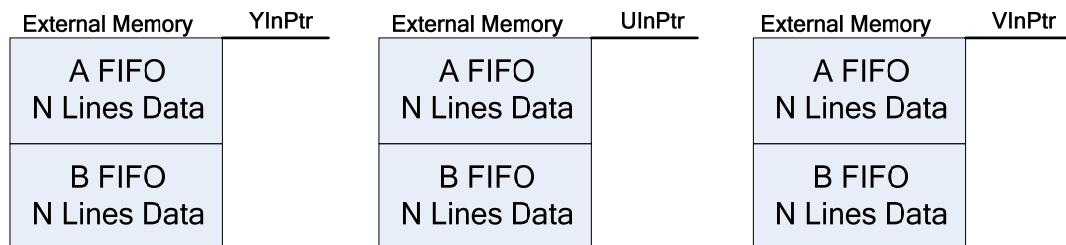
The following table shows the X width selection under YUYVXX32 and YUYVXX64 input mode.

IN_XDIV_SEL[1:0]	INF = YUYVXX32	INF = YUYVXX64	Maximum input X width.
0b00	32x32	64x64	8191
0b01	8x32	8x64	2047
0b10	16x32	16x64	4095
0b11	Reserved.	Reserved.	Reserved.

The following diagram shows the actual format on the external memory for YUV separate format.



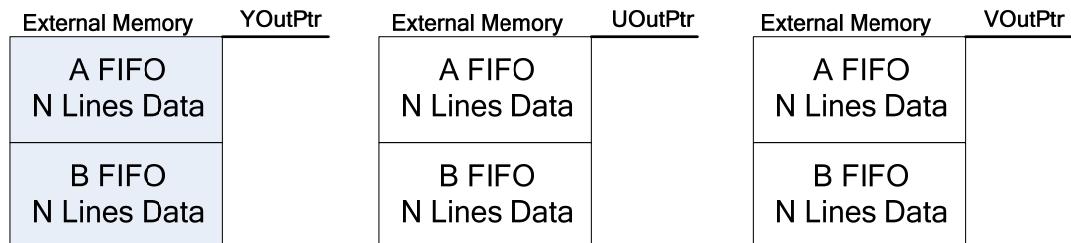
The following diagram shows the data format when input FIFO mode is used.



The following table shows the necessary system memory for different kinds of FIFO size, where W is the width of the input image.

FIFO SIZE	LINES	Input Format					
		RGB1555, RGB565, RGBG, GRGB, YUYV, UYVY, VYURGB4444	ARGB8888	YUV422	YUV420, YUV411	YUV444	YONLY
1	2*16	64 Bytes	128 Bytes	W*32bytes W*16bytes W*16bytes	W*32bytes W*8bytes W*8bytes	W*32bytes W*32bytes W*32bytes	W*32bytes
2	2*32	128 Bytes	256 Bytes	W*64bytes W*32bytes W*32bytes	W*64bytes W*16bytes W*16bytes	W*64bytes W*64bytes W*64bytes	W*64bytes
3	2*64	256 Bytes	512 Bytes	W*128bytes W*64bytes W*64bytes	W*128bytes W*32bytes W*32bytes	W*128bytes W*128bytes W*128bytes	W*128bytes
4	2*128	512 Bytes	1024 Bytes	W*256bytes W*128bytes W*128bytes	W*256bytes W*64bytes W*64bytes	W*256bytes W*256bytes W*256bytes	W*256bytes
5	2*256	W*1024 Bytes	W*2048 Bytes	W*512bytes W*256bytes W*256bytes	W*512bytes W*128bytes W*128bytes	W*512bytes W*512bytes W*512bytes	W*512bytes

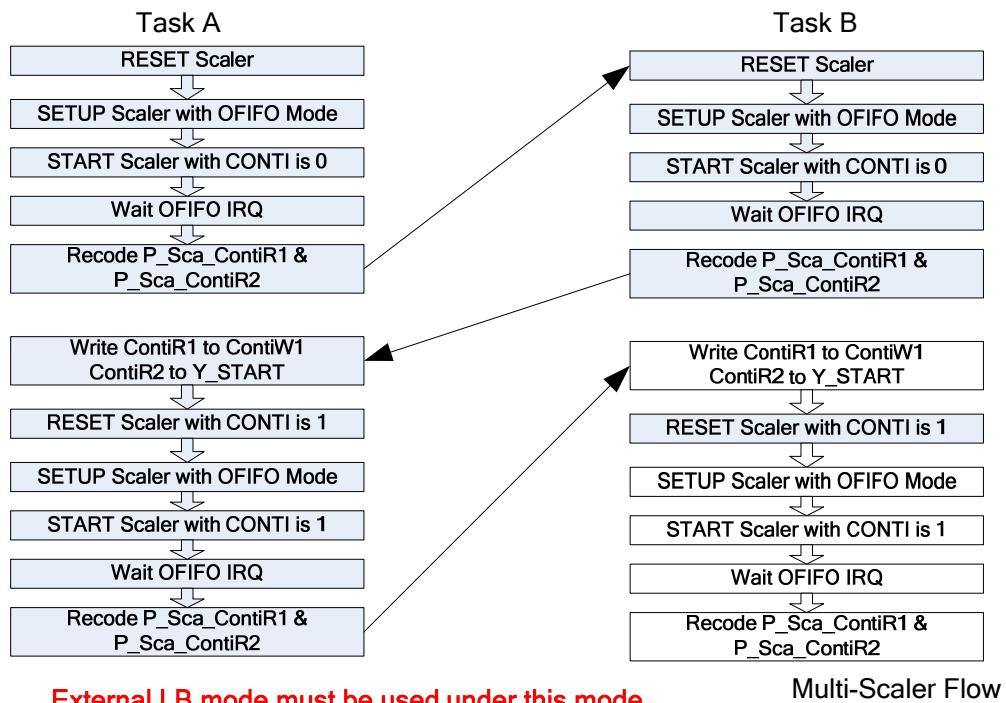
The following diagram shows the data format when output FIFO mode is used. (When YUYVXx32 or YUYV Xx64 mode is selected, the output FIFO mode is not allowed to be used)



The following table shows the necessary system memory for different kinds of output FIFO size, where W is the width of the output image.

OFIFO SIZE	LINES	Output Format					
		RGB1555, RGB565, RGBG, GRGB, YUYV, UYVY, VYUY, ARGB4444	ARGB8888	YUV422	YUV420, YUV411	YUV444	YONLY
1	2*16	W*64 Bytes	W*128 Bytes	Y: W*32bytes U: W*16bytes V: W*16bytes	Y: W*32bytes U: W*8bytes V: W*8bytes	Y: W*32bytes U: W*32bytes V: W*32bytes	Y: W*32bytes
2	2*32	W*128 Bytes	W*256 Bytes	Y: W*64bytes U: W*32bytes V: W*32bytes	Y: W*64bytes U: W*16bytes V: W*16bytes	Y: W*64bytes U: W*64bytes V: W*64bytes	Y: W*64bytes
3	2*64	W*256 Bytes	W*512 Bytes	Y: W*128bytes U: W*64bytes V: W*64bytes	Y: W*128bytes U: W*32bytes V: W*32bytes	Y: W*128bytes U: W*128bytes V: W*128bytes	Y: W*128bytes

If output FIFO is selected and multiple scaler tasks are used, the following procedure must be used to control the scaler engine.



19.5.2 P_SCALER2_OB_COLOR

The P_SCALER2_OB_COLOR registers are used to control scaler output color when out-of-boundary. This register is used only when OUTMODE is 1.

P_SCALER2_

0x93002004

Out-of-Boundary Color Register

OB_COLOR

Bit	23	22	21	20	19	18	17	16
Function	OBColor[23:16]							
Default	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
	OBColor[15:8]							
	1	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	OBColor[7:0]							
	1	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:24			Reserved.	
23:0	OBColor	R/W	Scaler out-of-boundary register. The data format is YCbCr type.	0x008080: Black

19.5.3 P_SCALER2_OUTX_WIDTH

The P_SCALER2_OUTX_WIDTH registers are used to control scaler output resolution at the X axis.

P_SCALER2_OUTX_WIDTH

0x93002008

Scaler X output width Register

Bit	15	14	13	12	11	10	9	8
Function	OutX_Width[12:8]							
Default	0	0	0	0	0	0	1	0
	7	6	5	4	3	2	1	0
	OutX_Width[7:0]							
	0	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
31:13			Reserved.	
12:0	OutX_Width	R/W	Scaler X output width selection. When output width is N, programmer must fill (N-1) to this register.	

The OutX_Width has some limitation depend on the OUTF register. The following table shows the allowed setting of this register under different OUTF.

OUTF[3:0]	Format	Allowed OutX_Width
0x0	RGB1555	(8*N) – 1
0x1	RGB565	(8*N) – 1
0x2	RGBG	(8*N) – 1
0x3	GRGB	(8*N) – 1
0x4	YUYV	(8*N) – 1
0x5	UYVY	(8*N) – 1
0x6	YUYV 32x32	(8*N) – 1
0x7	YUYV 64x64	(8*N) – 1
0x8	YUV422	(16*N) – 1
0x9	YUV420	(16*N) – 1
0xA	YUV411	(32*N) – 1
0xB	YUV444	(16*N) – 1
0xC	Y Only	(8*N) – 1

*N is a positive integer.

19.5.4 P_SCALER2_OUTY_WIDTH

The P_SCALER2_OUTY_WIDTH registers are used to control scaler output resolution at the Y axis.

P_SCALER2_OUTY_WIDTH 0x9300200C Scaler Y output width Register										
Bit	15	14	13	12	11	10	9			
Function				OutY_Width[12:8]						
Default	0	0	0	0	0	0	0	1		
	7	6	5	4	3	2	1	0		
	OutY_Width[7:0]									
	1	1	0	1	1	1	1	1		

Bit	Function	Type	Description	Condition
31:13			Reserved.	
12:0	OutY_Width	R/W	Scaler Y output width selection. When output width is N, programmer must fill (N-1) to this register.	

19.5.5 P_SCALER2_X_FACTOR

The P_SCALER2_X_FACTOR registers are used to control scaler output scale factor at the X axis.

P_SCALER2_X_FACTOR 0x93002010

Scaler X factor Register

Bit	23	22	21	20	19	18	17	16
Function	X_Factor[23:16]							
Default	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
	X_Factor[15:8]							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	X_Factor[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:24			Reserved.	
23:0	X_Factor	R/W	Scaler X scale factor register.	0x10000: Original Size 0x08000: Zoom-in two times. 0x20000: Zoom-out two times.

The value in this register can be calculated by the following equation.

$$X_Factor = (\ln X / \text{Out} X) * 2^{16};$$

19.5.6 P_SCALER2_Y_FACTOR

The P_SCALER2_Y_FACTOR registers are used to control scaler output scale factor at the Y axis.

P_SCALER2_Y_FACTOR 0x93002014

Scaler Y factor Register

Bit	23	22	21	20	19	18	17	16
Function	Y_Factor[23:16]							
Default	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
	Y_Factor[15:8]							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	Y_Factor[7:0]							
	0	0	0	0	0	0	0	0



Bit	Function	Type	Description	Condition
31:24			Reserved.	
23:0	Y_Factor	R/W	Scaler Y scale factor register.	0x10000: Original Size 0x08000: Zoom-in two times. 0x20000: Zoom-out two times.

The value in this register can be calculated by the following equation.

Y_Factor = (In_W/Out_W) * 2^16;

19.5.7 P_SCALER2_X START

The P_SCALER2_X_START registers are used to control scaler internal accumulation initial value at the X axis. This register is used to control the input start point on X-axis.

P_SCALER2_X_START 0x93002018 Scaler X start Register

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
<u>X_Start[7:0]</u>							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:30			Reserved.	
29:0	X_Start	R/W	Scaler X internal accumulator initial value register.	

19.5.8 P_SCALER2_Y_START

The P_SCALER2_Y_START registers are used to control scaler internal accumulation initial value at the Y axis. This register is used to control the input start point on Y-axis. **This register should be set to 0 when FIFO mode is used.**

P_SCALER2_Y_START 0x9300201C Scaler Y start Register

23	22	21	20	19	18	17	16	
Y_Start[23:16]								
0	0	0	0	0	0	0	0	0
Y_Start[15:8]								
0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	
Y_Start[7:0]								
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:30			Reserved.	
29:0	Y_Start	R/W	Scaler Y internal accumulator initial value register.	

19.5.9 P_SCALER2_IN_X_Width

The P_SCALER2_IN_X_Width registers are used to control scaler input resolution at the X axis.

P_SCALER2_IN_X_Width 0x93002020

Scaler X input width Register

Bit	15	14	13	12	11	10	9	8	
InX_Width[12:8]									
Function	0	0	0	0	0	0	1	0	
InX_Width[7:0]									
Default	0	1	1	1	1	1	1	1	
	7	6	5	4	3	2	1	0	
	InX_Width[7:0]								
	0	1	1	1	1	1	1	1	

Bit	Function	Type	Description	Condition
31:13			Reserved.	
12:0	InX_Width	R/W	Scaler X input width selection. When input width is N, programmer must fill (N-1) to this register.	

19.5.10P_SCALER2_IN_Y_Width

The P_SCALER2_IN_Y_Width registers are used to control scaler input resolution at the Y axis.

P_SCALER2_IN_Y_Width 0x93002024

Scaler Y input width Register

Bit	15	14	13	12	11	10	9	8
Function	InY_Width[12:8]							
Default	0	0	0	0	0	0	0	1

7	6	5	4	3	2	1	0	1
InY_Width[7:0]								

1	1	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

Bit	Function	Type	Description						Condition
31:13			Reserved.						
12:0	InY_Width	R/W	Scaler Y input width selection. When input width is N, programmer must fill (N-1) to this register.						

19.5.11P_SCALER2_Y_INPTR

The P_SCALER2_Y_INPTR registers are used to control scaler input pointer of Y buffer.

P_SCALER2_Y_INPTR 0x93002028

Scaler Y input pointer Register

Bit	31	30	29	28	27	26	25	24
Function	YInPtr[31:24]							
Default	0	0	0	0	0	0	0	1

23	22	21	20	19	18	17	16	
YInPtr[23:16]								

0	0	0	0	0	0	0	0	
YInPtr[15:8]								

0	0	0	0	0	0	0	0	
YInPtr[7:0]								

0	0	0	0	0	0	0	0	
---	---	---	---	---	---	---	---	--

Bit	Function	Type	Description						Condition
31:0	YInPtr	R/W	Scaler input buffer pointer for Y. This value must be 4 bytes align.						

19.5.12P_SCALER2_U_INPTR

The P_SCALER2_U_INPTR registers are used to control scaler input pointer of U buffer.

P_SCALER2_U_INPTR 0x9300202C Scaler U input pointer Register

Bit	31	30	29	28	27	26	25	24
Function	UInPtr[31:24]							
Default	0	0	0	0	0	0	0	1
	23	22	21	20	19	18	17	16
	UInPtr[23:16]							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	UInPtr[15:8]							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	UInPtr[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:0	UInPtr	R/W	Scaler input buffer pointer for U. This value must be 4 bytes align.	

19.5.13P_SCALER2_V_INPTR

The P_SCALER2_V_INPTR registers are used to control scaler input pointer of V buffer.

P_SCALER2_V_INPTR 0x93002030 Scaler V input pointer Register

Bit	31	30	29	28	27	26	25	24
Function	VInPtr[31:24]							
Default	0	0	0	0	0	0	0	1
	23	22	21	20	19	18	17	16
	VInPtr[23:16]							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	VInPtr[15:8]							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	VInPtr[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:0	VInPtr	R/W	Scaler input buffer pointer for V. This value must be 4 bytes align.	

For different kind of input format, different input pointer is used as the following table.

INF[3:0]	Format	Necessary Pointer
0x0	RGB1555	YInPtr for RGB1555 start address.
0x1	RGB565	YInPtr for RGB565 start address.
0x2	RGBG	YInPtr for RGBG start address.
0x3	GRGB	YInPtr for GRGB start address.
0x4	YUYV	YInPtr for YUYV start address.
0x5	UYVY	YInPtr for UYVY start address.
0x8	YUV422	YInPtr for Y start address UInPtr for U start address VInPtr for V start address
0x9	YUV420	YInPtr for Y start address UInPtr for U start address VInPtr for V start address
0xA	YUV411	YInPtr for Y start address UInPtr for U start address VInPtr for V start address
0xB	YUV444	YInPtr for Y start address UInPtr for U start address VInPtr for V start address
0xC	Y Only	YInPtr for Y start address.

19.5.14P_SCALER2_Y_OUTPTR

The P_SCALER2_Y_OUTPTR registers are used to control scaler output pointer of Y buffer.

P_SCALER2_Y_OUTPTR 0x93002034								Scaler Y output pointer Register								
Bit	31	30	29	28	27	26	25	24								
Function	YOutPtr[31:24]															
Default	0	0	0	0	0	0	0	1	23	22	21	20	19	18	17	16
									YOutPtr[23:16]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	YOutPtr[15:8]							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
YOutPtr[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:0	YOutPtr	R/W	Scaler Output buffer pointer for Y. This value must be 4 bytes align.	

19.5.15P_SCALER2_U_OUTPTR

The P_SCALER2_U_OUTPTR registers are used to control scaler output pointer of U buffer.

P_SCALER2_U_OUTPTR 0x93002038

Scaler U output pointer Register

Bit	31	30	29	28	27	26	25	24
UOutPtr[31:24]								
Default	0	0	0	0	0	0	0	1

23	22	21	20	19	18	17	16
UOutPtr[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
UOutPtr[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
UOutPtr[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:0	UOutPtr	R/W	Scaler Output buffer pointer for U. This value must be 4 bytes align.	

19.5.16P_SCALER2_V_OUTPTR

The P_SCALER2_V_OUTPTR registers are used to control scaler output pointer of V buffer.

P_SCALER2_V_OUTPTR 0x9300203C

Scaler V output pointer Register

Bit	31	30	29	28	27	26	25	24
VOutPtr[31:24]								
Default	0	0	0	0	0	0	0	1

23	22	21	20	19	18	17	16	
VOutPtr[23:16]								
0	0	0	0	0	0	0	0	0
VOutPtr[15:8]								
0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	
VOutPtr[7:0]								
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:0	VOutPtr	R/W	Scaler Output buffer pointer for V. This value must be 4 bytes align.	

For different kind of output format, different output pointer is used as the following table.

OUTF[3:0]	Format	Necessary Pointer
0x0	RGB1555	YOutPtr for RGB1555 start address.
0x1	RGB565	YOutPtr for RGB565 start address.
0x2	RGBG	YOutPtr for RGBG start address.
0x3	GRGB	YOutPtr for GRGB start address.
0x4	YUYV	YOutPtr for YUYV start address.
0x5	UYVY	YOutPtr for UYVY start address.
0x6	YUYV 32x32	YOutPtr for YUYV start address.
0x7	YUYV 64x64	YOutPtr for YUYV start address.
0x8	YUV422	YOutPtr for Y start address UOutPtr for U start address VOutPtr for V start address
0x9	YUV420	YOutPtr for Y start address UOutPtr for U start address VOutPtr for V start address
0xA	YUV411	YOutPtr for Y start address UOutPtr for U start address VOutPtr for V start address
0xB	YUV444	YOutPtr for Y start address UOutPtr for U start address VOutPtr for V start address
0xC	Y Only	YOutPtr for Y start address.

19.5.17P_SCALER2_CUR_LINE

The P_SCALER2_CUR_LINE registers represent the current process line.

P_SCALER2_CUR_LINE 0x93002024

Scaler Y input width Register

Bit	15	14	13	12	11	10	9	8
Function	CurLine[12:8]							

Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0

CurLine[7:0]
0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
31:13			Reserved.	
12:0	CurLine	R	Indicate current process line.	

19.5.18P_SCALER2_A11

The P_SCALER2_A11 register is used to control scaler color matrix A11 parameter.

P_SCALER2_A11 0x93002044

Scaler A11 Register

Bit	15	14	13	12	11	10	9	8
Function	A11[9:8]							

Default	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0

A11[7:0]
0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A11	R/C	Scaler color matrix A11 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

19.5.19P_SCALER2_A12

The P_SCALER2_A12 register is used to control scaler color matrix A12 parameter.

P_SCALER2_A12 0x93002048 Scaler A12 Register							
Bit	15	14	13	12	11	10	9 8
Function							A12[9:8]
Default	0	0	0	0	0	0	0
	7	6	5	4	3	2	1 0
	0	0	0	0	0	0	0
	A12[7:0]						
	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A12	R/C	Scaler color matrix A12 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

19.5.20P_SCALER2_A13

The P_SCALER2_A13 register is used to control scaler color matrix A13 parameter.

P_SCALER2_A13 0x9300204C Scaler A13 Register							
Bit	15	14	13	12	11	10	9 8
Function							A13[9:8]
Default	0	0	0	0	0	0	0
	7	6	5	4	3	2	1 0
	0	0	0	0	0	0	0
	A13[7:0]						
	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A13	R/C	Scaler color matrix A13 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

19.5.21P_SCALER2_A21

The P_SCALER2_A21 register is used to control scaler color matrix A21 parameter.

P_SCALER2_A12 0x93002050 Scaler A21 Register								
Bit	15	14	13	12	11	10	9	8
Function							A21[9:8]	
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
	A21[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A21	R/C	Scaler color matrix A21 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

19.5.22P_SCALER2_A22

The P_SCALER2_A22 register is used to control scaler color matrix A22 parameter.

P_SCALER2_A22 0x93002054 Scaler A22 Register								
Bit	15	14	13	12	11	10	9	8
Function							A22[9:8]	
Default	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
	A22[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A22	R/C	Scaler color matrix A22 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

19.5.23P_SCALER2_A23

The P_SCALER2_A23 register is used to control scaler color matrix A23 parameter.

P_SCALER2_A23 0x93002058								Scaler A23 Register								
Bit	15	14	13	12	11	10	9	8	A23[9:8]							
Function									A23[7:0]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0		0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A23	R/C	Scaler color matrix A22 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

19.5.24P_SCALER2_A31

The P_SCALER2_A31 register is used to control scaler color matrix A31 parameter.

P_SCALER2_A31 0x9300205C								Scaler A31 Register								
Bit	15	14	13	12	11	10	9	8	A31[9:8]							
Function									A31[7:0]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0		0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A31	R/C	Scaler color matrix A31 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

19.5.25P_SCALER2_A32

The P_SCALER2_A32 register is used to control scaler color matrix A32 parameter.

P_SCALER2_A32 0x93002060								Scaler A32 Register								
Bit	15	14	13	12	11	10	9	8	A32[9:8]							
Function									A32[7:0]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A32	R/C	Scaler color matrix A32 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

19.5.26P_SCALER2_A33

The P_SCALER2_A33 register is used to control scaler color matrix A33 parameter.

P_SCALER2_A33 0x93002064								Scaler A33 Register								
Bit	15	14	13	12	11	10	9	8	A33[9:8]							
Function									A33[7:0]							
Default	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:10			Reserved.	
9:0	A33	R/C	Scaler color matrix A33 register. This is a 10 bits signed value.	0x000: 0.00 0x080: 0.50 0x380: -0.50 0x100: 1.00 0x300: -1.00

The following matrix will be applied to the YUV data when the CM_EN (0x0080 bit 2) is set to 1.

$$\begin{bmatrix} Y' \\ U' \\ V' \end{bmatrix} = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix} * \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

Where the U/V is -128~127 signed value. The YUV data here is the result of scaler + Y gamma process.

19.5.27P_SCALER2_IN_RX_WIDTH

The P_SCALER2_IN_RX_WIDTH registers are used to control scaler real input resolution at the X axis. This register is used when the JPEG's real X size does not match the input X size, i.e. with some padding data.

P_SCALER2_IN_RX_WIDTH 0x93002068 Scaler X real input width Register

Bit	15	14	13	12	11	10	9	8
Function	InRX_Width[12:8]							
Default	0	0	0	0	0	0	1	0

7	6	5	4	3	2	1	0
InRX_Width[7:0]							
0	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
31:13			Reserved.	
12:0	InRX_Width	R/W	Scaler X real input width selection. When real input width is N, programmer must fill (N-1) to this register.	0: Disable this function, use InX_Width as real input X width. Other: Valid X width.

19.5.28P_SCALER2_IN_RY_WIDTH

The P_SCALER2_IN_RY_WIDTH registers are used to control scaler real input resolution at the Y axis. This register is used when the JPEG's real Y size does not match the input X size, i.e. with some padding data.

P_SCALER2_IN_RY_WIDTH 0x9300206C Scaler Y real input width Register

Bit	15	14	13	12	11	10	9	8
Function	InRY_Width[12:8]							
Default	0	0	0	0	0	0	0	1

7	6	5	4	3	2	1	0
InRY_Width[7:0]							
1	1	0	1	1	1	1	1

Bit	Function	Type	Description	Condition
31:13			Reserved.	
12:0	InRY_Width	R/W	Scaler Y real input width selection. When real input width is N, programmer must fill (N-1) to this register.	0: Disable this function, use InY_Width as real input Y width. Other: Valid Y width.

19.5.29P_SCALER2_OUT_OFFSET

The P_SCALER2_OUT_OFFSET registers are used to control scaler output horizontal offset when the actual frame buffer is larger than the OUT_X_WIDTH. This is usefull when you only wish to update part of the frame buffer.

P_SCALER2_OUT_OFFSET 0x93002070 Scaler output offset Register										
Bit	15	14	13	12	11	10	9			
Function				Out_Offset[12:8]						
Default	0	0	0	0	0	0	0	0		
Out_Offset [7:3]										
	7	6	5	4	3	2	1	0		
	0	0	0	0	0	0	0	0		

Bit	Function	Type	Description	Condition
31:13			Reserved.	
12:0	Out_Offset	R/W	Scaler horizontal offset register. This register is used when the real buffer size is larger than OUT_X_WIDTH. For example, when the frame buffer width is 640 and OUT_X_WIDTH is 320, programmer only need to write 320 to this register, then the scaler engine will output to the 640 frame buffer with 320 pixels offset at end of every line. This value should be multiple of 8.	0: No offset. 8: 8 pixels offset.

19.5.30P_SCALER2_LB_PTR

The P_SCALER2_LB_PTR registers are used to control scaler external line buffer location. The size of external line buffer is depending on the setting mode of the EXT_LB_MODE. If programmer wish to use the external line buffer mode, this register should be set correctly before turn on the scaler. We suggest use the internal RAM as the line buffer to save the bandwidth and maximize the system performance.

P_SCALER2_LB_PTR 0x93002074 Scaler LB pointer Register							
Bit	31	30	29	28	27	26	25
Function	LBPtr[31:24]						
Default	0	0	0	0	0	0	1

23	22	21	20	19	18	17	16	
LBPtr[23:16]								
0	0	0	0	0	0	0	0	0
LBPtr[15:8]								
0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	
LBPtr[7:0]								
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:0	LBPtr	R/W	Scaler external line buffer pointer. This value must be 4 bytes align.	

19.5.31P_SCALER2_INT

The P_SCALER2_INT register is used to control scaler interrupt.

P_SCALER2_INT 0x9300207C Scaler Interrupt Register								
Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
						SCAOUTINT	DONE	SCAINT
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:2			Reserved.	
2	SCAOUTINT	R/C	Scaler output FIFO mode interrupt register. When the output FIFO reach the setting value, this value will be set.	Read 0: Interrupt not happen Read 1: Interrupt happen. Write 0: No effect. Write 1: Clear this bit.
1	DONE	R	Scaler Done status flag, this bit will indicate if the whole scaler process is complete. This bit will be cleared after SCAINT is cleared.	0: The SCAINT is only FIFO's interrupt. 1: The SCAINT is indicate the whole scaler process is complete.
0	SCAINT	R/C	Scaler interrupt register. When scaler completes the scale job, this bit will be set to 1. Write 1 to clear this bit.	Read 0: Interrupt not happen Read 1: Interrupt happen. Write 0: No effect. Write 1: Clear this bit.

19.5.32P_SCALER2_POST

The P_SCALER2_POST register is used to control post effect of scaler's output interrupt.

P_SCALER2_POST 0x93002080

Scaler Post Effect Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
		EXT_LB			CM_EN	YGAMMA	COUNT

0 0 0 0 0 0 0 1

Bit	Function	Type	Description	Condition
31:6			Reserved.	
[5:4]	EXT_LB	R/W	External line buffer control.	0: Internal Line buffer mode 1: 1024 internal line buffer + external Line buffer mode. 2: All external line buffer mode.
3			Reserved.	
2	CM_EN	R/W	Color matrix function enable register.	0: Disable color matrix function. 1: Enable color matrix function.
1	YGAMMA	R/W	Y gamma function enable register.	0: Disable Y gamma function. 1: Enable Y gamma function.
0	COUNT	R/C	Scaler auto calculate function control register. This register is used to turn-on the YUV max/min/sum function and Y histogram function.	0: Disable count function. 1: Enable count function.

19.5.33P_SCALER2_MAX_Y

The P_SCALER2_MAX_Y register is used to read the maximum Y value of scaler's output.

P_SCALER2_MAX_Y 0x93002084

Scaler Maximum Y Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
MaxY[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:8			Reserved.	
7:0	MaxY	R	Maximum Y of the scaler's output. This value will be reset to 0x00 if the RESET bit is set. This value will be valid after the whole picture is processed.	

19.5.34P_SCALER2_MIN_Y

The P_SCALER2_MIN_Y register is used to read the minimum Y value of scaler's output.

P_SCALER2_MIN_Y 0x93002088

Scaler Minimum Y Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
MinY[7:0]							
1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
31:8			Reserved.	
7:0	MinY	R	Minimum Y of the scaler's output. This value will be reset to 0xFF if the RESET bit is set. This value will be valid after the whole picture is processed.	

19.5.35P_SCALER2_SUM_Y

The P_SCALER2_SUM_Y register is used to read the summation of Y value of scaler's output.

P_SCALER2_SUM_Y 0x9300208C

Scaler Y Summation Register

Bit	31	30	29	28	27	26	25	24
Function	SumY[27:16]							
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	SumY[23:16]							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	SumY[15:8]							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	SumY[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:28			Reserved.	
27:0	SumY	R	Y summation of the scaler's output. This value will be reset to 0x00 if the RESET bit is set. This value will be valid after the whole picture is processed. This value dose not have overflow control, so care must be taken when the output size is too huge.	

19.5.36P_SCALER2_MAX_U

The P_SCALER2_MAX_U register is used to read the maximum U value of scaler's output.

P_SCALER2_MAX_U 0x93002090

Scaler Maximum U Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	MaxU[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:8			Reserved.	
7:0	MaxU	R	Maximum U of the scaler's output. This value will be reset to 0x00 if the RESET bit is set. This value will be valid after the whole picture is processed. This is an 8 bits unsigned value.	

19.5.37P_SCALER2_MIN_U

The P_SCALER2_MIN_U register is used to read the minimum U value of scaler's output.

P_SCALER2_MIN_U 0x93002094

Scaler Minimum U Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
MinU[7:0]							
1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
31:8			Reserved.	
7:0	MinU	R	Minimum U of the scaler's output. This value will be reset to 0xFF if the RESET bit is set. This value will be valid after the whole picture is processed.	

19.5.38P_SCALER2_SUM_U

The P_SCALER2_SUM_U register is used to read the summation of U value of scaler's output.

P_SCALER2_SUM_U 0x93002098

Scaler U Summation Register

Bit	31	30	29	28	27	26	25	24
Function								
SumU[27:16]								

23	22	21	20	19	18	17	16
SumU[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SumU[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SumU[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:28			Reserved.	
27:0	SumU	R	U summation of the scaler's output. This value will be reset to 0x00 if the RESET bit is set. This value will be valid after the whole picture is processed. This value does not have overflow control, so care must be taken when the output size is too huge. This is a 28 bits signed value.	

19.5.39P_SCALER2_MAX_V

The P_SCALER2_MAX_V register is used to read the maximum V value of scaler's output.

P_SCALER2_MAX_V 0x9300209C

Scaler Maximum V Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
MaxV[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:8			Reserved.	
7:0	MaxV	R	Maximum V of the scaler's output. This value will be reset to 0x00 if the RESET bit is set. This value will be valid after the whole picture is processed. This is an 8 bits unsigned value.	

19.5.40P_SCALER2_MIN_V

The P_SCALER2_MIN_V register is used to read the minimum V value of scaler's output.

P_SCALER2_MIN_V 0x930020A0

Scaler Minimum V Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
MinV[7:0]							
1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
31:8			Reserved.	
7:0	MinV	R	Minimum V of the scaler's output. This value will be reset to 0xFF if the RESET bit is set. This value will be valid after the whole picture is processed.	

19.5.41P_SCALER2_SUM_V

The P_SCALER2_SUM_V register is used to read the summation of V value of scaler's output.

P_SCALER2_SUM_V 0x930020A4

Scaler V Summation Register

Bit	31	30	29	28	27	26	25	24
Function					SumV[27:16]			
Default	0	0	0	0	0	0	0	0
SumV[23:16]								
0	0	0	0	0	0	0	0	0
SumV[15:8]								
0	0	0	0	0	0	0	0	0
SumV[7:0]								
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:28			Reserved.	
27:0	SumV	R	V summation of the scaler's output. This value will be reset to 0x00 if the RESET bit is set. This value will be valid after the whole picture is processed. This value does not have overflow control, so care must be taken when the output size is too huge. This is a 28 bits signed value.	

19.5.42P_SCALER2_Y_HIS0

The P_SCALER2_Y_HIS0 register is used to read the Y0~Y15 histogram of scaler's output.

P_SCALER2_Y_HIS0 0x930020C0

Scaler Y Histogram 0 Register

Bit	23	22	21	20	19	18	17	16
Function					Y_His0[19:16]			
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Y_His0[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Y_His0[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His0	R	Y histogram 0 of scaler's output. This register store the number of Y in range 0~15. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF is the Y in this range is too much.	

19.5.43P_SCALER2_Y_HIS1

The P_SCALER2_Y_HIS1register is used to read the Y16~Y31 histogram of scaler's output.

P_SCALER2_Y_HIS1 0x930020C4

Scaler Y Histogram 1 Register

Bit	23	22	21	20	19	18	17	16
Y_His1[19:16]								

Default 0 0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
Y_His1[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Y_His1[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His1	R	Y histogram 1 of scaler's output. This register store the number of Y in range 16~31. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.	

19.5.44P_SCALER2_Y_HIS2

The P_SCALER2_Y_HIS2 register is used to read the Y32~Y47 histogram of scaler's output.

P_SCALER2_Y_HIS2 0x930020C8

Scaler Y Histogram 2 Register

Bit	23	22	21	20	19	18	17	16
Function	Y_His2[19:16]							
Default	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Y_His2[15:8]								
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Y_His2[7:0]								
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His2	R	Y histogram 2 of scaler's output. This register store the number of Y in range 32~47. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.	

19.5.45P_SCALER2_Y_HIS3

The P_SCALER2_Y_HIS3 register is used to read the Y48~Y63 histogram of scaler's output.

P_SCALER2_Y_HIS3 0x930020CC

Scaler Y Histogram 3 Register

Bit	23	22	21	20	19	18	17	16
Function	Y_His3[19:16]							
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Y_His3[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Y_His3[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
31:20			Reserved.					
19:0	Y_His3	R	Y histogram 3 of scaler's output. This register store the number of Y in range 48~63. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.					

19.5.46P_SCALER2_Y_HIS4

The P_SCALER2_Y_HIS4 register is used to read the Y64~Y79 histogram of scaler's output.

P_SCALER2_Y_HIS4 0x930020D0

Scaler Y Histogram 4 Register

Bit	23	22	21	20	19	18	17	16
Function								Y_His4[19:16]
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Y_His4[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Y_His4[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
31:20			Reserved.					
19:0	Y_His4	R	Y histogram 4 of scaler's output. This register store the number of Y in range 64~79. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.					

19.5.47P_SCALER2_Y_HIS5

The P_SCALER2_Y_HIS5 register is used to read the Y80~Y95 histogram of scaler's output.

P_SCALER2_Y_HIS5 0x930020D4

Scaler Y Histogram 5 Register

Bit	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----

Function **Y_His5[19:16]**

Default 0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
----	----	----	----	----	----	---	---

Y_His5[15:8]

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

Y_His5[7:0]

0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His5	R	Y histogram 5 of scaler's output. This register store the number of Y in range 80~95. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.	

19.5.48P_SCALER2_Y_HIS6

The P_SCALER2_Y_HIS6 register is used to read the Y96~Y111 histogram of scaler's output.

P_SCALER2_Y_HIS6 0x930020D8

Scaler Y Histogram 6 Register

Bit	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----

Function **Y_His6[19:16]**

Default 0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
----	----	----	----	----	----	---	---

Y_His6[15:8]

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

Y_His6[7:0]

0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His6	R	Y histogram 6 of scaler's output. This register store the number of Y in range 96~111. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.	

19.5.49P_SCALER2_Y_HIS7

The P_SCALER2_Y_HIS7 register is used to read the Y112~Y127 histogram of scaler's output.

P_SCALER2_Y_HIS7 0x930020DC

Scaler Y Histogram 7 Register

Bit	23	22	21	20	19	18	17	16
Function	Y_His7[19:16]							
Default	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	Y_His7[15:8]							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	Y_His7[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His7	R	Y histogram 7 of scaler's output. This register store the number of Y in range 112~127. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.	

19.5.50P_SCALER2_Y_HIS8

The P_Sca_Y_His8 register is used to read the Y128~Y143 histogram of scaler's output.

P_SCALER2_Y_HIS8 0x930020E0

Scaler Y Histogram 8 Register

Bit	23	22	21	20	19	18	17	16
Function	Y_His8[19:16]							
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Y_His8[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Y_His8[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
31:20			Reserved.					
19:0	Y_His8	R	Y histogram 8 of scaler's output. This register store the number of Y in range 128~143. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.					

19.5.51P_SCALER2_Y_HIS9

The P_SCALER2_Y_HIS9 register is used to read the Y144~Y159 histogram of scaler's output.

P_SCALER2_Y_HIS9 0x930020E4

Scaler Y Histogram 9 Register

Bit	23	22	21	20	19	18	17	16
Function								Y_His9[19:16]
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Y_His9[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Y_His9[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
31:20			Reserved.					
19:0	Y_His9	R	Y histogram 9 of scaler's output. This register store the number of Y in range 144~159. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.					

19.5.52P_SCALER2_Y_HIS10

The P_SCALER2_Y_HIS10 register is used to read the 160~Y175 histogram of scaler's output.

P_SCALER2_Y_HIS10 0x930020E8

Scaler Y Histogram 10 Register

Bit	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----

Function **Y_His10[19:16]**

Default 0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
----	----	----	----	----	----	---	---

Y_His10[15:8]

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

Y_His10[7:0]

0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His10	R	Y histogram 10 of scaler's output. This register store the number of Y in range 160~175. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.	

19.5.53P_SCALER2_Y_HIS11

The P_SCALER2_Y_HIS11 register is used to read the Y176~Y191 histogram of scaler's output.

P_SCALER2_Y_HIS11 0x930020EC

Scaler Y Histogram 11 Register

Bit	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----

Function **Y_His11[19:16]**

Default 0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
----	----	----	----	----	----	---	---

Y_His11[15:8]

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

Y_His11[7:0]

0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His11	R	Y histogram 11 of scaler's output. This register store the number of Y in range 176~191. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.	

19.5.54P_SCALER2_Y_HIS12

The P_SCALER2_Y_HIS12 register is used to read the Y192~Y207 histogram of scaler's output.

P_SCALER2_Y_HIS12 0x930020F0

Scaler Y Histogram 12 Register

Bit	23	22	21	20	19	18	17	16
Function	Y_His12[19:16]							
Default	0	0	0	0	0	0	0	0
<hr/>								
	15	14	13	12	11	10	9	8
Y_His12[15:8]								
	0	0	0	0	0	0	0	0
<hr/>								
	7	6	5	4	3	2	1	0
Y_His12[7:0]								
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His12	R	Y histogram 12 of scaler's output. This register store the number of Y in range 192~207. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this range is too much.	

19.5.55P_SCALER2_Y_HIS13

The P_SCALER2_Y_HIS13 register is used to read the Y208~Y223 histogram of scaler's output.

P_SCALER2_Y_HIS13 0x930020F4

Scaler Y Histogram 13 Register

Bit	23	22	21	20	19	18	17	16
Function	Y_His13[19:16]							
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Y_His13[15:8]							

0	0	0	0	0	0	0	0
Y_His13[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His13	R	Y histogram 13 of scaler's output. This register store the number of Y in range 208~223. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF is the Y in this range is too much.	

19.5.56P_SCALER2_Y_HIS14

The P_SCALER2_Y_HIS14 register is used to read the Y224~Y239 histogram of scaler's output.

P_SCALER2_Y_HIS14 0x930020F8

Scaler Y Histogram 14 Register

Bit	23	22	21	20	19	18	17	16
Function	Y_His14[19:16]							
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
Y_His14[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Y_His14[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His14	R	Y histogram 14 of scaler's output. This register store the number of Y in range 224~239. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF is the Y in this range is too much.	

19.5.57P_SCALER2_Y_HIS15

The P_SCALER2_Y_HIS15 register is used to read the Y240~Y255 histogram of scaler's output.

P_SCALER2_Y_HIS15 0x930020FC Scaler Y Histogram 225 Register								
Bit	23	22	21	20	19	18	17	16
Function	Y_His15[19:16]							
Default	0	0	0	0	0	0	0	0
Y_His15[15:8]								
0 0 0 0 0 0 0 0 0								
Y_His15[7:0]								
0 0 0 0 0 0 0 0 0								

Bit	Function	Type	Description	Condition
31:20			Reserved.	
19:0	Y_His15	R	Y histogram 15 of scaler's output. This register store the number of Y in range 240~255. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF is the Y in this range is too much.	

19.5.58P_SCALER2_CONTI_W1

The P_SCALER2_CONTI_W1 registers are used to write the internal message back to scaler engine when doing the multi task switch. The value of this register will be updated into the scaler engine when CONTI is set to 1 and RESET flag is set.

P_SCALER2_CONTI_W1 0x93002100 Scaler Continuous W1 Register								
Bit	31	30	29	28	27	26	25	24
Function	ContiW1[31:24]							
Default	0	0	0	0	0	0	0	1
ContiW1[23:16]								
0 0 0 0 0 0 0 0 0								
ContiW1[15:8]								
0 0 0 0 0 0 0 0 0								

7	6	5	4	3	2	1	0
ContiW1[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:0	ContiW1	R/W	Scaler internal status write back register 1. The value of this register should be read form ContiR1 register.	

19.5.59P_SCALER2_CONTI_R1

The P_SCALER2_CONTI_R1 registers are used to represent the internal message of scaler engine when doing the multi task switch.

P_SCALER2_CONTI_R1 0x93002104								Scaler Continuous R1 Register								
Bit	31	30	29	28	27	26	25	24								
Function	ContiR1[31:24]															
Default	0	0	0	0	0	0	0	1								
23	22	21	20	19	18	17	16									
0	0	0	0	0	0	0	0									
15	14	13	12	11	10	9	8									
0	0	0	0	0	0	0	0									
7	6	5	4	3	2	1	0									
0	0	0	0	0	0	0	0									

Bit	Function	Type	Description	Condition
31:0	ContiR1	R/W	Scaler internal status read register 1. The value of this register should be read-out when a task is end and write back to ContiW1 register when a task is re-start.	

19.5.60P_SCALER2_CONTI_R2

The P_SCALER2_CONTI_R2 registers are used to represent the internal message of scaler engine when doing the multi task switch.

P_SCALER2_CONTI_R2 0x93002108

Scaler Continuous R2 Register

Bit	31	30	29	28	27	26	25	24
Function	ContiR2[31:24]							
Default	0	0	0	0	0	0	0	1
	23	22	21	20	19	18	17	16
	ContiR2[23:16]							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	ContiR2[15:8]							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	ContiR2[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:0	ContiR1	R/W	Scaler internal status read register 2. The value of this register should be read-out when a task is end and write back to Y_Start register when a task is re-start.	

19.5.61P_SCALER2_Y_GAMMA 0~ P_SCALER2_Y_GAMMA 255

The P_Sca_Y_GammaX register is used to control the gamma curve of scaler's output.

P_SCALER2_Y_GAMMAX 0x93002400 + 4*X

Scaler Y Gamma Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	Y_GammaX[7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31:8			Reserved.	
7:0	Y_GammaX	R/W	The Y gamma value of Y with value X. The default value is X. For example, the default value of Y_Gamma0 is 0, and default value of Y_Gamma255 is 255.	

19.5.62P_SCALER2_Y_HIS0~ P_SCALER2_Y_HIS255

The P_SCALER2_Y_HISX register is used to represent the detail Y histogram of scaler's output.

P_SCALER2_Y_HIS0X 0x93002800 + 4*X

Scaler Y Histogram Register

Bit	15	14	13	12	11	10	9	8
Function								

Default 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
Y_HisX[7:0]							

0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
31:8			Reserved.	
7:0	Y_HisX	R/W	Y histogram X of scaler's output. This register will be reset to 0x00 after RESET bit is set to 1. This register has over-flow protection so the maximum value will be kept at 0xFFFF if the Y in this value is too much.	

20 CMOS Sensor Interface (CSI0)

20.1 Introduction

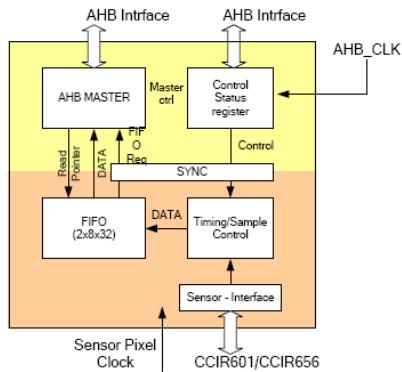
GPL32900A supports several interfaces: Standard CCIR601 / CCIR656. The maximum sensor resolution can reach 4095x4095. Both the capture mode and preview mode are supported.

20.2 Features

- Support parallel data interface: CCIR656/CCIR601
- Support YUV/YCbCr/RGB565/RGB1555
- Programmable capture Horizontal/Vertical Pixel Count.
- Programmable phase and polarity of master clock
- Built-in received FIFO size is Duplex 8x32 bit.
- Support YUV/YCbCr to RGB transform.
- Support RGB to YUV/YCbCr transform.
- Support 3 frame buffer address setting.

20.3 Block Diagram

The following diagram is a functional block diagram of the SPI module.



20.4 Sensor Control Pin Configuration

Name	I/O	Description
SENSOR0_D0	I	Sensor data 0 input.
SENSOR0_D1	I	Sensor data 1 input.
SENSOR0_D2	I	Sensor data 2 input.
SENSOR0_D3	I	Sensor data 3 input.
SENSOR0_D4	I	Sensor data 4 input.
SENSOR0_D5	I	Sensor data 5 input.

Name	I/O	Description
SENSOR0_D6	I	Sensor data 6 input.
SENSOR0_D7	I	Sensor data 7 input.
SENSOR0_HSYNC	O	Sensor hsync signal output.
SENSOR0_VSYNC	O	Sensor vsync signal output.
SENSOR0_PCLK	I	Sensor pixel clock input
SENSOR0_MCLK	O	Sensor main clock output

20.5 Register Summary

Name	Address	Description
P_CSI0_CR0	0x93003000	CSI0 control register 0
P_CSI0_CR1	0x93003004	CSI0 control register 1
P_CSI0_HSET	0x93003008	CSI0 horizontal setting
P_CSI0_VSET	0x9300300C	CSI0 vertical setting
P_CSI0_LSTP	0x9300301C	CSI0 per line offset in frame buffer
P_CSI0_FBADR0	0x93003020	CSI0 frame buffer address0
P_CSI0_FBADR1	0x93003024	CSI0 frame buffer address1
P_CSI0_FBADR2	0x93003028	CSI0 frame buffer address2
P_CSI0_HOLD	0x9300302C	CSI0 hold next frame control bit
P_CSI0_IRQEN	0x93003078	CSI0 IRQ enable register
P_CSI0_IRQSTS	0x9300307C	CSI0 IRQ status

20.6 Register Definition

P_CSI0_CR0 **0x93003000** **CSI0 control register 0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	UPDATE	-		FRMDIS	-	BUFSEL	-	EDGE		FRMEND		OUTMODE	-			

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RGBFMT	-	UVINV	INSEQ	FMTOUT	FMTIN	-	INTL	FIELDODD	VSACT	HSACT	CCIR	EN			

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description										Condition
31	UPDATE	R/W	Indicate the sensor controller to update current setting when frame end event or bit0 EN set from 0 to 1										0: no action 1: Update when frame end event or EN set 0 to 1
[30:28]	Reserve		Reserve										

Bit	Function	Type	Description	Condition
27	FRMDIS	R/W	Ref. GID15	0: Normal 1: Frame Disable
26	Reserve		Reserve	
[25:24]	BUFSEL	R/W	Select the input data to address 0 or 1 or 2	00: Select Frame Buffer Addr0 01 Select Frame Buffer Addr1 10: Select Frame Buffer Addr2 11: Select Frame Buffer Addr0
23	Reserve		Reserve	
22	EDGE	R/W	Select the polarity of sensor pclk to sample the sensor data input	0: sample data at positive edge 1: sample data at negative edge
[21:20]	FRMEND	R/W	Frame end mode	Non-interlace mode: Configuration update and frame end IRQ generate at every frame(field) end Interlace mode: 00: every frame(field) end 01: odd field end 10: even field end 11: every frame(field) end
19	OUTMODE	R/W	Output data mode	0: Normal mode Data0- V ₀ Y ₁ U ₀ Y ₀ Data1- V ₁ Y ₃ U ₁ Y ₂ Data2- V ₂ Y ₅ U ₂ Y ₄ Data3- V ₃ Y ₇ U ₃ Y ₆ 1: Normal mode Data0- Y ₃ Y ₂ Y ₁ Y ₀ Data1- U ₃ U ₂ U ₁ U ₀ Data2- Y ₇ Y ₆ Y ₅ Y ₄ Data3- V ₃ V ₂ V ₁ V ₀
15	RGBFMT	R/W	Select input data to RGB1555 or RGB565	0: RGB565 1: RGB1555
11	UVINV	R/W	Inverse the UV value or not which input from sensor	0: Output the UV value from sensor input data 1: Inverse the MSB(+128) of the UV from sensor input data and output to frame buffer or internal color transfer module for YUV/YCbCr -> RGB transfer
10	INSEQ	R/W	Sensor input sequence	0: U->Y->V->Y 1: Y->U->Y->V

Bit	Function	Type	Description	Condition
9	FMTOUT	R/W	Output data format	0: RGB format 1: YUV format
8	FMTIN	R/W	Input data format	0: RGB format 1: YUV format
5	INTL	R/W	Select interlace or non-interlace mode	0: Non-Interlace mode 1: Interlace mode, odd field data start address offset is 0, and even field data start address offset is P_CSI0_LSTP
4	FIELDODD	R/W		0: The field is odd when field signal is low 1: The field is odd when field signal is high
3	VSACT	R/W	The polarity of vsync to valid data input	0: Active low 1: Active high
2	HSACT	R/W	The polarity of hsync to valid data input	0: Active low 1: Active high
1	CCIR	R/W	CCIR interface select	0: CCIR601 1: CCIR□
0	EN	R/W	Sensor interface enable register	0: Disable 1: Enable

P_CSI0_CR1
0x93003004
CSI0 control register 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function														AHBEN		RST
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:3]	Reserve			
2	AHBEN	R/W	AHB Master enable	0: disable 1: enable
1	Reserve			
0	RST	R/W	Reset CSI0 PCLK domain	0: normal 1: reset CSI0 PCLK domain

P_CSI0_HSET

0x93003008

Horizontal setting

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	HDS															HSTART
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	-	-	-	-	HSIZE														
Default	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1			

Bit	Function	Type	Description												Condition
[31:28]	HDS	R/W	Horizontal down sample												
[27:16]	HSTART	R/W	Horizontal pixel start position												
[15:12]	Reserve														
[11:0]	HSIZE	R/W	Horizontal pixel count-1												Reset value: 0x27F

P_CSI0_VSET

0x9300300C

Vertical setting

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	VDS															VSTART
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	-	-	-	-	VSIZE														
Default	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1			

Bit	Function	Type	Description												Condition
[31:28]	VDS	R/W	Vertical down sample												
[27:16]	VSTART	R/W	Vertical pixel start position												
[15:12]	Reserve														
[11:0]	VSIZE	R/W	Vertical pixel count-1												Reset value: 0x1DF

P_CSI0_LSTP

0x9300301C

Per line offset in frame buffer

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LINESTEP															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	Reserve			
[15:0]	LINESTEP	R/W	Per line offset in frame buffer LINESTEP[4:0] must be 0	Reset value: 0x500

P_CSI0_FBADR0 **0x93003020** **Frame buffer address 0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function																
Default																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default																

Bit	Function	Type	Description	Condition
[31:5]	FBADR0	R/W	Frame buffer address 0	

P_CSI0_FBADR1 **0x93003024** **Frame buffer address 1**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function																
Default																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default																

Bit	Function	Type	Description	Condition
[31:5]	FBADR1	R/W	Frame buffer address 1	

P_CSI0_FBADR2 **0x93003028** **Frame buffer address 2**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function																
Default																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default																



Bit	Function	Type	Description	Condition
[31:5]	FBADR2	R/W	Frame buffer address 2	

Bit	Function	Type	Description	Condition
[31:1]	Reserve			
0	HOLD	R/W	Hold the CSI0	

Bit	Function	Type	Description	Condition
[31:8]	Reserve			
7	UPDATE	R/W	Control Register Update IRQ enable	0: disable 1: enable
6	LASTLINE	R/W	Lastline of frame IRQ enable	0: disable 1: enable
[5:3]	Reserve			
2	FEND	R/W	CSI0 Frame-End IRQ enable	0: disable 1: enable
1	HOLD	R/W	Hold the frame ack IRQ enable	0: disable 1: enable
0	OF	R/W	CSI0 FIFO Over-Flow IRQ enable	0: disable 1: enable

P_CSI0_IRQSTS
0x9300307C
IRQ status register

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	UPDATE	LASTLINE	-	-	-	FEND	HOLD	OF
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

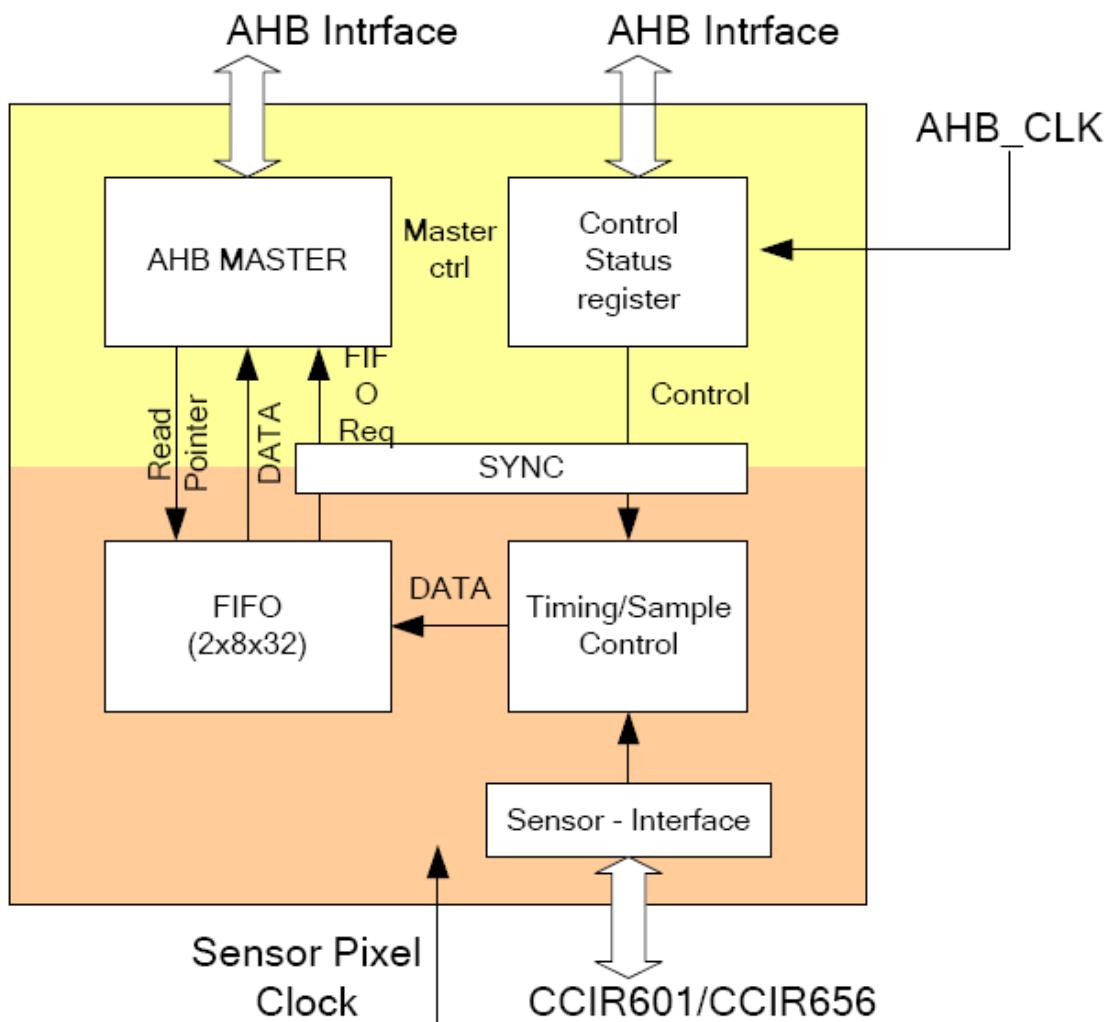
Bit	Function	Type	Description	Condition
[31:8]	Reserve			
7	UPDATE	R/WC	Control Register Update IRQ status	
6	LASTLINE	R/WC	Lastline of frame IRQ status	
[5:3]	Reserve			
2	FEND	R/WC	CSI0 Frame-End IRQ status	
1	HOLD	R/WC	Hold the frame ack IRQ status	
0	OF	R/WC	CSI0 FIFO Over-Flow IRQ status	

21 CMOS Sensor Interface 1

21.1 Features

- Parallel Data Interface (CCIR601/CCIR656).
- Support YUV/YCbCr/RGB565/RGB1555 Sensor.
- Programmable capture Horizontal/Vertical Pixel count.
- Received FIFO size is Duplex 8x32 bits.
- Support YUV/YCbCr to RGB Transform.
- Support RGB to YUV/YCbCr Transform.
- Support 3 Frame Buffer Address Setting.

21.2 Block Diagram



21.3 CSI1 Register

Name	Address	Description
P_CSI1_CR0	0x93003000	CONTROL REGISTER0
P_CSI1_CR1	0x93003004	CONTROL REGISTER1
P_CSI1_HSET	0x93003008	HORIZONTAL SETTING
P_CSI1_VSET	0x9300300C	VERTICAL SETTING
P_CSI1_LSTP	0x93003010	LINE OFFSET IN FRAME BUFFER
P_CSI1_FBADR0	0x93003020	FRAME BUFFER ADDRESS0
P_CSI1_FBADR1	0x93003024	FRAME BUFFER ADDRESS1
P_CSI1_FBADR2	0x93003028	FRAME BUFFER ADDRESS2
P_CSI1_HOLD	0x9300302C	HOLD NEXT FRAME CONTROL BIT
P_CSI1_IRQEN	0x9300302C	IRQ ENABLE REGISTER
P_CSI1_IRQSTS	0x9300302C	IRQ STATUS

21.4 Register Definition

21.4.1 Sensor General Control Register

The general control registers are used to control the timing type and data type of sensor. Many of these registers are depend on the sensor type, please check sensor's datasheet for detail about these setting.

21.4.2 CSI1 CONTROL REGISTER0 Register

CSI1 CONTROL REGISTER0								
Bit	31	30	29	28	27	26	25	24
Function	CSI_Update	-			Frame_Disable	-	Buffer_Select	
Default	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	-	Sample_Edge	Frame_end_Mode	Mp4_Out			-	
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	RGB1555_In	-		UV_Inverse	INSEQ	YUVOUT	YUVIN	
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	-	Interlace	FieldHOdd	VsyncHActive	HsyncHActive	CCIR656	CSIEN	
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31]	CSI_Update	R/W	0 : no Action 1 : To indicate the sensor controller to update current setting when frame_end event or CSIEN set from 0 to 1	-
[30:28]	-	-	Reserved	-
[27]	Frame_Disable	R/W	0 : Normal 1 : Frame Disable	
[26]	-	-	Reserved	
[25:24]	Buffer_Select	R/W	00: Select Frame Buffer Address0 01: Select Frame Buffer Address1 10: Select Frame Buffer Address2 11: Select Frame Buffer Address0	
[23]	-	-	Reserved	-
[22]	Sample_Edge	R/W	0 : sample data at positive edge of sensor pclk 1 : sample data at negative edge of sensor pclk	
[21:20]	Frame_end_Mode	R/W	Non-Interlace : configuration update and frame_end IRQ generate at every frame(field) end Interlace : 00: every frame(field) end 01: odd field end 10: even field end 11: every frame(field) end	-
[19]	Mp4_Out	R/W	0 : output data is normal mode data0 = V0Y1U0Y0 data1 = V1Y3U1Y2 data2 = V2Y5U2Y4 data3 = V3Y7U3Y6 1 : output data is MP4 mode data0 = Y3Y2Y1Y0 data1 = U3U2U1U0 data2 = Y7Y6Y5Y4 data3 = V3V2V1V0	-
[18:16]	-	-	Reserved	
[15]	RGB1555_In	R/W	0 : Input data is RGB565 format 1 : Input data is RGB555 format	
[14:12]	-	-	Reserved	



Bit	Function	Type	Description	Condition
[11]	UV_Inverse	R/W	0 : output the UV value from sensor input data 1 : Inverse the MSB (+128) of the UV from sensor input data and output to frame-buffer or internal color transfer module for YUV/YcbCr -> RGB transfer	-
[10]	INSEQ	R/W	0 : sensor input sequence is U->Y->V->Y 1 : sensor input sequence is Y->U->Y->V	-
[9]	YUVOUT	R/W	0 : Output data is RGB format 1 : Output data is YUV format	-
[8]	YUVIN	R/W	0 : Input data is RGB format 1 : Input data is YUV format	
[7:6]	-	-	Reserved	
[5]	Interlace	R/W	0 : Non-Interlace mode 1 : In Interlace mode, odd field data start address offset is 0, and even field start address offset is CSI_LSTP.	
[4]	FieldHOdd	R/W	0 : the field is odd when field signal is low 1 : the field is odd when field signal is high	
[3]	VsyncHActive	R/W	0 : Data is valid when Vsync is low 1 : Data is valid when Vsync is high	
[2]	HsyncHActive	R/W	0 : Data is valid when Hsync is low 1 : Data is valid when Hsync is high	
[1]	CCIR656	R/W	0 : CCIR601 Interface. 1 : CCIR656 Interface.	-
[0]	BLANKBOTTOM	R/W	Blanking From BOTTOM 0	-

21.4.3 CSI1 CONTROL REGISTER1 Register



Bit	Function	Type	Description	Condition
[31:3]	-	-	Reserved	-
[2]	AHBMASTEREN	R/W	0 : Disable AHB MASTER 1 : Enable AHB MASTER	
[1]	-	-	Reserved	-
[0]	CSIRST	R/W	0: normal 1: Reset CSI PCLK domain	

21.4.4 CSI1 HORIZONTAL SETTING Register

Bit	Function	Type	Description	Condition
[31:28]	HDS	R/W	Horizontal Down Sample	-
[27:16]	HSTART	R/W	Horizontal Pixel Start Position	-

Bit	Function	Type	Description				Condition
[15:12]	-	-	Reserved				-
[11:0]	HSIZE	R/W	Horizontal Pixel Count -1				

21.4.5 CSI1 VERTICAL SETTING Register

P_CSI1_VSET 0x9300300C CSI1 VERTICAL SETTING								
Bit	31	30	29	28	27	26	25	
Function	VDS[31:28]				VSTART[27:24]			
Default	0	0	0	0	0	0	0	

23	22	21	20	19	18	17	16
VSTART[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-				VSIZE[11:8]			
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
VSIZE[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:28]	VDS	R/W	Vertical Down Sample				-
[27:16]	VSTART	R/W	Vertical Pixel Start Position				
[15:12]	-	-	Reserved				-
[11:0]	VSIZE	R/W	Vertical Pixel Count -1				

21.4.6 CSI1 PER LINE OFFSET IN FRAME BUFFER Register

P_CSI1_LSTP 0x93003010 CSI1 PER LINE OFFSET IN FRAME BUFFER							
Bit	31	30	29	28	27	26	25
Function	-						
Default	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
LINE_STEP[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
LINE_STEP[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:16]	-	-	Reserved					-
[15:0]	LINE_STEP	R/W	Vertical Pixel Count -1					

21.4.7 CSI1 FRAME BUFFER ADDRESS0 Register

P_CSI1_FBADR0 **0x93003020** **CSI1 FRAME BUFFER ADDRESS0**

Bit	31	30	29	28	27	26	25	24
CSI_FBADR0[31:24]								

Default	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---

23	22	21	20	19	18	17	16
CSI_FBADR0[23:16]							

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
CSI_FBADR0[15:8]							

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
CSI_FBADR0[7:5]							

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:5]	CSI_FBADR0	R/W	Frame Buffer Address0					-
[4:0]	-	-	Reserved					-

21.4.8 CSI1 FRAME BUFFER ADDRESS1 Register

P_CSI1_FBADR1								0x93003024								CSI1 FRAME BUFFER ADDRESS1																
Bit	31	30	29	28	27	26	25	24	CSI_FBADR1[31:24]																							
Function									CSI_FBADR1[31:24]																							
Default	0	0	0	0	0	0	0	0																								
	23	22	21	20	19	18	17	16	CSI_FBADR1[23:16]																							
	0	0	0	0	0	0	0	0	CSI_FBADR1[23:16]																							
	15	14	13	12	11	10	9	8	CSI_FBADR1[15:8]																							
	0	0	0	0	0	0	0	0	CSI_FBADR1[15:8]																							
	7	6	5	4	3	2	1	0	CSI_FBADR1[7:5]																							
	0	0	0	0	0	0	0	0	CSI_FBADR1[7:5]																							
Bit	Function	Type	Description						Condition																							
[31:5]	CSI_FBADR1	R/W	Frame Buffer Address1						-																							
[4:0]	-	-	Reserved						-																							

21.4.9 CSI1 FRAME BUFFER ADDRESS2 Register

P_CSI1_FBADR2								0x93003028								CSI1 FRAME BUFFER ADDRESS2								
Bit	31	30	29	28	27	26	25	24	CSI_FBADR2[31:24]															
Function									CSI_FBADR2[31:24]															
Default	0	0	0	0	0	0	0	0																
	23	22	21	20	19	18	17	16	CSI_FBADR2[23:16]															
	0	0	0	0	0	0	0	0	CSI_FBADR2[23:16]															
	15	14	13	12	11	10	9	8	CSI_FBADR2[15:8]															
	0	0	0	0	0	0	0	0	CSI_FBADR2[15:8]															

7	6	5	4	3	2	1	0		
CSI_FBADR2[7:5]			-						
0	0	0	0	0	0	0	0		

Bit	Function	Type	Description	Condition
[31:5]	CSI_FBADR2	R/W	Frame Buffer Address2	-
[4:0]	-	-	Reserved	-

21.4.10 CSI1 HOLD NEXT FRAME CONTROL BIT Register

P_CSI1_HOLD								0x9300302C								CSI1 HOLD NEXT FRAME CONTROL BIT										
Bit	31	30	29	28	27	26	25	24	Function	-								Default	0	0	0	0	0	0	0	
	23	22	21	20	19	18	17	16		-									0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8		-									0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0		-									CSI_HOLD							
	0	0	0	0	0	0	0	0		-									0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:1]	-	-	Reserved	-
[0]	CSI_HOLD	R/W	Hold the CSI0	-

21.4.11 CSI1 IRQ ENABLE Register

P_CSI1_IRQEN								0x93003078								CSI1 IRQ ENABLE									
Bit	31	30	29	28	27	26	25	24	Function	-								Default	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24		-									0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
IRQ_CRUPDATE_EN	IRQ_LASTLINE_EN	-	IRQ_FRAMEEND_EN	IRQ_HOLD_EN	IRQ_OF_EN	-	-
0	0	0 0 0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:1]	-	-	Reserved				-
[7]	IRQ_CRUPDATE_EN	R/W	Control Register Update IRQ Enable				
[6]	IRQ_LASTLINE_EN	R/W	Lastline of a frame IRQ Enable				
[5:3]	-	-	Reserved				-
[2]	IRQ_FRAMEEND_EN	R/W	CSI0 Frame-End IRQ Enable				
[1]	IRQ_HOLD_EN	R/W	Hold The Frame Ack IRQ Enable				
[0]	IRQ_OF_EN	R/W	CSI0 FIFO Over-Flow IRQ Enable				-

21.4.12 CSI1 IRQ STATUS Register

P_CSI1_IRQSTS		0x9300307C								CSI1 IRQ STATUS							
Bit	Function	31	30	29	28	27	26	25	24								
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	-	-	-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	-	-	-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	-	-	-	-	-	-	-	-	-	-
IRQ_CRUPDATE_STS	IRQ_LASTLINE_STS	-	IRQ_FEND_STS	IRQ_HOLD_STS	IRQ_OF_STS	-	-	-	-	-	-	-	-	-	-	-	-
0	0	0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:1]	-	-	Reserved	-
[7]	IRQ_CRUPDATE_STS	R/W	Control Register Update IRQ Status	
[6]	IRQ_LASTLINE_STS	R/W	Lastline of a frame IRQ Status	
[5:3]	-	-	Reserved	-
[2]	IRQ_FRAMEEND_EN	R/W	CSI0 Frame-End IRQ Status	
[1]	IRQ_HOLD_STS	R/W	Hold The Frame Ack IRQ Status	
[0]	IRQ_OF_STS	R/W	CSI0 FIFO Over-Flow IRQ Status	-

21.5 FUNCTION / DESIGN Description

Color Transform Equation:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} Y \\ Cb - 128 \\ Cr - 128 \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.172 & -0.339 & 0.511 \\ 0.511 & -0.428 & -0.083 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.371 \\ 1 & -0.336 & -0.698 \\ 1 & 1.732 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.371 \\ 1 & -0.336 & -0.698 \\ 1 & 1.732 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ Cb - 128 \\ Cr - 128 \end{bmatrix}$$

22 Color DSP (CDSP)

22.1 Introduction

A Color DSP (CDSP) controller built-in GPL32900A, essentially facilitates process the image data. The CDSP supported 3 kinds of image input source, front end sensor, mipi sensor and sdram. The CDSP input data format could be Raw8/Raw10 and YUV422. It built in many powerful image process units like bad pixel、optical block、lens compensation、white balance gain control、Lut gamma・interpolation、edge enhance・suppression・de-noise filter・yuv horizontal average and yuv special mode functions. It also supported auto focus・auto white balance and auto exposure for image process. CDSP supported RGB horizontal scale・image crop, yuv horizontal scale down and yuv vertical scale down units for image size scale.

22.2 Features

- Supported real time image processing on the image stream from sensor.
- Supported image source from DRAM.
- Supported max resolution 4095x4095.
- Supported bad pixel cancel, optional black subtraction, lens compensation, crop, scaling, white balance.
- Supported gamma correction, color correction, rgb to yuv conversion, yuv filtering, hue shifting, edge enhancement, noise suppression.
- Supported auto focus, auto white balance, auto expose data collection.

23 Mobile Industry Processor Interface (MIPI)

23.1 Introduction

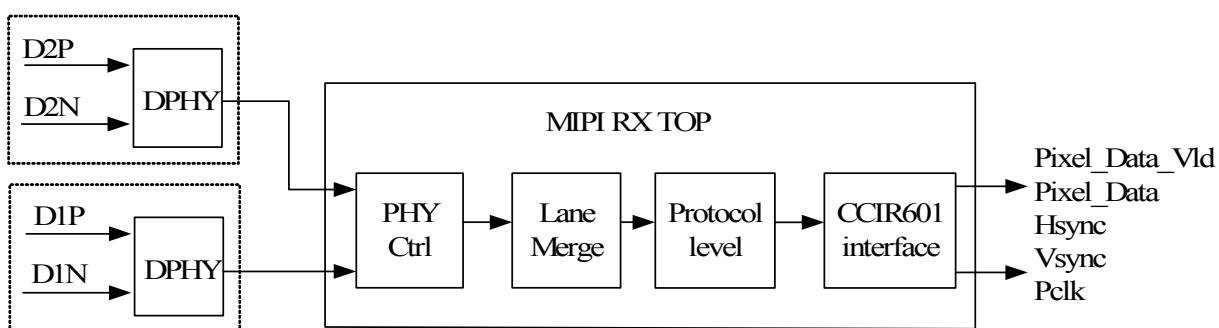
A mobile industry processor interface (MIPI) controller, built-in GPL32900A, essentially facilitates the communication with the sensor device. Generally it can support 1 or 2 data lanes and the Mipi Controller output Interface is CCIR601 interface. There are six hardware control signals related to the Data1+, Data1-, Data2+, Data2-, Clock+ and Clock-.

23.2 Features

- Support 1,2 Data Lanes
- All lanes are unidirectional high speed data transfer
- Don't support Escape Mode functionality
- Continuously running clock (MIPI clock lane)
- The minimum data unit shall be one byte
- ECC check for header and CRC check for data
 - MIPI Alliance Standard for Camera Serial Interface 2
- The pixel data output interface is CCIR601 interface
 - CPU configure HBP (Horizontal Back Porch)
 - CPU configure HFP (Horizontal Front Porch)
 - CPU configure HDW (Horizontal Data Width)
 - CPU configure VDH (Vertical Data Height)

23.3 Block Diagram

The following diagram is a functional block diagram of the MIPI module.



23.4 MIPI Control Pin Configuration

Name	I/O	Description
Data2+ /Data2-	I	Data Lane 2.
Data1+ /Data1-	I	Data Lane 1.
Clock+ / Clock-	I	Clock input.

23.5 Register Summary

Name	Address	Description
P_MIPI_GLB_CSR	0x9300D000	Mipi global configure
P_MIPI_PHY_RST	0x9300D004	Mipi D-Phy software reset
P_MIPI_RC_CTRL	0x9300D008	Mipi D-Phy RC control
P_MIPI_ECC_ORDER	0x9300D00C	Mipi payload header sequence for ECC calculation
P_MIPI_CCIR601_TIMING	0x9300D010	Mipi CCIR601 interface timing
P_MIPI_IMAGE_SIZE	0x9300D014	Mipi image size set
P_MIPI_DATA_FORMAT	0x9300D020	Mipi data lane number
P_MIPI_PAYLOAD_HEADER	0x9300D024	Mipi payload header
P_MIPI_CTRL_STATE	0x9300D028	Mipi data lane control state
P_MIPI_HEADER_DATA	0x9300D030	Mipi received hrader data
P_MIPI_HEADER_DATA_VLD	0x9300D034	Mipi received header data valid
P_MIPI_INT_EN	0x9300D040	Mipi interrupt enable
P_MIPI_INT_SRC	0x9300D080	Mipi interrupt source
P_MIPI_INT_STATUS	0x9300D0C0	Mipi interrupt status

23.6 Register Definition

P_MIPI_GLB_CSR 0x9300D000 Mipi global configure							
Bit	15	14	13	12	11	10	9
Function	-		SEL_PIX_CLK		-		LANE_NUM_SYS
Default	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	BYTE_CLK_NEG_SAMPLE	LOW_POWER_EN		-			MIPI_ENABLE
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:13]	-	-	Reserved	-
12	SEL_PIX_CLK	R/W	Select output pixel clock. 0: use the MIPI D-PHY BYTE_CLK. 1: use generated clock as output pixel clock	0: use the MIPI D-PHY BYTE_CLK. 1: use generated clock as output pixel clock
[11:9]	-	-	Reserved	-
8	LANE_NUM_SYS	R/W	Mipi data lane set. 0: mipi 1 lane 1: mipi 2 lanes.	0: mipi 1 lane 1: mipi 2 lanes.
[7:6]	-	-	Reserved	-
5	BYTE_CLK_NEG_SAMPLE	R/W	1: The design use negedge of the MIPI D-PHY BYTE_CLK to sample the D-PHY output byte data. 0: The design use posedge of the MIPI D-PHY BYTE_CLK to sample the D-PHY output byte data.	0: Posedge 1: Negedge.
4	LOW_POWER_EN	R/W	Low power enable, when D-lane/CLK-lane enter Low Power Mode, disable HS_RX for low power consumption.	0: Disable 1: Enable
[3:1]	-	-	Reserved	-
0	MIPI_ENABLE	R/W	MIPI transfer turn on. When MIPI_ENABLE=0; The MIPI_RX_TOP will go to initial state; during the initial state, all the MIPI bus line states will be ignored.	0: Disable 1: Enable

P_MIPI_PHY_RST
0x9300D004
Mipi D-Phy software reset

Bit	15	14	13	12	11	10	9	8
Function	-							

Default

0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
-		NOCK_RST_EN		-	SOFT_RST_DA		SOFT_RST_CK
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:5]	-	-	Reserved					-
4	NOCK_RST_EN	R/W	When MIPI clock lane does not have MIPI DDR bit clock (MMR 0x2c, CHECK_MIPI_CLK_D == 1' b1 and CHECK_MIPI_CLK_2D == 1' b1), reset D-PHY clock lane and data lane enable. Used when system clock frequency is smaller than MIPI CKP frequency.					
[3:2]	-	-	Reserved					-
1	SOFT_RST_DA	W	Write 1 software reset the D-PHY data lane.					
0	SOFT_RST_CK	W	Write 1 software reset the D-PHY clock lane.					

P_MIPI_RC_CTRL 0x9300D008 Mipi D-Phy RC control							
Bit	15	14	13	12	11	10	9
Function	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0
	7	6	5	4	3	2	1
	-	-	-	-	C_CON	R_CON	-
	0	0	0	0	0	1	0
	0	0	0	0	0	0	1

Bit	Function	Type	Description					Condition
[15:4]	-	-	Reserved					-
1	C_CON	R/W	PHY capacitance control					-
0	R_CON	R/W	PHY resistance control					-

P_MIPI_ECC_ORDER 0x9300D00C Mipi payload header sequence for ECC calculation							
Bit	31	30	29	28	27	26	25
Function	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0
	23	22	21	20	19	18	17
	-	-	-	-	-	-	CHECK_HS_SEQ
	0	0	0	0	0	0	1
	15	14	13	12	11	10	9
	-	-	-	-	-	-	DA_MASK_CNT[7:0]
	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-					ECC_CHECK_EN	ECC_ORDER	
0	0	0	0	0	1	0	0

Bit	Function	Type	Description	Condition
[31:17]	-	-	Reserved	-
16	CHECK_HS_SEQ	R/W	Check HS sequence set.	1: Check HS sequence when enter HS mode. 0: Just check LP-00 for enter into HS mode.
[15:8]	DA_MASK_CNT	R/W	When LP-00(HS PHY termination enable, HS begin), mask DA_MASK_CNT high speed receiver data, then search SOT in the HS RX data	
[7:3]	-	-	Reserved	-
2	ECC_CHECK_EN	R/W	Enable the header ECC check.	0: Disable 1: Enable
[1:0]	ECC_ORDER	R/W	PH sequence for ECC calculation:	Bit 0: 0={WC_L,WC_H}; 1={WC_H,WC_L}; Bit 1: 0= {DI,WC}; 1={WC,DI};

P_MIPI_CCIR601_TIMING 0x9300D010 Mipi CCIR601 interface timing

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							BLANKING_LINE_EN

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
-							-

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
HFP				HBP			
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:17]	-	-	Reserved	-
16	BLANKING_LINE_EN	R/W	Blanking line enable;	0: mask HSYNC when VSYNC=0 1: unmask HSYNC when VSYNC=0
[15:8]	-	-	Reserved	-
[7:4]	HFP	R/W	Horizontal front porch	
[3:0]	HBP	R/W	Horizontal back porch	

P_MIPI_IMAGE_SIZE 0x9300D014 Mipi image size set								
Bit	31	30	29	28	27	26	25	24
Function	VHD[15:8]							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
VHD[7:0]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
HDW[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
HDW[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	VDW	R/W	Vertical data height					
[15:0]	HDW	R/W	Horizontal data width, Min value = 1.					

P_MIPI_DATA_FORMAT 0x9300D020 Mipi data lane number								
Bit	31	30	29	28	27	26	25	24
Function	DUMMY [24:16]							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
DUMMY [15:8]							
0	0	0	0	0	0	0	0
DUMMY [7:0]							
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
DATA_TYPE_CDSP_SYS	DATA_TYPE_MMR	-	-	-	-	-	DATA_FROM_MMR
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]	DUMMY	R/W	Dummy	
7	DATA_TYPE_CDSP_SYS	R/W	Mipi data output set	1: output to CDSP dataout[9:0] = {data[7:0],2' b0} 0: output to PPU dataout[9:0] = {2' b0, data[7:0]}
[6:4]	DATA_TYPE_MMR	R/W	CPU configure the data type (data format)	0: YVU 422 8-bit 1: RGB888 2: RGB565 3: RAW8 4: RAW10 5: RAW12 6: Generic 8-bit data 7: User defined byte data
[3:1]	-	-	Reserved	
0	DATA_FROM_MMR	R/W	MMR decide the data type enable	0: disable 1: Enable

P_MIPI_PAYLOAD_HEADER		0x9300D024								Mipi payload header	
Bit	15	14	13	12	11	10	9	8			
Function	-										
Default	0	0	0	0	0	0	0	0	0	0	
PLD_HEADER_MIPI											
	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:0]	PLD_HEADER_MIPI	R	MIPI protocol layer payload header (virtual data ID and the data type) after ECC correct if ECC_CHECK_EN(MMR 0x0C [2]) is set to 1.	-

P_MIPI_CTRL_STATE **0x9300D028** **Mipi data lane control stste**

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-							PHY_STATE_D1
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]	-	-	Reserved	-
[10:8]	PHY_STATE_CLK	R	Clock lane PHY state.	000: initial state 001: Control model state 010: High speed mode request state LP-01 011: High speed mode 100: Low power mode request LP-10 101: Ultra low power mode
[7:2]	-	-	Reserved	[15:8]
[1:0]	PHY_STATE_D1	R	Data-lane1 PHY state.	00: RX initial state 01: control model state 10: HS_REQ, High speed mode request state 11: HS_MODE, HS mode state

P_MIPI_HEADER_DATA **0x9300D030** **Mipi received hrader data**

Bit	31	30	29	28	27	26	25	24
Function	ECC_CODE[7:0]							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
WORD_CNT[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
WORD_CNT[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
DATA_ID[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:24]	ECC_CODE	R	The received ECC code data from the MIPI bus.			-	-	-
[23:8]	WORD_CNT	R	The received word count bytes from the MIPI bus without ECC correct.			-	-	-
[7:0]	DATA_ID	R	The received data ID byte from the MIPI bus without ECC correct.			-	-	-

P_MIPI_HEADER_DATA_VLD 0x9300D034 Mipi received header data valid

Bit	15	14	13	12	11	10	9	8
Function	-							

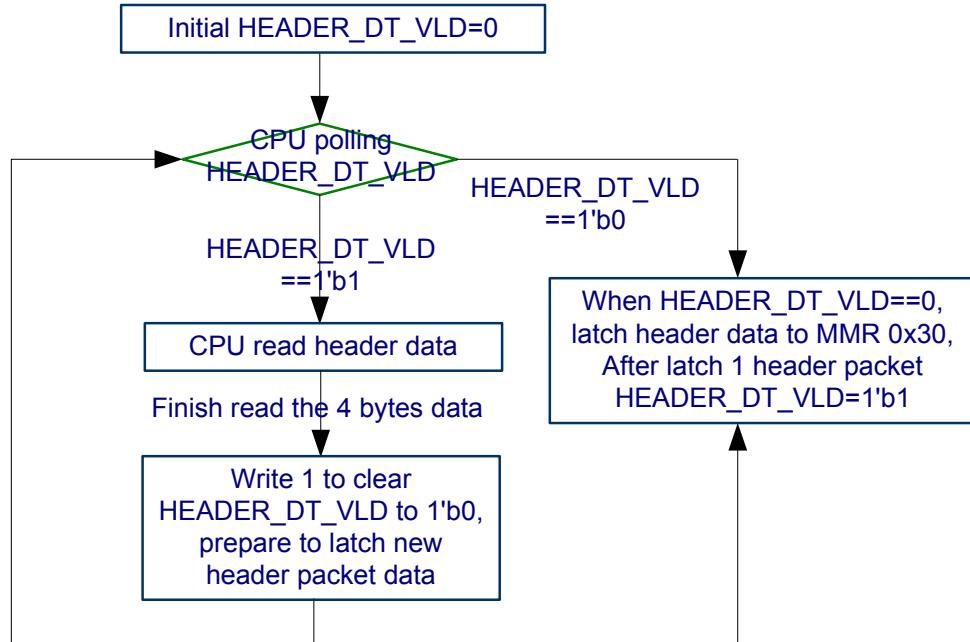
Default	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
-							HEADER_DT_VLD

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[15:1]	-	-	Reserved			-	-	-
0	HEADER_DT_VLD	W/R	The received data ID byte from the MIPI bus without ECC correct.			-	-	-

The following figure shows the CPU to read header data flow:



P_MIPI_INT_EN								0x9300D040								Mipi interrupt enable								
Bit	15	14	13	12	11	10	9	8																
Function	-																							
Default	0 0 0 0 0 0 0 0								0 0 0 0 0 0 0 0								0 0 0 0 0 0 0 0							

7	6	5	4	3	2	1	0
-	SOF_FLAG	SOT_SYNC_ERR1	CRC_ERR	HD_ERR	HD_ERR_1BIT	SOT_SYNC_ERR0	
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]	-	-	Reserved	-
5	SOF_FLAG	R/W	Start of frame (CCIR IF) flag interrupt enable	0: Disable 1: Enable
4	SOT_SYNC_ERR1	R/W	Lane1 MIPI high speed start of transmission synchronization error, (can't search sync_pattern) interrupt enable.	0: Disable 1: Enable
3	CRC_ERR	R/W	MIPI payload data CRC error interrupt enable	0: Disable 1: Enable
2	HD_ERR	R/W	MIPI header data error (error bit>1) interrupt enable	0: Disable 1: Enable

Bit	Function	Type	Description					Condition
1	HD_ERR_1BIT	R/W	MIPI header data error (error bit=1) interrupt enable					0: Disable 1: Enable
0	SOT_SYNC_ERR0	R/W	Lane0 MIPI high speed start of transmission synchronization error, (can't search sync_pattern) interrupt enable.					0: Disable 1: Enable

P_MIPI_INT_SRC								0x9300D080	Mipi interrupt source
Bit	15	14	13	12	11	10	9	8	
Function	-								

Default	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	-	SOF_FLAG	SOT_SYNC_ERR1	CRC_ERR	HD_ERR	HD_ERR_1BIT	SOT_SYNC_ERR0		
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description					Condition
[15:6]	-	-	Reserved					-
5	SOF_FLAG	R/W	MIPI payload data CRC error interrupt source.					Write 1 clear.
4	SOT_SYNC_ERR1	R/W	Lane1 MIPI high speed start of transmission synchronization error, (can't search sync_pattern) interrupt source.					Write 1 clear.
3	CRC_ERR	R/W	MIPI payload data CRC error interrupt source.					Write 1 clear.
2	HD_ERR	R/W	MIPI header data error (error bit>1) interrupt source.					Write 1 clear.
1	HD_ERR_1BIT	R/W	MIPI header data error (error bit=1) interrupt source.					Write 1 clear.
0	SOT_SYNC_ERR0	R/W	Lane0 MIPI high speed start of transmission synchronization error, (can't search sync_pattern) interrupt source.					Write 1 clear.

P_MIPI_INT_STATUS								0x9300D0C0	Mipi interrupt status
Bit	15	14	13	12	11	10	9	8	
Function	-								
Default	0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0	
-	SOF_FLAG	SOT_SYNC_ERR1	CRC_ERR	HD_ERR	HD_ERR_1BIT	SOT_SYNC_ERR0		
0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:6]	-	-	Reserved	-
5	SOF_FLAG	R	Start of Frame flag. Debug only (did not assert MIPI interrupt flag)	
4	SOT_SYNC_ERR1	R	Lane1 MIPI high speed start of transmission synchronization error, (can't search sync_pattern) interrupt flag.	
3	CRC_ERR	R	MIPI payload data CRC error interrupt flag.	
2	HD_ERR	R	MIPI header data error (error bit>1) interrupt flag.	
1	HD_ERR_1BIT	R	MIPI header data error (error bit=1) interrupt flag.	
0	SOT_SYNC_ERR0	R	Lane0 MIPI high speed start of transmission synchronization error, (can't search sync_pattern) interrupt flag.	

24 TFT1 LCD

24.1 Introduction

A TFT1 LCD controller is built-in GPL32900A to support UPS051/UPS052/CCIR 601/CCIR 656, parallel RGB, TCON mode and memory interface mode. The screen resolution from PPU output can be set to 320x240~1024x768 when TFT1 LCD panel is used. For TFT1 LCD panel with memory interface, such as i80 mode, Generalplus strongly suggest to connect it to the embedded TFT1 LCD interface, not memory bus, so that PPU can directly output the data via TFT1 LCD interface to TFT1 LCM for display.

24.2 Features

- The maximum horizontal and vertical pixels are individually 1920x1080, and all can be programmed
- The TFT1 clock can be a divisor of system clock (/1, ~ /32)
- TFT1_REQ_RANGE signal for PPU prepare data & key-scan.
- H_COMPRESS register for horizontal compressing for 2 times. It only works when mode = UPS051.
- The width and polarity of HSYNC and VSYNC are programmable
- The horizontal pixels, including active region and blank region, can be programmed
- The vertical lines, including active region and blank region, can be programmed
- The RGB order is programmable for panel color adjustment.
- Supports parallel mode ; RGB data can be 888, 565 and 666 bits
- Supports TCON mode ; RGB data can be 565 and 666 bits
- Supports interrupt when the showing frame enters the blanking state
- Supports the adjustable order of RGB data for each odd line and even line in RGB mode
- Inputs only RGB parallel mode.
- Supports JxK to MxN panel transfer, where M < J N<K
- In memory mode, programmer can show the memory I/F panel window beginning from any point int the PPU 320x240 or 640x480 size.
- Supports tearing effect in memory mode
- Output can be 10 type formats:
UPS051
UPS052
CCIR601
CCIR656
Parallel RGB mode
TCON mode
Memory mode: Command Write (RS=0)
Memory mode: Command Read(RS=0)
Memory mode: Data Write (RS=1)

Memory mode: Data Read(RS=1)

Memory mode: Data Show Continuous (RS=1)

Memory mode: Data Show Once(RS=1)

Byte access (P_TFT1_CTRL[13])

Note: Where memory mode access can be programmed.

24.3 Interface Signals

Pin Name	I/O	Description
VSYNC	O	VSYNC / CKV signal
Hsync	O	Hsync / LD signal
DCLK	O	DCLK signal
DE	O	DATA valid / OEV signal
DISP_OUT [23:0]	O	TFT1 DATA output
STVU/D	O	Vertical pulse in TCON mode
STHR/L	O	Horizontal pulse in TCON mode
POL	O	Polarity in TCON mode

The following table shows the connections when using panel with memory mode:

Pin Name	I/O	Description
LCM_CS	O	CS signal
LCM_RS	O	D/C(Data/Command) signal
LCM_WR	O	WE signal
LCM_RD	O	OE signal
DISP_OUT [15:0]	O	TFT1 DATA output
DISP_OUT [9]	I	Tearing effect input

24.4 Register Summary

Name	Address	Description
P_TFT1_CTRL	0x93020140	TFT1 Control Register
P_TFT1_V_PERIOD	0x93020144	TFT1 Vertical Period Register
P_TFT1_VS_WIDTH	0x93020148	TFT1 VSYNC Width Register
P_TFT1_V_START	0x9302014C	TFT1 Vertical Start Register
P_TFT1_V_END	0x93020150	TFT1 Vertical End Register
P_TFT1_H_PERIOD	0x93020154	TFT1 Horizontal Period Register
P_TFT1_HS_WIDTH	0x93020158	TFT1 HSYNC Width Register
P_TFT1_H_START	0x9302015C	TFT1 Horizontal Start Register
P_TFT1_H_END	0x93020160	TFT1 Horizontal End Register
P_TFT1_LINE_RGB_ORDER	0x93020164	TFT1 Line RGB Order Control Register
P_TFT1_STATUS	0x93020168	TFT1 Status Register

Name	Address	Description
P_TFT1_MEM_BUFFER_WR	0x9302016C	TFT1 Memory Buffer Write Register
P_TFT1_MEM_BUFFER_RD	0x93020170	TFT1 Memory Buffer Read Register
P_TFT1_TE_CTRL	0x93020180	TFT1 Tearing Effect Control Register
P_TFT1_TE_HS_WIDTH	0x93020184	TFT1 Tearing Effect HSYNC Width Register
P_TFT1_VS_START	0x930201B0	TFT1 Vertical Show Start Register
P_TFT1_VS_END	0x930201B4	TFT1 Vertical Show End Register
P_TFT1_HS_START	0x930201B8	TFT1 Horizontal Show Start Register
P_TFT1_HS_END	0x930201BC	TFT1 Horizontal Show End Register
P_TFT1_CLIP_V_START	0x930203B0	TFT1 Vertical Clip Start Register
P_TFT1_CLIP_V_END	0x930203B4	TFT1 Vertical Clip End Register
P_TFT1_CLIP_H_START	0x930203B8	TFT1 Horizontal Clip Start Register
P_TFT1_CLIP_H_END	0x930203BC	TFT1 Horizontal Clip End Register
P_TFT1_TS_CKV	0x930203C0	TFT1 CKV Start Position Register
P_TFT1_TW_CKV	0x930203C4	TFT1 CKV Width Register
P_TFT1_TS_MISC	0x930203C8	TFT1 Misc Control Register
P_TFT1_TS_POL	0x930203CC	TFT1 POL Start Position Register
P_TFT1_TS_STV	0x930203D0	TFT1 STV Start Position Register
P_TFT1_TW_STV	0x930203D4	TFT1 STV Pulse Width Register
P_TFT1_TS_STH	0x930203D8	TFT1 STH Start Position Register
P_TFT1_TW_STH	0x930203DC	TFT1 STH Pulse Width Register
P_TFT1_TS_OEV	0x930203E0	TFT1 OEV Start Position Register
P_TFT1_TW_OEV	0x930203E4	TFT1 OEV Pulse Width Register
P_TFT1_TS_LD	0x930203E8	TFT1 LD Start Position Register
P_TFT1_TW_LD	0x930203EC	TFT1 LD Pulse Width Register
P_TFT1_DITHER_TAB0	0x930203F0	TFT1 Dithering Parameter0 Register
P_TFT1_DITHER_TAB1	0x930203F4	TFT1 Dithering Parameter1 Register
P_TFT1_DITHER_TAB2	0x930203F8	TFT1 Dithering Parameter2 Register
P_TFT1_DITHER_TAB3	0x930203FC	TFT1 Dithering Parameter3 Register

24.5 Register Definition

P_TFT1_CTRL		0x93020140								TFT1 Control Register							
Bit	15	14	13	12	11	10	9	8									
Function	VSU	INTL	MEM_BYTE	HCMP	DINV	CINV	HINV	VINV									
Default	0	0	0	0	0	0	0	0									

7	6	5	4	3	2	1	0
MODE				CLK_SEL			TFT1EN
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	VSU	R/W	VSYNC Unit	0: Unit is base on DCLK 1: Unit is base on HSYNC
14	INTL	R/W	TFT1 Interlace Mode Selection	0: Non-interlaced 1: Interlaced
13	MEM_BYTE	R/W	BYTE Access of Memory Mode	0: WORD mode of memory mode 1: BYTE mode of memory mode
12	HCMP	R/W	Horizontal Compression	0: Disable horizontal compression 1: Enable horizontal compression
11	DINV	R/W	Data Enable Inversion	0: Not invert DATA_EN output 1: Invert DATA_EN output
10	CINV	R/W	DCLK Inversion	0: Not invert DCLK output 1: Invert DCLK output
9	HINV	R/W	HSYNC Inversion	0: Not invert HSYNC output 1: Invert HSYNC output
8	VINV	R/W	VSYNC Inversion	0: Not invert VSYNC output 1: Invert VSYNC output.
[7:4]	MODE	R/W	TFT1 Mode Selection	0000: Serial UPS051 : RGB mode 0001: Serial UPS052 : RGBD mode 0010: Serial CCIR656 0011: Parallel RGB mode. 1000: Memory mode: Command Write(RS=0) 1001: Memory mode: Command Read(RS=0) 1010: Memory mode: Data Write(RS=1) 1011: Memory mode: Data Read(RS=1) 1100: Memory mode: Data Show continuous For Display "continuous auto-send Data" from PPU mode 1101: Memory mode: Data Show once For Display data only "auto get 1 frame data" from PPU

Bit	Function	Type	Description	Condition
[3:1]	CLK_SEL	R/W	TFT1 Clock Selection	TFT1 controller R/W access clock,use with P_TFT_TS_MISC[7:6]: TFT_CLK_SEL={P_TFT_TS_MISC[6],P_TFT_TS_MISC[7],CLK_SEL[2:0]} 00000 (0): System Clock/1 00001 (1): System Clock/2 00010 (2): System Clock/3 . . 11111 (31): System Clock/32
0	TFT1EN	R/W	TFT1 Controller Enable	0: The TFT1 controller is disabled 1: The TFT1 controller is enabled

P_TFT1_V_PERIOD 0x93020144
TFT1 Vertical Period Register

Bit	15	14	13	12	11	10	9	8
Function	TFT1_V_Period							
Default	0	0	0	0	0	0	0	0
	TFT1_V_Period							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	-
[11:0]	TFT1_V_Period	R/W	TFT1 Vertical Period Register	Memory mode: Vertical period

P_TFT1_VS_WIDTH 0x93020148
TFT1 VSYNC Width Register

Bit	15	14	13	12	11	10	9	8
Function	TFT1_VS_Width							
Default	0	0	0	0	0	0	0	0
	TFT1_VS_Width							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]	-	-	Reserved	-
[5:0]	TFT1_VS_WIDTH	R/W	TFT1 VSYNC Width Register	Memory mode: Blanking time

P_TFT1_V_START 0x9302014C TFT1 Vertical Start Register

Bit	15	14	13	12	11	10	9	8
Function								

Default 0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TFT1_V_Start							

0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[15:12]	-	-	Reserved	-
[11:0]	TFT1_V_Start	R/W	TFT1 Vertical Start Register	Memory mode: Setup time

P_TFT1_V_END 0x93020150 TFT1 Vertical End Register

Bit	15	14	13	12	11	10	9	8
Function								

Default 0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TFT1_V_End							

0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[15:10]	-	-	Reserved	-
[11:0]	TFT1_V_End	R/W	TFT1 Vertical End Register	-

P_TFT1_H_PERIOD 0x93020154 TFT1 Horizontal Period Register

Bit	15	14	13	12	11	10	9	8
Function								

Default 0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TFT1_H_Period							

0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[15:13]	-	-	Reserved	-
[12:0]	TFT1_H_Period	R/W	TFT1 Horizontal Period Register	Memory mode: Horizontal period

P_TFT1_HS_WIDTH 0x93020158
TFT1 HSYNC Width Register

Bit	15	14	13	12	11	10	9	8
Function								TFT1_HS_Width
Default	0	0	0	0	0	0	0	0
TFT1_HS_Width								
0 0 0 0 0 0 0 0 0								

Bit	Function	Type	Description	Condition
[15:9]	-	-	Reserved	-
[8:0]	TFT1_HS_Width	R/W	TFT1 HSYNC Width Register	Memory mode: R/W pulse “low” period = (TFT1_HS_Width + 1)

Note: In memory mode, R/W pulse low/high setting has to lead the TFT1EN signal.

P_TFT1_H_START 0x9302015C
TFT1 Horizontal Start Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
TFT1_H_Start								
0 0 0 0 0 0 0 0 0								

Bit	Function	Type	Description	Condition
[15:13]	-	-	Reserved	-
[12:0]	TFT1_H_Start	R/W	TFT1 Horizontal Start Register	Memory mode: R/W pulse “high” period = (TFT1_H_Start + 1)

Note: In memory mode, R/W pulse low/high setting is greater than “0”, i.e., the minimum value is 2-cycle.

P_TFT1_H_END 0x93020160
TFT1 Horizontal End Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_H_End							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:13]	-	-	Reserved				-
[12:0]	TFT1_H_End	R/W	TFT1 Horizontal End Register				-

P_TFT1_LINE_RGB_ORDER 0x93020164
TFT1 Line RGB Order Control Register

Bit	15	14	13	12	11	10	9	8
Function	-	-	BYTE_ORDER	OEB_INV	RS_INV	XD_INV	SWAP	MASK_DCLK
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
MEM_CSB_MODE		RGB_Even_Field				RGB_Odd_Field	
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:14]	-	-	Reserved	-
13	BYTE_ORDER	R/W	Swap Byte Order of Byte Access of Memory Mode	0: Normal output 1: Reverse output
12	OEB_INV	R/W	OEB Reverse of Memory Mode	0: Normal output 1: Reverse output
11	RS_INV	R/W	RS Reverse of Memory Mode	0: Normal output 1: Reverse output
10	XD_INV	R/W	TFT1 Data Output Reverse	0: Normal output 1: Reverse output
9	SWAP	R/W	Cr SWAP in the CCIR656 Mode	0: No SWAP 1: SWAP
8	MASK_DCLK	R/W	Mask DCLK when MEM_SHOW enable in memory mode	0: Normal 1: Mask DCLK
7	MEM_CSB_MODE	R/W	CSB/POL Timing Control Enable Bit When this bit is set as '1', the low pulse width of CS signal will be the same as WE/OE signals which means the settings in "P_TFT1_H_START" and "P_TFT1_HS_WIDTH" will be invalid. This bit is only valid when using memory mode or TCON mode.	0: Disable 1: Enable

Bit	Function	Type	Description	Condition
[6:4]	RGB_Even_Field	R/W	RGB Order in Even Field	000: RGB 001: RBG 010: GRB 011: GBR 100: BRG 101: BGR
3	Debug_En	R/W	Output TFT1_REQ to TFT1_DATA[0]	0: Turn off 1: Turn on
[2:0]	RGB_Odd_Field	R/W	RGB Order in Odd Field	000: RGB 001: RBG 010: GRB 011: GBR 100: BRG 101: BGR

P_TFT1_STATUS 0x93020168
TFT1 Status Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
HSTS		VSTS		-		INT	-
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:6]	HSTS	R	Horizontal Status Register	MEM_STATE[3:0] when memory mode
[5:4]	VSTS	R	Vertical Status Register	_idle = 4'b0000; _blanking = 4'b0001; _setup = 4'b0010; _active_CLK_H = 4'b0100; _active_CLK_L = 4'b0101; _active_wait = 4'b0110; _cmd_read_H = 4'b1000; _cmd_read_L = 4'b1001; _cmd_write_H = 4'b1010; _cmd_write_L = 4'b1011;

Bit	Function	Type	Description	Condition
[3:2]	-	-	Reserved	-
1	-	-	Reserved	
0	-	-	Reserved	

P_TFT1_MEM_BUFF_WR 0x9302016C TFT1 Memory Buffer Write Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
TFT1_MEM_BUFFER_WR								
	7	6	5	4	3	2	1	0
TFT1_MEM_BUFFER_WR								
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TFT1_MEM_BUFFER_WR	W	Buffer of memory mode for Write command data	-

P_TFT1_MEM_BUFF_RD 0x93020170 TFT1 Memory Buffer Read Register

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
TFT1_MEM_BUFFER_RD								
	7	6	5	4	3	2	1	0
TFT1_MEM_BUFFER_RD								
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TFT1_MEM_BUFFER_RD	R	Buffer of memory mode for Read command data	-

P_TFT1_TE_CTRL 0x93020180 TFT1 TE Control Register

Bit	31	30	29	28	27	26	25	24
Function								
Default	0	0	0	0	0	0	0	0
TFT1_TE_WR_HS_CNT								
	23	22	21	20	19	18	17	16
TF1_TE_MODE								
	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	TF1_TE_EN

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TF1_TE_EN

Bit	Function	Type	Description	Condition
[28:24]	TFT1_TE_WR_HS_CNT	R/W	Starting memory write after the count of horizontal sync pulse	
16	TFT_TE_MODE	R/W	Tearing Effect Mode selection 0: MCU write is faster than panel read 1: MCU write is slower than panel read	
8	TFT_TE_RGB888_EN	R/W	Tearing Effect RGB888 Enable 0: RGB565 1: RGB888	
0	TFT1_TE_EN	R/W	Tearing Effect Enable 0: Disable 1: Enable	

P_TFT1_TE_HSYNC_CNT 0x93020184
TFT1 TE HSYNC CNT Register

Bit	23	22	21	20	19	18	17	16
Function								
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	TF1_TE_HSYNC_CNT

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[17:0]	TFT1_TE_HSYNC_CNT	R/W	Hsync pulse width. SW should set the value just larger than HYSC pulse Unit:1/system clock	

P_TFT1_VS_START 0x930201B0 TFT1 Vertical Show Start Register

Bit	15	14	13	12	11	10	9	8
-----	----	----	----	----	----	----	---	---

Function TFT1_V_SHOW_Start

Default 0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

TFT1_V_SHOW_Start

0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description				Condition
[15:0]	TFT1_V_SHOW_Start	R/W	Show Window Register for Vertical Show Start Address				Default: 0 = Disable

P_TFT1_VS_END 0x930201B4 TFT1 Vertical Show End Register

Bit	15	14	13	12	11	10	9	8
-----	----	----	----	----	----	----	---	---

Function TFT1_V_SHOW_End

Default 0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

TFT1_V_SHOW_End

0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description				Condition
[15:0]	TFT1_V_SHOW_End	R/W	Show Window Register for Vertical Show End Address				Default: 0 = Disable

P_TFT1_HS_START 0x930201B8 TFT1 Horizontal Show Start Register

Bit	15	14	13	12	11	10	9	8
-----	----	----	----	----	----	----	---	---

Function TFT1_H_SHOW_Start

Default 0 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

TFT1_H_SHOW_Start

0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description				Condition
[15:0]	TFT1_H_SHOW_Start	R/W	Show Window Register for Horizontal Show Start Address				Default: 0 = Disable

P_TFT1_HS_END 0x930201BC TFT1 Horizontal Show End Register

Bit	15	14	13	12	11	10	9	8
Function	TFT1_H_SHOW_End							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_H_SHOW_End							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	TFT1_H_SHOW_End	R/W	Show Window Register for Horizontal Show End Address				Default: 0 = Disable

P_TFT1_CLIP_V_START 0x930203B0 TFT1 Vertical Clip Start Register

Bit	15	14	13	12	11	10	9	8
Function	TFT1_Clip_V_Start							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_Clip_V_Start							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:10]	-	-	Reserved				-
[9:0]	TFT1_Clip_V_Start	R/W	Vertical Start Position for Panel Display				-

P_TFT1_CLIP_V_END 0x930203B4 TFT1 Vertical Clip End Register

Bit	15	14	13	12	11	10	9	8
Function	TFT1_Clip_V_End							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_Clip_V_End							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:10]	-	-	Reserved				-
[9:0]	TFT1_Clip_V_End	R/W	Vertical End Position for Panel Display				-

P_TFT1_CLIP_H_START 0x930203B8
TFT1 Horizontal Clip Start Register

Bit	15	14	13	12	11	10	9	8
Function	-							TFT1_Clip_H_Start
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_Clip_H_Start							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:13]	-	-	Reserved					-
[12:0]	TFT1_Clip_H_Start	R/W	Horizontal Start Position for Panel Display					-

P_TFT1_CLIP_H_END 0x930203BC
TFT1 Horizontal Clip End Register

Bit	15	14	13	12	11	10	9	8
Function	-							TFT1_Clip_H_End
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_Clip_H_End							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:13]	-	-	Reserved					-
[12:0]	TFT1_Clip_H_End	R/W	Horizontal End Position for Panel Display					-

P_TFT1_TS_CKV 0x930203C0
TFT1 CKV Start Position Register

Bit	15	14	13	12	11	10	9	8
Function	-							TFT1_TsCKV
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_TsCKV							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:10]	-	-	Reserved					-
[9:0]	TFT1_TsCKV	R/W	Horizontal Start Position of CKV Signal					-

P_TFT1_TW_CKV
0x930203C4
TFT1 CKV Width Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_TwCKV							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]	-	-	Reserved	-
[9:0]	TFT1_TwCKV	R/W	Pulse Width of CKV Signal	-

P_TFT1_TS_MISC
0x930203C8
TFT1 Misc Control Register

Bit	23	22	21	20	19	18	17	16
Function	De_dly_reg							
Default	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
DCLK_SEL	Gamam_en	-	DAAMOD_i	Dith_mod	Dith_en		
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT_CLK_SEL[3]	TFT_CLK_SEL[4]	REG_REV	REG_POL	-	TsSTV_1		
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[23:22]	De_dly_reg	R/W	TFT1_DE postpone selection based on DELC cell	-
21	Slide_en	R/W	Enable sliding window function	-
[20:18]	Delay_reg	R/W	TFT1_DCLK shift selection based on DELC cell	-
[17:16]	-			-
[15:14]	DCLK_SEL	R/W	TFT1 DCLK shift selection based on system clock	0/1/2/3: Shift pixel clock by 1/2/3/4 system clock
13	Gamma_en	R/W	Enable gamma function	-
12	-			-
[11:10]	DATAMOD_i	R/W	0/1/2 for RGB 8/565/666-bit setting	-
9	Dith_mod	R/W	0/1 is for 8-to-6 / 8-to-4-bit dithering	-
8	Dith_en	R/W	Enable dithering function	-
7	TFT_CLK_SEL[3]	R/W	Use with P_TFT1_CTRL[3:1] to select TFT clock	-
6	TFT_CLK_SEL[4]	R/W	Use with P_TFT1_CTRL[3:1] to select TFT clock	-
5	REG_REV	R/W	Reverse of TFT1 data	-
4	REG_POL	R/W	Polarity selection in TCON mode	Must be tied to 0
3	-	R/W	Reserved	-
[2:0]	TsSTV_1	R/W	Vertical start position based on the active area in TCON mode	-

P_TFT1_TS_POL 0x930203CC TFT1 POL Start Position Register

Bit	15	14	13	12	11	10	9	8
Function	-		TFT1_TsPOL					
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_TsPOL							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:13]	-	-	Reserved					-
[12:0]	TFT1_TsPOL	R/W	Horizontal Start Position of POL Signal					-

P_TFT1_TS_STV 0x930203D0 TFT1 STV Start Position Register

Bit	15	14	13	12	11	10	9	8
Function	-							

Default 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TFT1_TsSTV_2							

0 0 0 0 0 0 0 0

Bit	Function	Type	Description					Condition
[15:13]	-	-	Reserved					-
[12:0]	TFT1_TsSTV_2	R/W	Horizontal Start Position of STV Signal					-

P_TFT1_TW_STV 0x930203D4 TFT1 STV Pulse Width Register

Bit	15	14	13	12	11	10	9	8
Function	-							

Default 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TFT1_TwSTV							

0 0 0 0 0 0 0 0

Bit	Function	Type	Description					Condition
[15:10]	-	-	Reserved					-
[9:0]	TFT1_TwSTV	R/W	Pulse Width Of STV Signal					-

P_TFT1_TS_STH 0x930203D8 TFT1 STH Start Position Register

Bit	15	14	13	12	11	10	9	8
Function	-							

Default 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
TFT1_TsSTH							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:13]	-	-	Reserved				-
[12:0]	TFT1_TsSTH	R/W	Horizontal Start Position of STH Signal				-

P_TFT1_TW_STH 0x930203DC TFT1 STH Pulse Width Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_TwSTH							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:13]	-	-	Reserved				-
[12:0]	TFT1_TwSTH	R/W	Pulse Width of the STH Signal				-

P_TFT1_TS_OEV 0x930203E0 TFT1 OEV Start Position Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_TsOEV							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:10]	-	-	Reserved				-
[9:0]	TFT1_TsOEV	R/W	Start Position of CKV Signal				-

P_TFT1_TW_OEV 0x930203E4 TFT1 OEV Pulse Width Register

Bit	15	14	13	12	11	10	9	8
Function	-						TFT1_TwOEV	
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_TwOEV							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:10]	-	-	Reserved					-
[9:0]	TFT1_TwOEV	R/W	Pulse Width of the OEV Signal					-

P_TFT1_TS_LD 0x930203E8 TFT1 LD Start Position Register

Bit	15	14	13	12	11	10	9	8
Function	-						TFT1_TsLD	
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_TsLD							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:10]	-	-	Reserved					-
[9:0]	TFT1_TsLD	R/W	Start Position of the LD Signal					-

P_TFT1_TW_LD 0x930203EC TFT1 LD Pulse Width Register

Bit	15	14	13	12	11	10	9	8
Function	-						TFT1_TwLD	
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_TwLD							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:10]	-	-	Reserved					-
[9:0]	TFT1_TwLD	R/W	Pulse Width of the LD Signal					-

P_TFT1_DITHER_TAB0 0x930203F0 TFT1 Dithering Parameter0 Register

Bit	15	14	13	12	11	10	9	8
Function	TFT1_DITHER_TAB0							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_DITHER_TAB0							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	TFT1_DITHER_TAB0	R/W	Dithering Parameter 0					-

P_TFT1_DITHER_TAB1 0x930203F4 TFT1 Dithering Parameter1 Register

Bit	15	14	13	12	11	10	9	8
Function	TFT1_DITHER_TAB1							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_DITHER_TAB1							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	TFT1_DITHER_TAB1	R/W	Dithering Parameter 1					-

P_TFT1_DITHER_TAB2 0x930203F8

TFT1 Dithering Parameter2 Register

Bit	15	14	13	12	11	10	9	8
Function	TFT1_DITHER_TAB2							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TFT1_DITHER_TAB2							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TFT1_DITHER_TAB2	R/W	Dithering Parameter 2	-

P_TFT1_TAB3

0x930203FC

TFT1 Dithering Parameter3 Register

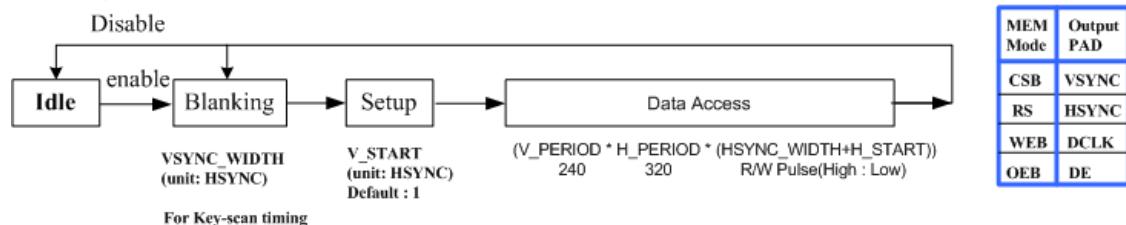
Bit	15	14	13	12	11	10	9	8
Function	TFT1_DITHER_TAB3							
Default	0	0	0	0	0	0	0	0

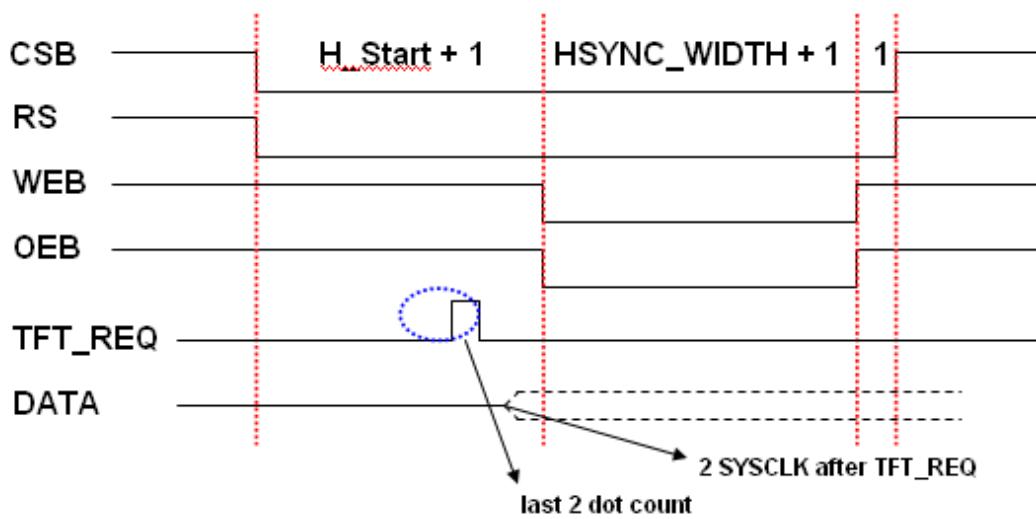
7	6	5	4	3	2	1	0
TFT1_DITHER_TAB3							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TFT1_DITHER_TAB3	R/W	Dithering Parameter 3	-

24.6 Programming Guide

Memory mode state machine:




Ex. Using 128X160 panel with memory interface to connect the TFT1 controller

```

Req(WR, 0x93020144, 240-1); // V_PERIOD = 240
Req(WR, 0x93020148, 20); // V_WIDTH = 20 = Blanking
Req(WR, 0x9302014C, 1); // V_START = 1 = Setup

Req(WR, 0x93020154, 320-1); // H_PERIOD = 320
Req(WR, 0x93020158, 3); // H_WIDTH = 4T = WEB/OEB pulse low
Req(WR, 0x9302015C, 3); // H_START = 4T = WEB/OEB pulse high

Req(WR, 0x930201B0, 0+y); // V_SHOW_START
Req(WR, 0x930201B4, 160-1+y); // V_SHOW_END
Req(WR, 0x930201B8, 0+x); // H_SHOW_START
Req(WR, 0x930201BC, 128-1+x); // H_SHOW_END

```

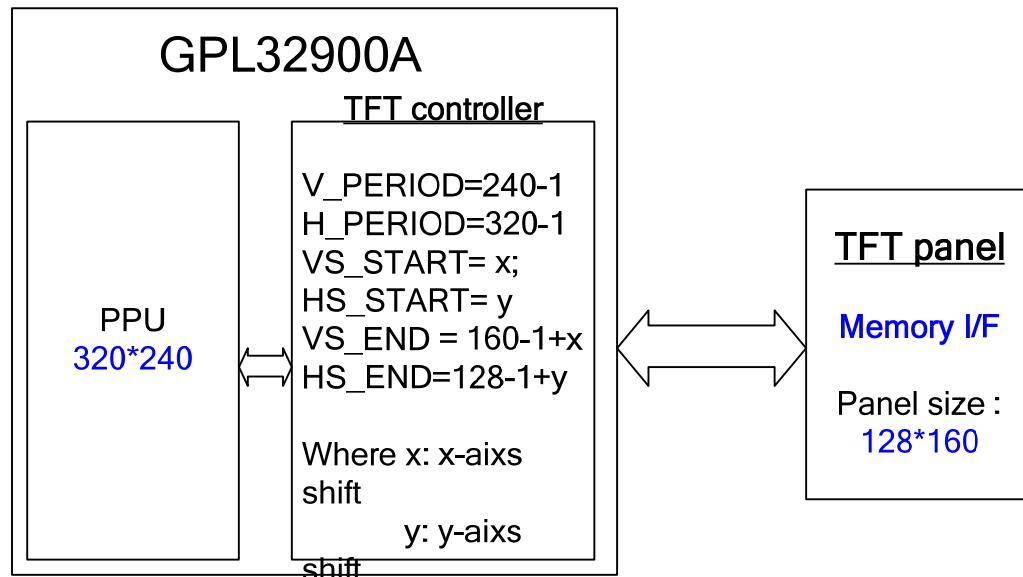
Note: x/y is the value of shift show window

Ex. When programmers want to show a 320x240(HxV) picture over 128x160, the following register setting are required.

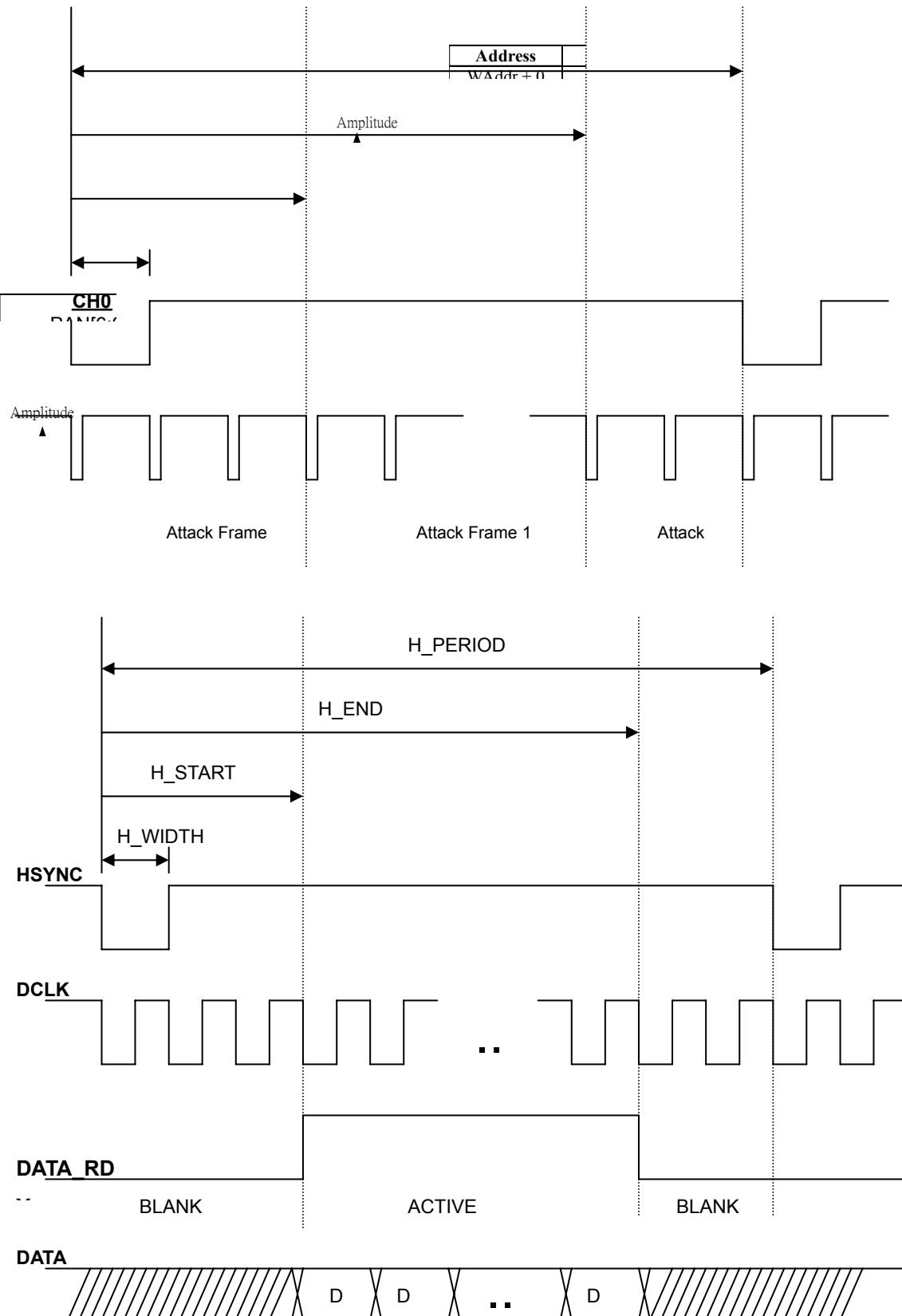
V_PERIOD = 240 -1
 H_PERIOD = 320 -1
 V_START = 1 (Setup time)
 VSYNC_WIDTH = 20 (Blanking time)
 H_START = 3 (WEB/OEV is 4T offset from CSB)
 HSYNC_WIDTH = 3(WEB/OEV is 4T width)

```
V_SHOW_START = 0  
V_SHOW_END = 160  
H_SHOW_START = 0  
H_SHOW_END = 128 - 1
```

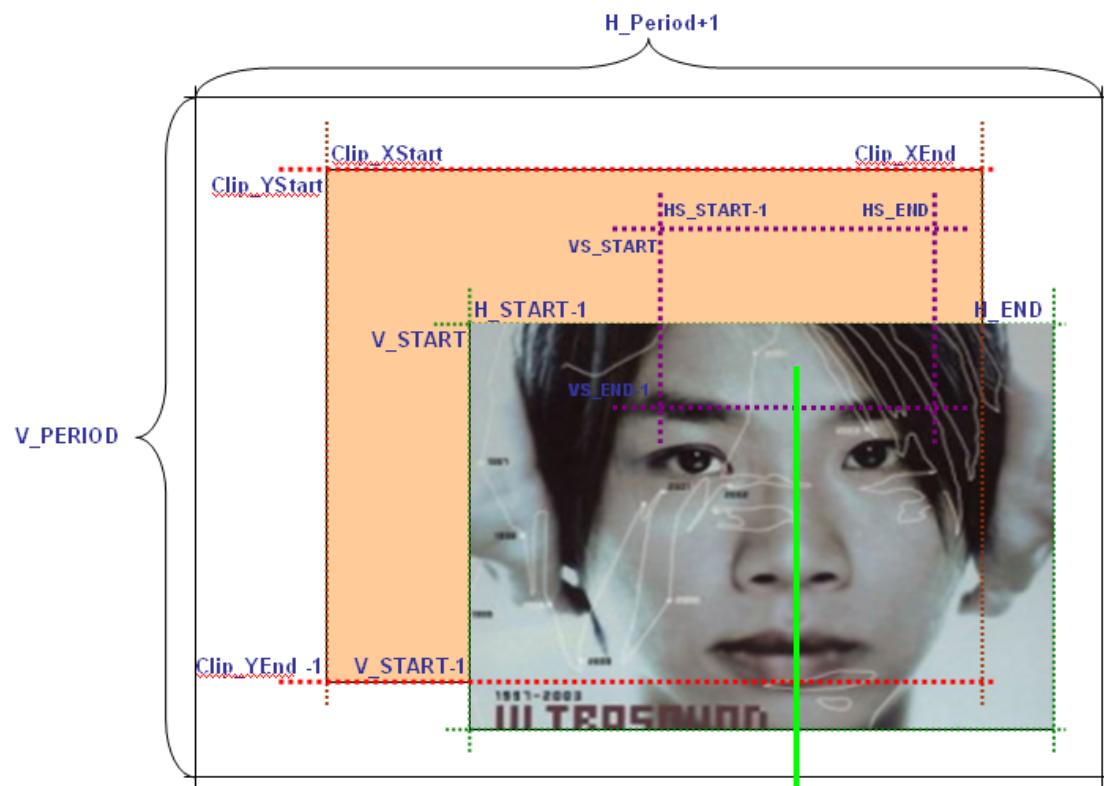
By the setting, panel will show the [0:128-1] X [0:160-1] area of the picture.



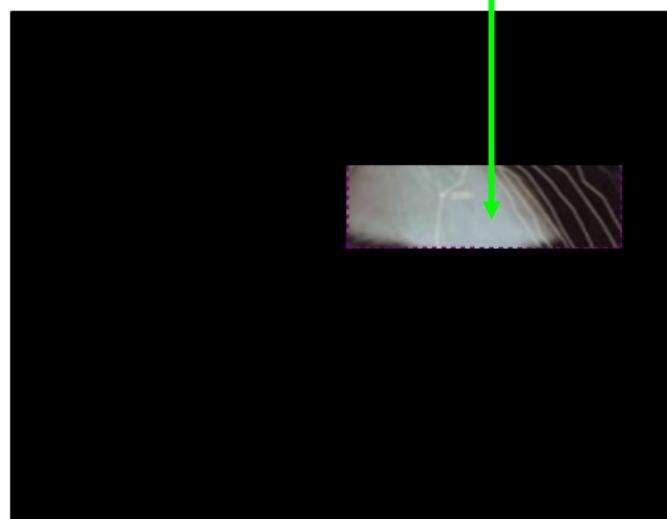
24.7 TFT1 Timing Diagram



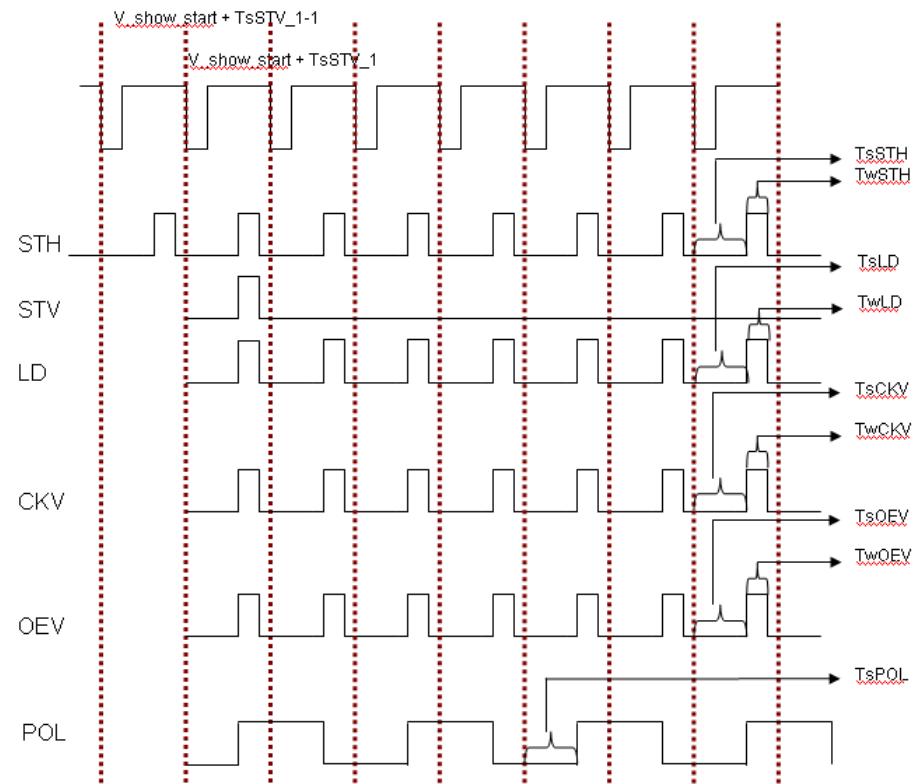
About resolution configuration



With such configuration, it will show on the panel with the following picture



24.8 TCON Timing Diagram



25 STN

25.1 Introduction

The GPL32900A supports general STN LCD interface with dithering engine built-in, which supports two different color modes: 16-gray-level mode and B/W (mono) mode. The STN resolution is configurable and a clipping function is available in order to support partial windows from PPU output (320x240 frame size in general).

25.2 Features

- Supports 512 x 512 x 1 black/white display (in B/W mode)
- Supports 512 x 512 x (4/3-bit) gray-level display (in Dithering mode)
- Configurable data bus : 1/2/4/8-bit
- B-type/C-type modulation
- Configurable CP wait cycle to adjust frame rate
- Dithering function inside to generate gray-level patterns
- Clipping function
- Configurable horizontal/vertical blanking time
- The RGB-to-YUV calculation
- CP / FP/ LP signal polarity output selection

25.3 STN Interface Configuration

Name	I/O	Description
STNFM	O	Frame Modulation Signal (shared with TFT1 DE)
STNFP	O	Frame Rate Signal (shared with TFT1 VSYNC)
STNLP	O	Line Scan Signal (shared with TFT1 HSYNC)
STNCP	O	Shifting Clock Signal (shared with TFT1 CLK)
STND0	O	STN LCD Data 0 (shared with DISP_OUT[0])
STND1	O	STN LCD Data 1 (shared with DISP_OUT[1])
STND2	O	STN LCD Data 2 (shared with DISP_OUT[2])
STND3	O	STN LCD Data 3 (shared with DISP_OUT[3])
STND4	O	STN LCD Data 4 (shared with DISP_OUT[4])
STND5	O	STN LCD Data 5 (shared with DISP_OUT[5])
STND6	O	STN LCD Data 6 (shared with DISP_OUT[6])
STND7	O	STN LCD Data 7 (shared with DISP_OUT[7])

The STNFM, STNFP, STNLP, and STNCP are synchronous signals used for transferring LCD display data from GPL32900A to the GPLD or SPLC series chips. The STND [7:0] is a data port, responsible for carrying transferred LCD data.

25.4 Register Summary

Name	Address	Description
P_STN_CTRL0	0x9302017C	STN Control Register 0
P_STN_SEG	0x93020200	STN Segment Register
P_STN_COM	0x93020204	STN Column Register
P_STN_PIC_COM	0x93020208	STN Picture Column Register
P_STN_CPWAIT	0x9302020C	STN CP Wait Register
P_STN_CTRL1	0x93020210	STN Control Register 1
P_STN_GTG_SEG	0x93020214	STN Global Timing Generator Segment Register
P_STN_GTG_COM	0x93020218	STN Global Timing Generator Column Register
P_STN_SEG_CLIP	0x9302021C	STN Clipping Start Segment Register
P_STN_COM_CLIP	0x93020144	STN Clipping Start Column Register

25.5 Register Definition

25.5.1 Control Registers

P_STN_CTRL0		0x9302017C								STN Control Register 0							
Bit	Function	15	14	13	12	11	10	9	8	-	-	-	-	-	-	-	-
Default		0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-
		7	6	5	4	3	2	1	0	Reserved	DPS	EN					
		0	0	0	0	0	0	0	0								

Bit	Function	Type	Description				Condition			
[15:1]	-	-	Reserved				-			
1	DPS	R/W	STN Data Polarity Selection				0: Black is 1, white is 0 (STN general case) 1: Black is 0, white is 1			
0	EN	R/W	STN Enable Bit				0: Disable 1: Enable			

P_STN_SEG
0x93020200
STN Segment Register

Bit	15	14	13	12	11	10	9	8
Function	-							STN_SEG
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
STN_SEG							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:9]	-	-	Reserved					-
[8:0]	STN_SEG	R/W	STN Segment Number (horizontal resolution)					-

P_STN_COM
0x93020204
STN Column Register

Bit	15	14	13	12	11	10	9	8
Function	-							STN_COM
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
STN_COM							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:9]	-	-	Reserved					-
[8:0]	STN_COM	R/W	STN Column Number (vertical resolution)					-

P_STN_PIC_COM
0x93020208
STN Picture Column Register

Bit	15	14	13	12	11	10	9	8
Function	-							STN_PIC_COM
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
STN_PIC_COM							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:9]	-	-	Reserved	-
[8:0]	STN_PIC_COM	R/W	STN Column Number for Picture Frame (vertical resolution) STN Segment Number for Picture Frame is fixed to be 320 (horizontal resolution)	-

P_STN_CPWAIT
0x9302020C
STN CP Wait Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0
STN_CPWAIT								
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:0]	STN_CPWAIT	R/W	STN CP Wait cycle for frame rate adjustment	-

P_STN_CTRL1
0x93020210
STN Control Register 1

Bit	15	14	13	12	11	10	9	8
Function	CTYPENUM							
Default	0	0	0	0	0	0	0	0
CLIP Y3B LFTYPE SPOL BCTYPE DTEN LDNUM								
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	CTYPENUM	R/W	Number of frame to exchange FM signal if C type modulation is active.	-
7	CLIP	R/W	Clipping function enable	0: Disable 1: Enable
6	Y3B	R/W	3-bit gray-level enable	0: 16-gray-level 1: 8-gray-level

Bit	Function	Type	Description	Condition
5	LFTYPE	R/W	LP / FP Signal Type	0: Short time delay between CP and LP/FP 1: Long time delay between CP and LP/FP
4	SPOL	R/W	The Polarity of CP / LP / FP signal	0: Non-inverse 1: Inverse
3	BCTYPE	R/W	C Type Modulation Active	0: B type 1: C type
2	DTEN	R/W	Dithering Function Enable	0: B/W mode 1: Gray level mode
[1:0]	LDNUM	R/W	Data Bus Mode	0: 1-bit 1: 2-bit 2: 4-bit 3: 8-bit

P_STN_GTG_SEG 0x93020214 STN Global Timing Generator Segment Register

Bit	15	14	13	12	11	10	9	8
Function	-							STN_GTG_SEG
Default	0	0	0	0	0	0	0	0
STN_GTG_SEG								
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:9]	-	-	Reserved	-
[8:0]	STN_GTG_SEG	R/W	STN Segment number for Global Timing Generator (horizontal resolution)	-

P_STN_GTG_COM 0x93020218 STN Global Timing Generator Column Register

Bit	15	14	13	12	11	10	9	8
Function	-							STN_GTG_COM
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
STN_GTG_COM							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:9]	-	-	Reserved					-
[8:0]	STN_GTG_COM	R/W	STN Column number for Global Timing Generator (vertical resolution)					-

P_STN_SEG_CLIP 0x9302021C STN Clipping Start Segment Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
STN_CLI_SEG							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:9]	-	-	Reserved					-
[8:0]	STN_CLI_SEG	R/W	STN start segment number for clipping (horizontal resolution)					-

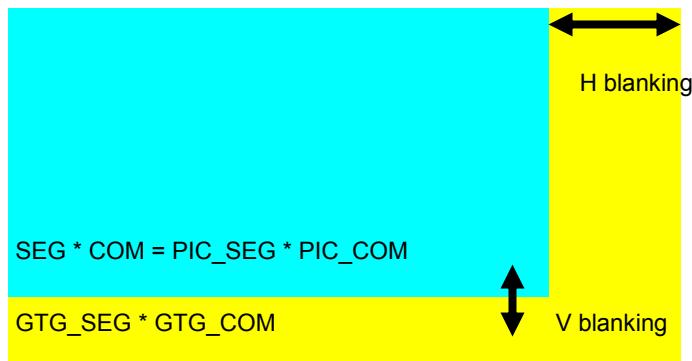
P_STN_COM_CLIP 0x93020144 STN Clipping Start Column Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0

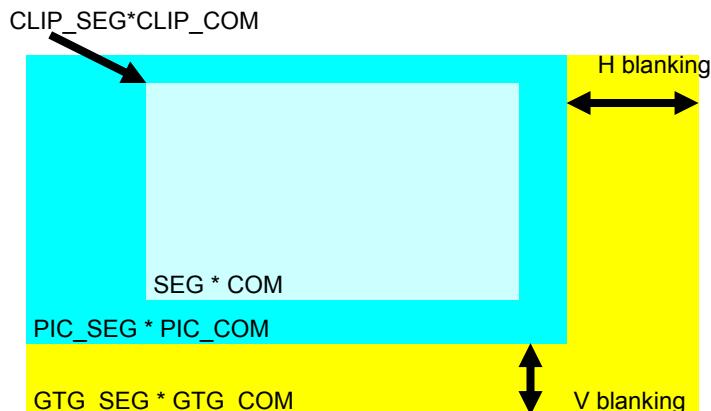
7	6	5	4	3	2	1	0
STN_CLI_COM							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:9]	-	-	Reserved					-
[8:0]	STN_CLI_COM	R/W	STN start column number for clipping (vertical resolution)					-

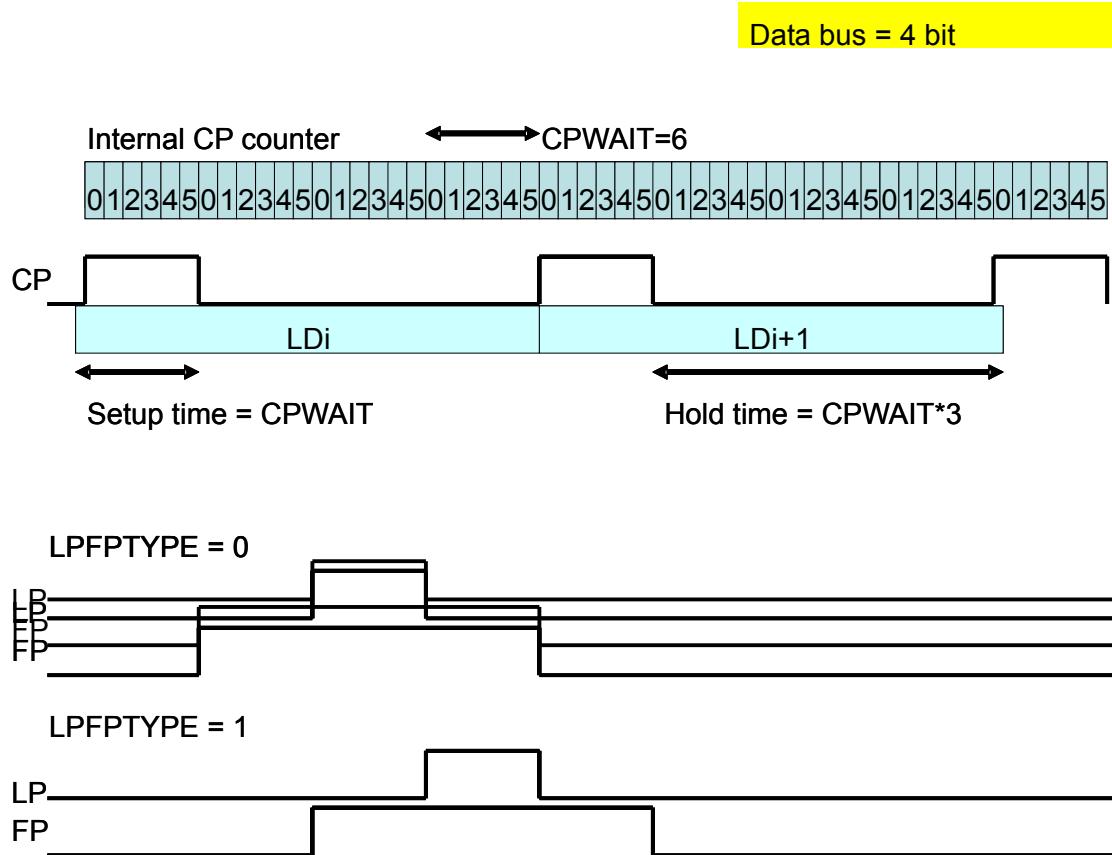
25.5.2 STN Panel without the Clipping Function

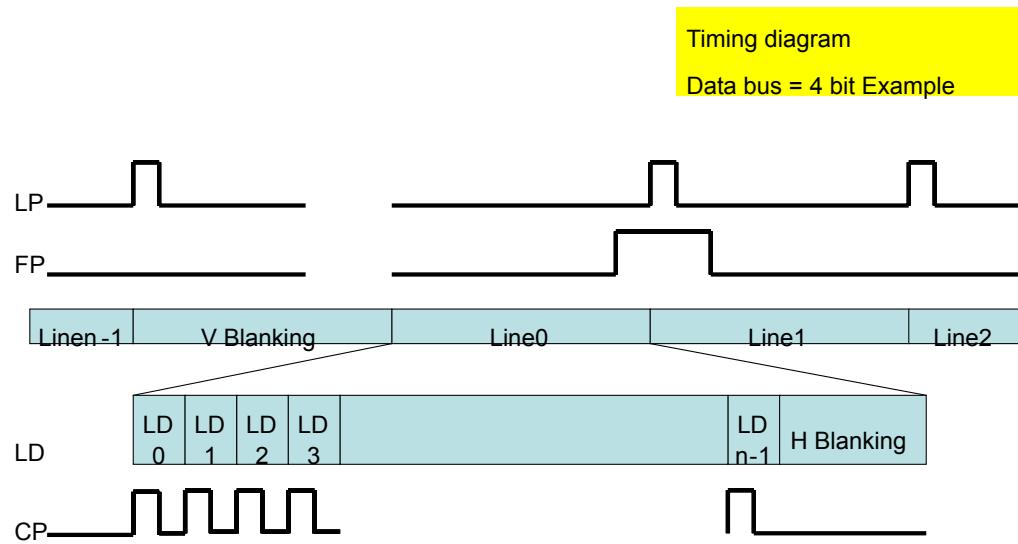
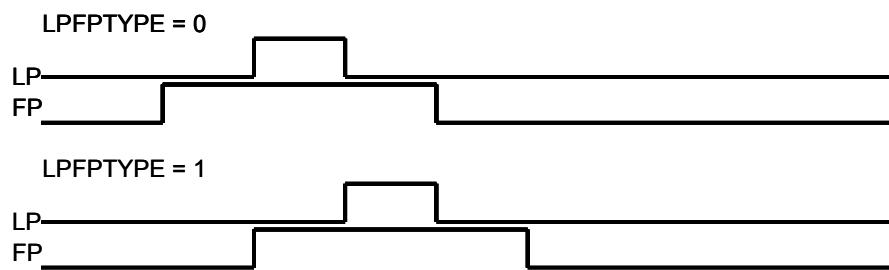
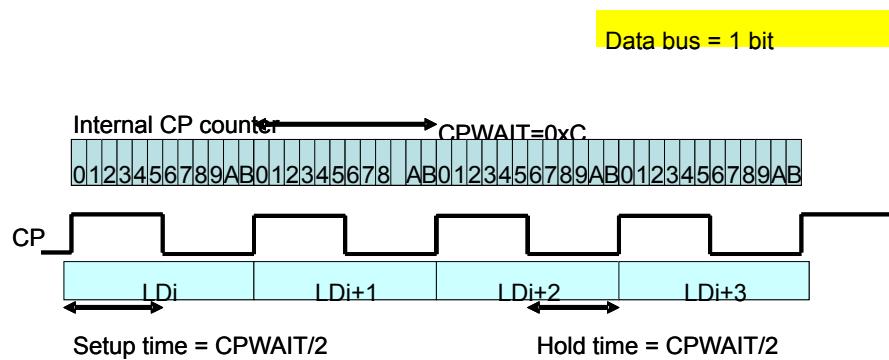


25.5.3 STN Panel with Clipping Function Enabled



25.5.4 STN Timing





26 DC-DC

26.1 Introduction

The GPL32900A supports 2-channel DC-DC Boost control circuit for LED Backlight and VGH/VGL voltage generator of TFT-LCD drivers to reduce the cost of application circuit.

26.2 Features

- DC-DC PWM0 for VGH/VGL and DC-DC PWM1 for backlight.
- Fixed 600kHz switch frequency.
- The DC-DC PWM0 is adjustable output voltage.
- The DC-DC PWM1 supports 7 setp of VREF which can adjust brightness for backlight.

26.3 Register Summary

Name	Address	Description
P_SCUB_PWM_CTRL	0x90005140	DC-DC PWM Control

26.4 Register Definition

P_SCUB_PWM_CTRL		0x90005140								DC-DC PWM Control									
Bit	31	30	29	28	27	26	25	24											
Function	-																		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	23	22	21	20	19	18	17	16											
	-																		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8											
	-																		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0											
				PWM1_VSET	-	PWM1_EN	PWM0_EN	CLK6M_EN											
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:7]	Reserved	-		
[6:4]	PWM1_VSET	R/W	PWM1 VREF Voltage Selection 000: 0.1V 001: 0.1V 010: 0.15V 011: 0.2V 100: 0.25V 101: 0.3V 110: 0.35V 111: 0.4V	
3	Reserved	-		
2	PWM1_EN	R/W	PWM1 Enable 0 = Disable 1 = Enable	
1	PWM0_EN	R/W	PWM0 Enable 0 = Disable 1 = Enable	
0	CLK6M_EN	R/W	PWM CLK 6M Enable 0 = Disable 1 = Enable	

27 TV0 Encoder Interface

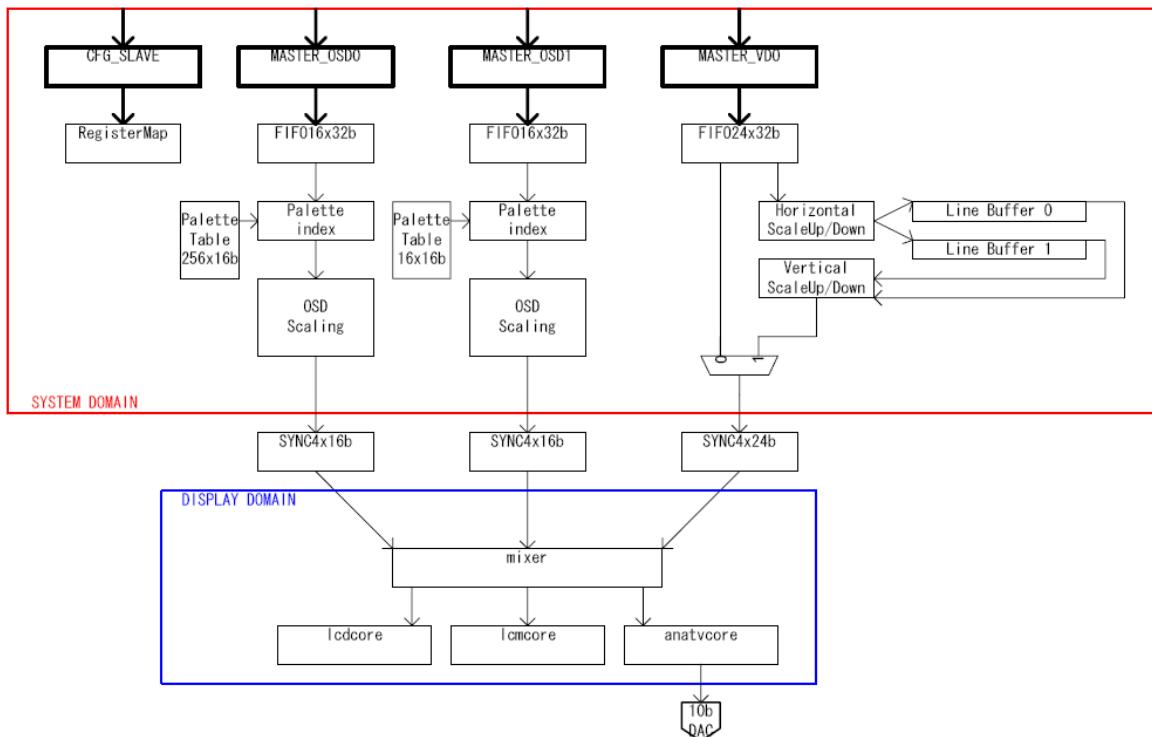
27.1 Introduction

This NTSC/PAL encoder is designed for video systems requiring composite, S-video (Y/C, and component YPbPr video signals. It converts RGB/YCbCr color component signals into their corresponding luminance and chrominance signals in accordance with either NTSC or PAL standards. Worldwide video standards are supported, including NTSC-M (North America, Taiwan), NTSC-J (Japan), PAL-B,D,G,H,I (Europe, Asia), PAL-M (Brazil), PAL-N (Uruguay, Paraguay), PAL-Nc (Argentina), PAL-60, NTSC-443.

27.2 Features

- Clock Input – 27Mhz (x2 upsampling)
- Composite NTSC-M/J/443, PAL-B/D/G/H/I/M/N/Nc/60
- S-Video (Y/C)
- Component YPbPr
- Chroma Low-pass Filter 1.5Mhz
- Non-Interlace Output (262/312)
- Progressive Output (525p/625p)
- 10-bit DAC interface
- Optional 7.5% Pedestal
- Output Amplitude Selectable

27.3 Block Diagram



27.4 TVE0 Registers

Name	Address	Description
P_TVE0_OUTRESOLUTION	0x93000000	TV output resolution
P_TVE0_BLANKVDIR	0x93000004	TV Blanking (Vertical)
P_TVE0_BLANKHDIR	0x93000008	TV Blanking (Horizontal)
P_TVE0_BLANKDATA	0x9300000C	TV Blanking Pattern
P_TVE0_FRAMEADR	0x93000010	TV Frame Buffer Address
P_TVE0_FRAMEPITCH	0x93000014	TV Frame Buffer Pitch (Byte)
P_TVE0_FRAMERESOLUTION	0x93000018	TV Frame Resolution
P_TVE0_SCALERESOLUTION	0x9300001C	TV Scale Resolution
P_TVE0_SCALEHFACTOR	0x93000020	TV Scale Parameter (Horizontal)
P_TVE0_SCALEVFACTOR	0x93000024	TV Scale Parameter (Vertical)
P_TVE0_SCALECONTROL	0x93000028	TV Scale Control
P_TVE0 OSD0BASEADR	0x93000030	TV OSD0 Base Address
P_TVE0 OSD0PITCH	0x93000034	TV OSD0 Pitch (Byte)
P_TVE0 OSD0RESOLUTION	0x93000038	TV OSD0 Resolution
P_TVE0 OSD0STARTXY	0x9300003C	TV OSD0 Starting Coordinate
P_TVE0 OSD0FORMAT	0x93000040	TV OSD0 Format
P_TVE0 OSD0CONTROL	0x93000044	TV OSD0 Control
P_TVE0 OSD1FORMAT	0x93000048	TV OSD1 Format
P_TVE0 OSD1CONTROL	0x9300004C	TV OSD1 Control
P_TVE0 OSD1BASEADR	0x93000050	TV OSD1 Base Address
P_TVE0 OSD1PITCH	0x93000054	TV OSD1 Pitch (Byte)
P_TVE0 OSD1RESOLUTION	0x93000058	TV OSD1 Resolution
P_TVE0 OSD1STARTXY	0x9300005C	TV OSD1 Starting Coordinate
P_TVE0 OSD0HFACTOR	0x93000060	TV OSD0ScaleParameter(Horizontal)
P_TVE0 OSD0VFACTOR	0x93000064	TV OSD0ScaleParameter(Vertical)
P_TVE0 OSD0VPARAM1	0x93000068	TV OSD0ScaleParameter(field 1)
P_TVE0 OSD0SCLWH	0x9300006C	TV OSD0ScaleResolution
P_TVE0 OSD1HFACTOR	0x93000070	TV OSD1ScaleParameter(Horizontal)
P_TVE0 OSD1VFACTOR	0x93000074	TV OSD1ScaleParameter(Vertical)
P_TVE0 OSD1VPARAM1	0x93000078	TV OSD1ScaleParameter(field 1)
P_TVE0 OSD1SCLWH	0x9300007C	TV OSD1ScaleResolution
P_TVE0_INTRQENABLE	0x93000080	TV Interrupt Enable (Mask)
P_TVE0_INTRQSOURCE	0x93000084	TV Interrupt Source

Name	Address	Description
P_TVE0_CBARCONTROL	0x93000090	TV Color-bar Control
P_TVE0_CBARCOLOR	0x93000094	TV Color-bar Single Color
P_TVE0_COLORMATRIX0	0x930000A0	TV Color Matrix Parameter 0
P_TVE0_COLORMATRIX1	0x930000A4	TV Color Matrix Parameter 1
P_TVE0_COLORMATRIX2	0x930000A8	TV Color Matrix Parameter 2
P_TVE0_COLORMATRIX3	0x930000AC	TV Color Matrix Parameter 3
P_TVE0_COLORMATRIX4	0x930000B0	TV Color Matrix Parameter 4
P_TVE0_COLORMATRIX5	0x930000B4	TV Color Matrix Parameter 5
P_TVE0_TVEAMPADJ	0x930000D8	TV Amplitude Adjust
P_TVE0_TVEPOSADJ	0x930000DC	TV Position Adjust
P_TVE0_LCMACTIME	0x930000E0	TV LCM AC Timing
P_TVE0_DISPFORMAT	0x930000F0	TV Display Format
P_TVE0_DISPCONTROL	0x930000FC	TV Display Control
P_TVE0_PALETTEPTR	0x93000800	TV Pallette Pointer
P_TVE0_LCMPROGRAMIO	0x93000F00	TV LCM Programming In-out

27.5 Register Definition

27.5.1 Output TV0 Resolution Register

P_TVE0_OUTRESOLUTION 0x93000000								TV0 output resolution								
Bit	31	30	29	28	27	26	25	24	RES_WIDTH[27:24]							
Function	-	-	-	-	RES_WIDTH[27:24]								RES_WIDTH[27:24]			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RES_WIDTH[27:24]								RES_WIDTH[27:24]								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RES_HEIGHT[11:8]								RES_HEIGHT[11:8]								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RES_HEIGHT[7:0]								RES_HEIGHT[7:0]								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description		Condition
31	-	-	Reserved		-
30	-	-	Reserved		-
29	-	-	Reserved		-
28	-	-	Reserved		-
[27:16]	RES_WIDTH	R/W	The Width of Active TV 0		-
15	-	-	Reserved		-
14	-	-	Reserved		-
13	-	-	Reserved		-
12	-	-	Reserved		-
[11:0]	RES_HEIGHT	R/W	The Height of Active TV 0		-

LIMIT:

** RES_WIDTH = BLANKLEFT + BLANKRIGHT + ACT_WIDTH (or SCL_WIDTH)

** RES_HEIGHT = BLANKTOP + BLANKBOTTOM + ACT_HEIGHT(or SCL_HEIGHT)

27.5.2 TV0 Blanking (Vertical) Register

P_TVE0_BLANKVDIR
0x93000004
TV0 Blanking (Vertical)

Bit	31	30	29	28	27	26	25	24				
Function	-	-	-	-	BLANKTOP[27:24]							
Default	0	0	0	0	0	0	0	0				

23	22	21	20	19	18	17	16
BLANKTOP[23:16]							

0	0	0	0	0	0	0	0			
BLANKTOP[23:16]										
15	14	13	12	11	10	9	8			
-	-		-	BLANKBOTTOM[11:8]						
0	0	0	0	0	0	0	0			

7	6	5	4	3	2	1	0
BLANKBOTTOM[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31	-	-	Reserved	-
30	-	-	Reserved	-
29	-	-	Reserved	-
28	-	-	Reserved	-
[27:16]	BLANKTOP	R/W	Blanking From TOP 0	-
15	-	-	Reserved	-
14	-	-	Reserved	-
13	-	-	Reserved	-
12	-	-	Reserved	-
[11:0]	BLANKBOTT OM	R/W	Blanking From BOTTOM 0	-

27.5.3 TV0 Blanking (Horizontal) Register

P_TVE0_BLANKVDIR
0x93000008
TV0 Blanking (Horizontal)

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	BLANKTOP[27:24]			
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
BLANKTOP[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-	-		-	BLANKBOTTOM[11:8]			
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
BLANKBOTTOM[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
31	-	-	Reserved	-
30	-	-	Reserved	-
29	-	-	Reserved	-
28	-	-	Reserved	-

Bit	Function	Type	Description	Condition
[27:16]	BLANKTOP	R/W	Blanking From TOP 0	-
15	-	-	Reserved	-
14	-	-	Reserved	-
13	-	-	Reserved	-
12	-	-	Reserved	-
[11:0]	BLANKBOTTOM	R/W	Blanking From BOTTOM 0	-

27.5.4 TV0 Blanking Pattern Register

P_TVE0_BLANKDATA		0x9300000C								TV0 Blanking Pattern							
Bit	31	30	29	28	27	26	25	24	31	30	29	28	27	26	25	24	
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
BLANKDATAHI[23:16]																	
23	22	21	20	19	18	17	16	0	0	0	0	0	0	0	0	0	
BLANKDATAMI[16:8]																	
15	14	13	12	11	10	9	8	0	0	0	0	0	0	0	0	0	
BLANKDATALO[7:0]																	
7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:24]	-	-	Reserved	-
[23:17]	BLANKDATAHI	R/W	Blanking R/Crdata	-
[16:8]	BLANKDATAMI	R/W	Blanking G/Cbdata	-
[7:0]	BLANKDATALO	R/W	Blanking B/Ydata	-

27.5.5 TV0 Frame Buffer Address Register

P_TVE0_FRAMEADR								0x93000010								TV0 Frame Buffer Address								
Bit	31	30	29	28	27	26	25	24	FRAMEADR[31:24]								FRAMEADR[23:16]							
Function	0	0	0	0	0	0	0	0	FRAMEADR[23:16]								0	0	0	0	0	0	0	0
Default	0	0	0	0	0	0	0	0	FRAMEADR[15:8]								0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	FRAMEADR[15:8]								0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	FRAMEADR[7:0]								0	0	0	0	0	0	0	0	
7	6	5	4	3	2	1	0	FRAMEADR[7:0]								0	0	0	0	0	0	0	0	
[31:0]	FRAMEADR	R/W	Frame Buffer Address						-															
Bit	Function	Type	Description						Condition															

27.5.6 TV0 Frame Buffer Pitch Register

P_TVE0_FRAMEPITCH								0x93000014								TV0 Frame Buffer Pitch								
Bit	31	30	29	28	27	26	25	24	SRC_PITCH[31:24]								SRC_PITCH[23:16]							
Function	0	0	0	0	0	0	0	0	SRC_PITCH[23:16]								0	0	0	0	0	0	0	0
Default	0	0	0	0	0	0	0	0	ACT_PITCH[15:8]								0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	ACT_PITCH[15:8]								0	0	0	0	0	0	0	0	
15	14	13	12	11	10	9	8	-																
7	6	5	4	3	2	1	0	-																
[31:0]	FRAMEPITCH	R/W	Frame Buffer Pitch						-															

7	6	5	4	3	2	1	0
ACT_PITCH[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition				
[31:16]	SRC_PITCH	R/W	Source Pitch				-				
[15:0]	ACT_PITCH	R/W	Active Frame Video Pitch				-				
LIMIT:											
** PITCH = WIDTH*BYTE_PER_PIXEL											
** PITCH % 4 == 0											

27.5.7 TV0 Frame Buffer Resolution Register

P_TVE0_FRAMERESOLUTION 0x93000018 TV0 Frame Buffer Resolution							
Bit	31	30	29	28	27	26	25
Function	-	-	-	-	ACT_WIDTH[27:24]		
Default	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
ACT_WIDTH[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-	-	-	-	ACT_HEIGHT[11:8]			
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ACT_HEIGHT[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:28]	-	-	Reserved				-
[27:16]	ACT_WIDTH	R/W	Active Frame Video Width				-
[15:12]	-	-	Reserved				-
[11:0]	ACT_HEIGHT	R/W	Active Frame Video Height				-

27.5.8 TV0 Scaling Resolution Register

P_TVE0_SCALERESOLUTION 0x9300001C TV0 Scaling Resolution							
Bit	31	30	29	28	27	26	25
Function	-	-	-	-	SCL_WIDTH[27:24]		
Default	0	0	0	0	0	0	0
	23	22	21	20	19	18	17
	SCL_WIDTH[23:16]						
	0	0	0	0	0	0	0
	15	14	13	12	11	10	9
	-	-	-	-	SCL_HEIGHT[11:8]		
	0	0	0	0	0	0	0
	7	6	5	4	3	2	1
	SCL_HEIGHT[7:0]						
	0	0	0	0	0	0	0
Bit	Function	Type	Description			Condition	
[31:28]	-	-	Reserved			-	
[27:16]	SCL_WIDTH	R/W	Scaling Width			-	
[15:12]	-	-	Reserved			-	
[11:0]	SCL_HEIGHT	R/W	Scaling Height			-	

27.5.9 TV0 Scaling Parameter (Horizontal) Register

P_TVE0_SCALEHFACTOR 0x93000020 TV0 Scaling Parameter (Horizontal)							
Bit	31	30	29	28	27	26	25
Function	SCL_HINITVAL[31:24]						
Default	0	0	0	0	0	0	0
	23	22	21	20	19	18	17
	SCL_HINITVAL[23:16]						
	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
SCL_HFACTOR[15:8]							

0	0	0	0	0	0	0	0
SCL_HFACTOR[7:0]							

Bit	Function	Type	Description					Condition					
[31:16]	SCL_HINITVAL	R/W	Horizontal Scaling Initial Value					-					
[15:0]	SCL_HFACTOR	R/W	Horizontal Scaling Factor					-					
$\text{HScaleUpFactor} = \text{Floor}((65536 * \text{ACT_WIDTH}) / \text{SCL_WIDTH})$													
$\text{HScaleDownFactor} = \text{ceil} ((65536 * \text{SCL_WIDTH}) / \text{ACT_WIDTH})$													

27.5.10 TV0 Scaling Parameter (Vertical) Register

P_TVE0_SCALEVFACTOR 0x93000024 TV0 Scaling Parameter (Vertical)

Bit	31	30	29	28	27	26	25	24
SCL_VINITVAL0[31:24]								

Default 0 0 0 0 0 0 0 0

23	22	21	20	19	18	17	16
SCL_VINITVAL0[23:16]							

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
SCL_VFACTOR[15:8]							

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
SCL_VFACTOR[7:0]							

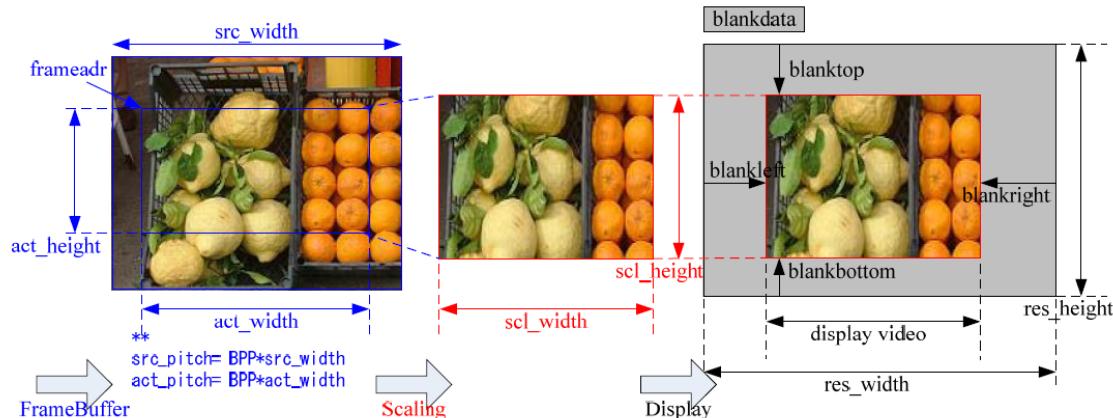
0 0 0 0 0 0 0 0



Bit	Function	Type	Description	Condition
[31:16]	SCL_VINITVAL0	R/W	Vertical Scaling Initial Value0	-
[15:0]	SCL_VFACTOR	R/W	Vertical Scaling Factor	-
** VScaleUpFactor = Floor((65536*ACT_HEIGHT)/SCL_HEIGHT)				
** VScaleDownFactor = ceil ((65536*SCL_HEIGHT)/ACT_HEIGHT)				
ScalingDown Limit:				
2*N*MAX_WIDTH*BUS_PERIOD < (HS_FRONT+HS_BACK)*DISP_PERIOD				
MAX_WIDTH = max(ACT_WIDTH,SCL_WIDTH)				
N = floor(ACT_HEIGHT/SCL_HEIGHT)				
For example1:				
512x512 -> 320x240, BUS_PERIOD= 5, DISP_PERIOD= 37, HS_FRONT = 24, HS_BACK = 128				
512/240 = 2.11 -> N = 2				
2*2*512*5 = 10240				
(24+128)*37 = 5624 (It cant scaling-down)				

27.5.11 TV0 Scaling Control Register

Bit	Function	Type	Description	Condition
[31:16]	SCL_VINITVAL1	-	Vertical Scaling Initial Value1	-
[15:4]	-	-	Reserved	-
[3]	SCL_VENABLE	R/W	Vertical Scaling Enable	
[2]	SCL_VSELECT	R/W	0: Vertical Scaling Up 1: Vertical Scaling Down	-
[1]	SCL_HENABLE	R/W	Horizontal Scaling Enable	-
[0]	SCL_HSELECT	R/W	0: Horizontal Scaling Up 1: Horizontal Scaling Down	-



27.5.12TV0 OSDx Base Address Register

P_TVE0_OSDNBASEADR 0x93000030 0x93000050 TV0 OSD N Base Address(N=0,1)

Bit	31	30	29	28	27	26	25	24
Function								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
BASEADR[23:16]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
BASEADR[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
BASEADR[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:0]	BASEADR	R/W	OSD Layer Base Address				-

27.5.13TV0 OSDx Pitch Register Register

P_TVE0_OSDNPITCH 0x93000034 0x93000054 TV0 OSD N Pitch (N=0,1)

Bit	31	30	29	28	27	26	25	24	
Function	-	-	-	-	OSD_SRCPITCH[27:24]				
Default	0	0	0	0	0	0	0	0	

23	22	21	20	19	18	17	16
OSD_SRCPITCH[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	
-	-	-	-	OSD_ACTPITCH[11:8]				
0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
OSD_ACTPITCH[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:28]	-	-	Reserved				-
[27:16]	OSD_SRCPITCH	R/W	Source OSD Layer Pitch				
[15:12]	-	-	Reserved				
[11:0]	OSD_ACTPITCH	R/W	Active OSD Layer Pitch				

27.5.14TV0 OSDx Resolution Register

P_TVE0_OSDNRESOLUTION 0x93000038 0x93000058 TV0 OSD N OSDx Resolution
(N=0,1)

Bit	31	30	29	28	27	26	25	24	
Function	-	-	-	-	OSD_ACTWIDTH[27:24]				
Default	0	0	0	0	0	0	0	0	

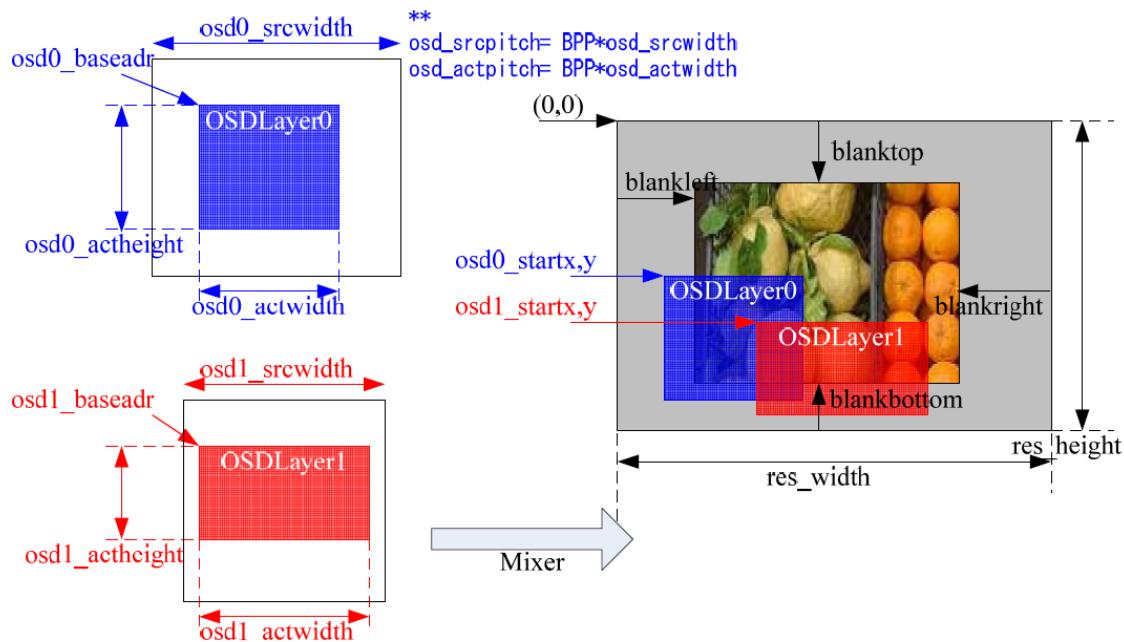


Bit	Function	Type	Description	Condition
[31:28]	-	-	Reserved	-
[27:16]	OSD_ACTWIDTH	R/W	Active OSD Layer Width	
[15:12]	-	-	Reserved	
[11:0]	OSD_ACTHEIGHT	R/W	Active OSD Layer Height	

27.5.15TV0 OSDx Starting Coordinate Register

P_TVE0_ 0x9300003C 0x9300005C **TV0 OSD N Starting Coordinate (N=0,1)**

Bit	Function	Type	Description	Condition
[31:28]	-	-	Reserved	-
[27:16]	OSD_STARTY	R/W	Starting Coordinate Y	
[15:12]	-	-	Reserved	
[11:0]	OSD_STARTX	R/W	Starting Coordinate X	



27.5.16 TV0 OSDx Format Register

P_TVE0_OSDNFORMAT 0x93000040 0x93000048 TV0 OSD N Format (N=0,1)							
Bit	31	30	29	28	27	26	25
Function	OSD_FORMAT	-	-	OSD_CONSTA	-	OSD_PPAMD	-
Default	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
OSD_ALPHA[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
OSD_CKEY[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
OSD_CKEY[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:30]	OSD_FORMAT	R/W	0: RGB565 1: RGB5515 2: RGB1555	-
[29:28]	-	-	Reserved	
[27]	OSD_CONSTA	R/W	0: Per-pixel-Alpha 1: Constant Alpha	
[26]	-	-	Reserved	
[25:24]	OSD_PPAMD	R/W	0: PerPixelAlpha Only 1: ColorKey Only 2: PerPixelAlpha+ColorKey	
[23:16]	OSD_ALPHA	R/W	Alpha Channel (0~64)	
[15:0]	OSD_CKEY	R/W	Transparent Color Key	

27.5.17 TV0 OSDx Control Register

P_TVE0_OSDNCONTROL 0x93000044 0x9300004C TV0 OSD N Control (N=0,1)

Bit	31	30	29	28	27	26	25	24
Function	OSD_FORMAT		-		OSD_CONSTA	-	OSD_PPAMD	
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
OSD_ALPHA[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
OSD_CKEY[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
OSD_CKEY[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:30]	OSD_FORMAT	R/W	0: RGB565 1: RGB5515 2: RGB1555	-
[29:28]	-	-	Reserved	
[27]	OSD_CONSTA	R/W	0: Per-pixel-Alpha 1: Constant Alpha	
[26]	-	-	Reserved	
[25:24]	OSD_PPAMD	R/W	0: PerPixelAlpha Only 1: ColorKey Only 2: PerPixelAlpha+ColorKey	
[23:16]	OSD_ALPHA	R/W	Alpha Channel (0~64)	
[15:0]	OSD_CKEY	R/W	Transparent Color Key	

27.5.18 TV0 OSDx Scaling Parameter (Horizontal) Register

P_TVE0_OSDNHFATOR 0x93000060 0x93000070									TV0 OSD N Scaling Parameter (Horizontal) (N=0,1)								
Bit	31	30	29	28	27	26	25	24	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0
Function	0	0	0	0	0	0	0	0	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0
Default	0	0	0	0	0	0	0	0	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0
	23	22	21	20	19	18	17	16	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0
	0	0	0	0	0	0	0	0	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0
	15	14	13	12	11	10	9	8	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0
	0	0	0	0	0	0	0	0	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0
	7	6	5	4	3	2	1	0	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0
	0	0	0	0	0	0	0	0	OSDHINITVAL[31:24]	OSDHINITVAL[23:16]	OSDHINITVAL[15:8]	OSDHINITVAL[7:0]	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:16]	OSDHINITVAL	R/W	Horizontal Scaling Initial Value				-
[15:0]	OSDHFACTOR	R/W	Horizontal Scaling Factor				-
$\text{** HscaleFactor} = \text{Floor}(2048 * (\text{OSD_ACTWIDTH}-1) / (\text{OSD_SCLWIDTH}-1))$							

27.5.19 TV0 OSDx Scaling Parameter (Vertical) Register

TV0 OSD N Scaling Parameter (Vertical) (N=0,1)								
Bit	31	30	29	28	27	26	25	24
Function	OSDVINITVAL0[31:24]							
Default	0	0	0	0	0	0	0	0
OSDVINITVAL0[23:16]								
	0	0	0	0	0	0	0	0
OSDVFACTOR[15:8]								
	0	0	0	0	0	0	0	0
OSDVFACTOR[7:0]								
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:16]	OSDVINITVAL0	R/W	Vertical Scaling Initial Value0				-
[15:0]	OSDVFACTOR	R/W	Vertical` Scaling Factor				-
$\text{** VscaleFactor} = \text{Floor}(2048 * (\text{OSD_ACTHEIGHT}-1) / (\text{OSD_SCLHEIGHT}-1))$							

27.5.20 TV0 OSDx Scaling Parameter (field 1) Register

TV0 OSD N Scaling Parameter (field 1) (N=0,1)								
Bit	31	30	29	28	27	26	25	24
Function	OSDVINITVAL1[31:24]							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
OSDVINITVAL1[23:16]							

0	0	0	0	0	0	0	0
-							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	OSDVINITVAL1	R/W	Vertical Scaling Initial Value1					-
[15:0]	-	-	Reserved					-

27.5.21 TV0 OSDx Scaling Resolution Register

P_TVE0_OSDNSCLWH 0x9300006C 0x9300007C TV0 OSD N Scaling Resolution
(N=0,1)

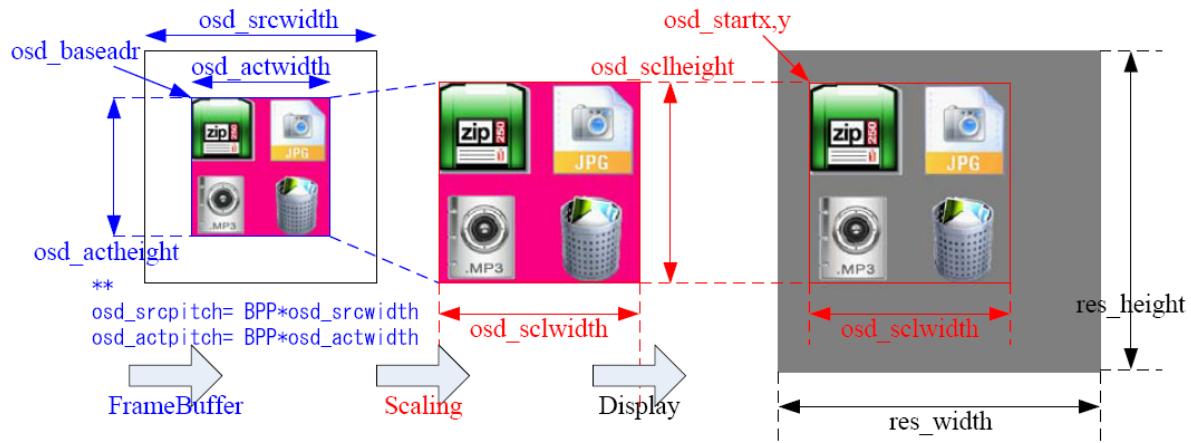
Bit	31	30	29	28	27	26	25	24
Function								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
OSD_SCLWIDTH[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
OSD_SCLHEIGHT[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
OSD_SCLHEIGHT[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	OSD_SCLWIDTH	R/W	OSD Layer Scaling Width	-
[15:0]	OSD_SCLHEIGHT	R/W	OSD Layer Scaling Height	-



27.5.22 TV0 Interrupt Enable Register

P_TVE0_INTRQENABLE								0x93000080								TV0 Interrupt Enable								
Bit	31	30	29	28	27	26	25	24	INTENA[31:24]								INTENA[23:16]							
Function	INTENA[31:24]								INTENA[23:16]								INTENA[15:8]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:0]	INTENA	R/W	Interrupt Enable				-

27.5.23 TV0 Interrupt Status Register

P_TVE0_INTRQENABLE

0x93000080

TV0 Interrupt Enable

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	-	OSD_BANK	-	-	GAMMA_BANK	-	-

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	UPDATE_FAIL	FIELD_END	FRAME_END	UPDATE_PARAM	DISPLAY_OFF	-	-

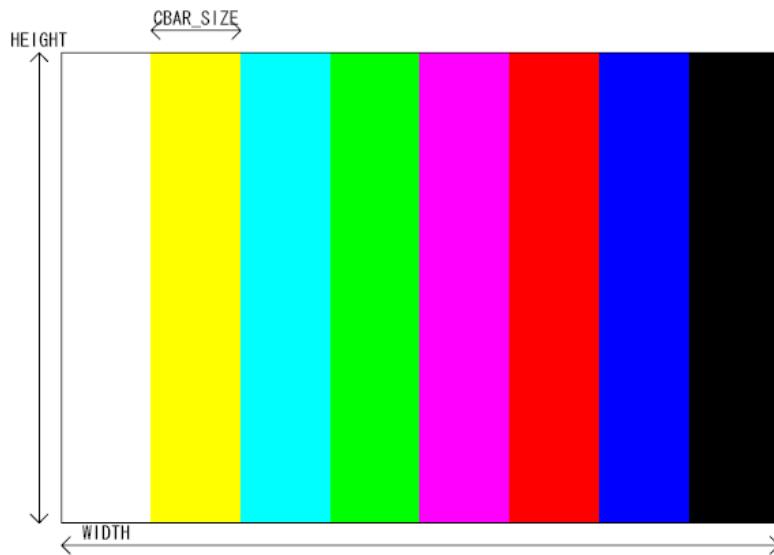
0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description				Condition
[20]	OSD_BANK	R/W	0: PLEPOINTER on OSD-0 1: PLEPOINTER on OSD-1				-
[17:16]	GAMMA_BANK	R/W	0: Gamma Table1 Select(R) 1: Gamma Table1 Select(G) 2: Gamma Table1 Select(B)				-
[4]	UPDATE_FAIL	R/W/C	MMRs don't update				-
[3]	FIELD_END	R/W/C	Each field end				-
[2]	FRAME_END	R/W/C	Each frame end				-
[1]	UPDATE_PARAM	R/W/C	Updates all of the settings				-
[0]	DISPLAY_OFF	R/W/C	Controller is disabled				-

27.5.24 TV0 Colorbar Control Register

P_TVE0_CBARCONTROL		0x93000090								TV0 Colorbar Control							
Bit	31	30	29	28	27	26	25	24									
Function	CBAR_GEN				-												
Default	0	0	0	0	0	0	0	0									
	23	22	21	20	19	18	17	16									
					-												
	0	0	0	0	0	0	0	0									
	15	14	13	12	11	10	9	8									
			-		CBAR_TYPE[11:8]												
	0	0	0	0	0	0	0	0									
	7	6	5	4	3	2	1	0									
					CBAR_SIZE[7:0]												
	0	0	0	0	0	0	0	0									

Bit	Function	Type	Description	Condition
[31]	CBAR_GEN	R/W	0: Disable 1: Enable	-
[30:12]	-	-	Reserved	-
[11:8]	CBAR_TYPE	R/W	0: Single color 1: 75/0/75/0 (gamma=1/0.45) 2: 75/0/75/0 (RGB) 3: 75/0/75/0 (YCbCr) 4: 100/0/75/0 (gamma=1/0.45) 5: 100/0/75/0 (RGB) 6: 100/0/75/0 (YCbCr) 7: 100/0/100/0 (RGB) 8: 100/0/100/0 (YCbCr) 9: 10step staircase (RGB) A: 10step staircase (YCbCr)	-
[7:0]	CBAR_SIZE	R/W	Column per bar	-



27.5.25TV0 Colorbar Color Register

P_TVE0_CBARCOLOR								0x93000094								TV0 Colorbar Color															
Bit	31	30	29	28	27	26	25	24																							
Function	-																														
Default	0 0 0 0 0 0 0 0																														
23 22 21 20 19 18 17 16								COLORHI[23:16]																							
0 0 0 0 0 0 0 0																															
15 14 13 12 11 10 9 8								COLORMI[15:8]																							
0 0 0 0 0 0 0 0																															
7 6 5 4 3 2 1 0								COLORLO[7:0]																							
0 0 0 0 0 0 0 0																															

Bit	Function	Type	Description	Condition
[31:0]	-	-	Reserved	-
[23:16]	COLORHI	R/W	Single color R/Crdata	-
[15:8]	COLORMI	R/W	Single color G/Cbdata	-
[7:0]	COLORLO	R/W	Single color B/Ydata	-

27.5.26TV0 Color Matrix Param0 Register

P_TVE0_COLORMATRIX0 0x930000A0 TV0 Color Matrix Param0							
Bit	31	30	29	28	27	26	25
Function	CMTX_a01[31:24]						
Default	0	0	0	0	0	0	0
CMTX_a01[23:16]							
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
CMTX_a00[15:8]							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
CMTX_a00[7:0]							
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
Bit	Function	Type	Description				Condition
[31:16]	CMTX_a01	R/W	Color Matrix Parameter				-
[15:0]	CMTX_a00	R/W	Color Matrix Parameter				-

27.5.27TV0 Color Matrix Param1 Register

P_TVE0_COLORMATRIX1 0x930000A4 TV0 Color Matrix Param1							
Bit	31	30	29	28	27	26	25
Function	CMTX_a10[31:24]						
Default	0	0	0	0	0	0	0
CMTX_a10[23:16]							
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
CMTX_a02[15:8]							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
CMTX_a02[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:16]	CMTX_a10	R/W	Color Matrix Parameter				-
[15:0]	CMTX_a02	R/W	Color Matrix Parameter				-

27.5.28TV0 Color Matrix Param2 Register

P_TVE0_COLORMATRIX2
0x930000A8
TV0 Color Matrix Param2

Bit	31	30	29	28	27	26	25	24
Function CMTX_a12[31:24]								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
CMTX_a12[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
CMTX_a11[15:8]							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
CMTX_a11[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:16]	CMTX_a12	R/W	Color Matrix Parameter				-
[15:0]	CMTX_a11	R/W	Color Matrix Parameter				-

27.5.29TV0 Color Matrix Param3 Register

P_TVE0_COLORMATRIX3
0x930000AC
TV0 Color Matrix Param3

Bit	31	30	29	28	27	26	25	24
Function CMTX_a21[31:24]								
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
CMTX_a21[23:16]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
CMTX_a20[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
CMTX_a20[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:16]	CMTX_a21	R/W	Color Matrix Parameter					-
[15:0]	CMTX_a20	R/W	Color Matrix Parameter					-

27.5.30 TV0 Color Matrix Param4 Register

P_TVE0_COLORMATRIX4
0x930000B0
TV0 Color Matrix Param4

Bit	31	30	29	28	27	26	25	24
CMTX_b0[31:24]								
Function	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
CMTX_b0[23:16]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

15	14	13	12	11	10	9	8
CMTX_a22[15:8]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
CMTX_a22[7:0]							

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Bit	Function	Type	Description					Condition
[31:16]	CMTX_b0	R/W	Color Matrix Parameter					-
[15:0]	CMTX_a22	R/W	Color Matrix Parameter					-

27.5.31 TV0 Color Matrix Param5 Register

P_TVE0_COLORMATRIX5								0x930000B0								TV0 Color Matrix Param5								
Bit	31	30	29	28	27	26	25	24	CMTX_b2[31:24]															
Function									CMTX_b2[31:24]															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16	CMTX_b2[23:16]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	CMTX_b1[15:8]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0	CMTX_b1[7:0]															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	Function	Type	Description						Condition															
[31:16]	CMTX_b2	R/W	Color Matrix Parameter						-															
[15:0]	CMTX_b1	R/W	Color Matrix Parameter						-															

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

27.5.32 TV0 Amplitude Adjust Register

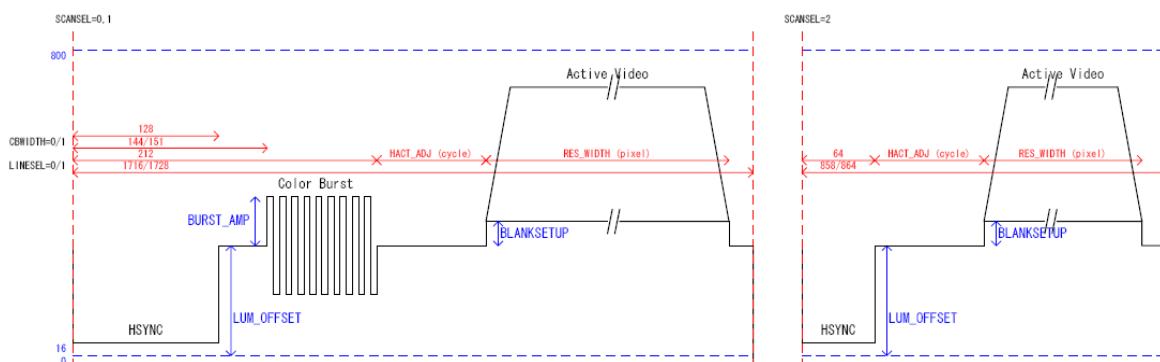
P_TVE0_TVEAMPADJ								0x930000D8								TV0 Amplitude Adjust								
Bit	31	30	29	28	27	26	25	24	LUM_OFFSET[29:24]															
Function	-								LUM_OFFSET[29:24]															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
LUM_OFFSET[23:20]				BLANKSETUP[19:16]			
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
BLANKSETUP[15:10]				BURST_AMP[9:8]			
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
BURST_AMP[7:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:30]	-	-	Reserved	-
[29:20]	LUM_OFFSET	R/W	Luminance Offset	-
[19:10]	BLANKSETUP	R/W	Blanking Pedestal	-
[9:0]	BURST_AMP	R/W	Color Burst Amplitude	-



27.5.33 TV0 Position Adjust Register

P_TVE0_TVEPOSADJ
0x930000DC
TV0 Position Adjust

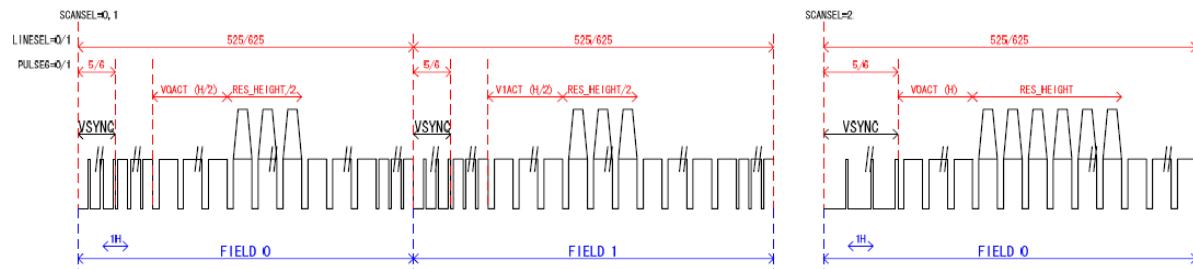
Bit	31	30	29	28	27	26	25	24
Function	-							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
V1ACT_ADJ[23:16]							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
V0ACT_ADJ[15:8]							

0	0	0	0	0	0	0	0
HACT_ADJ[7:0]							

Bit	Function	Type	Description	Condition
[31:24]	-	-	Reserved	-
[23:16]	V1ACT_ADJ	R/W	Vertical Adjust at Field 1	
[15:8]	V0ACT_ADJ	R/W	Vertical Adjust at Field 0	
[7:0]	HACT_ADJ	R/W	Horizontal Adjust	-

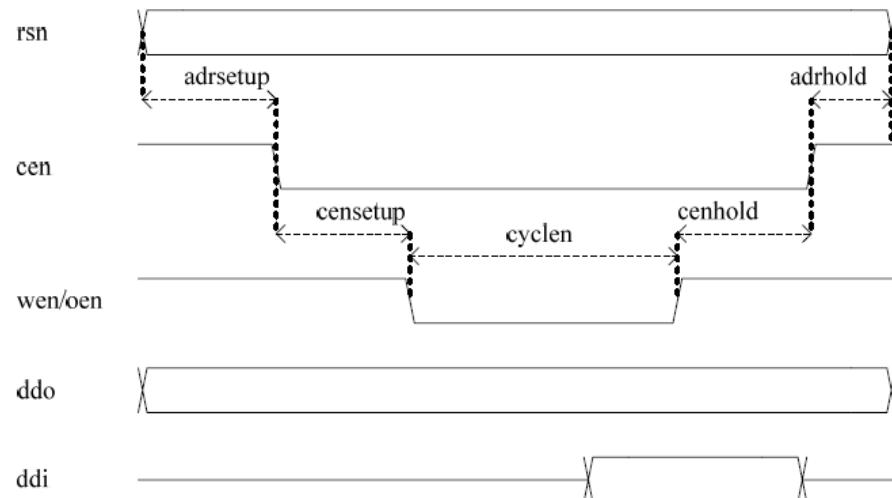


27.5.34 TV0 LCM AC Timing Register

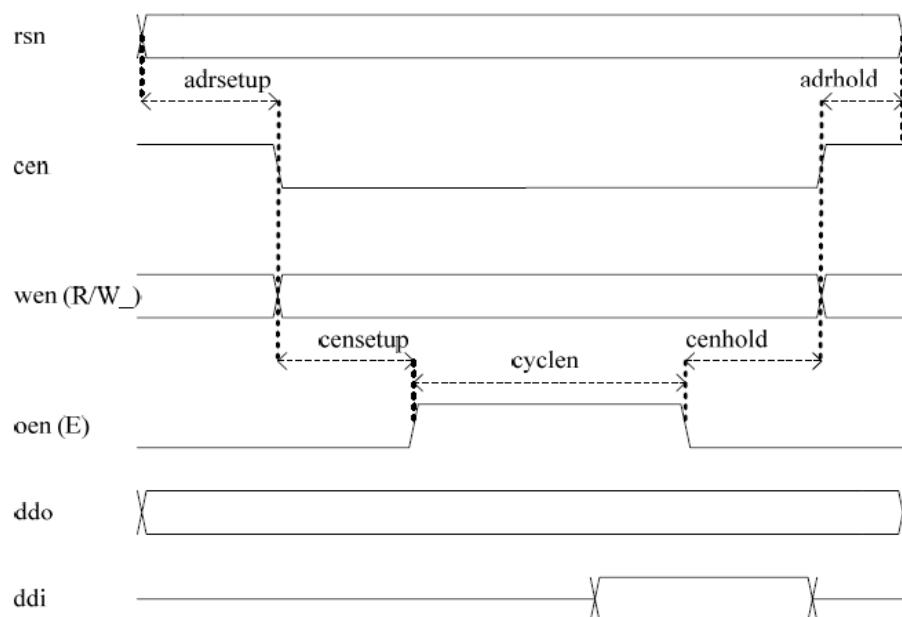
P_TVE0_LCMACTIME								0x930000E0								TV0 LCM AC Timing								
Bit	31	30	29	28	27	26	25	24	ADRSETUP[31:28]								ADRHOOLD[27:24]							
Function	Default								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<hr/>																								
23	22	21	20	19	18	17	16	ADRHOOLD[23:16]																
0	0	0	0	0	0	0	0																	
15	14	13	12	11	10	9	8	CENSETUP[15:12]								CENHOLD[11:8]								
0	0	0	0	0	0	0	0																	
7	6	5	4	3	2	1	0	CYCLEN [7:0]																
0	0	0	0	0	0	0	0																	

Bit	Function	Type	Description	Condition
[31:28]	ADRSETUP	R/W	Address Setup Time	-
[27:16]	ADRHOLD	R/W	Address Hold Time	-
[15:12]	CENSETUP	R/W	Chip Select Setup Time	-
[11:8]	CENHOLD	R/W	Chip Select Hold Time	-
[7:0]	CYCLEN	R/W	Strobe Length	-

MCU-8080 :



MCU-6800 :



(0x93000F00)LCM ProgramIO Mode (16bit)

TV LCM Programming In out

haddr				
Bit	Name	Type	Description	
11:8	RegMapOffset	RW	LCM ProgramIO Mode	1
7		RW		0
6		RW		0
5	RegisterSelect1	RW	RegisterSelect for Data1	X
4	RegisterSelect0	RW	RegisterSelect for Data0	X
3:0		RW	0	0
hwdata/hrdata				
Bit	Name	Type	Description	Reset Value
31:16	Data1	RW	Data channel1	0
15:0	Data0	RW	Data channel0	0

(0x93000F00)LCM ProgramIO Mode (8bit)

haddr				
Bit	Name	Type	Description	
11:8	RegMapOffset	RW	LCM ProgramIO Mode	1
7	RegisterSelect3	RW	RegisterSelect for Data3	X
6	RegisterSelect2	RW	RegisterSelect for Data2	X
5	RegisterSelect1	RW	RegisterSelect for Data1	X
4	RegisterSelect0	RW	RegisterSelect for Data0	X
3:0		RW	0	0
hwdata/hrdata				
Bit	Name	Type	Description	Reset Value
31:24	Data3	RW	Data channel3	0
23:16	Data2	RW	Data channel2	0
15:8	Data1	RW	Data channel1	0
7:0	Data0	RW	Data channel0	0

27.5.35TV0 Display Format Register

P_TVE0_DISPFORMAT								0x930000F0								TV0 Display Format										
Bit	31	30	29	28	27	26	25	24																		
Function																										
Default	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
	23	22	21	20	19	18	17	16																		
	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	15	14	13	12	11	10	9	8													CFTTYPE					
	ODITHER	-	ODITHSEL[13:12]						-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

7	6	5	4	3	2	1	0
CUPTYPE[7:6]	YUPTYPE[5:4]	CFUNC	CFORMAT	CTYPE[1:0]			
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	-	Reserved	-
[15]	ODITHER	R/W	Dithering Enable	
[14]	-	-	Reserved	
[13:12]	ODITHSEL	R/W	0: Ordered Dithering (Fixed) 1: Ordered Dithering (Wheel) 2: HorizontalErrorDiffusion	
[11:9]	-	-	Reserved	
[8]	CFTTYPE	R/W	0: None 1: [3 5 8 10 12 10 8 5 3]/64	
[7:6]	CUPTYPE	R/W	0: None 1: [1 1]/2 2: [1 2 1]/4 3: [-1 0 5 8 5 0 -1]/16	
[5:4]	YUPTYPE	R/W	Same as CUPTYPE	-
[3]	CFUNC	R/W	0: 0-255 RGB <-> 0-255 YCbCr 1: 0-255 RGB <->16-235 YCbCr	-
[2]	CFORMAT	R/W	Input Format: 0: RGB Color Space 1: YCbCr Color Space	-
[1:0]	CTYPE	R/W	Input type: RBG YCbCr/YUV 0: RGB565 YCbYCr 1: RGB555 4Y4Cb4Y4Cr 2: RGB888 YCbCr	-

27.5.36 TV0 Display Control Register

P_TVE0_DISPFORMAT		0x930000FC		TV0 Display Control					
Bit	31	30	29	28	27	26	25	24	
Function	DISPLAY[31:30]	-	DTYPE	-	-	-	-	-	BTYPE[25:24]
Default	0	0	0	0	0	0	0	0	

23	22	21	20	19	18	17	16
OCLKPOLAR	-	OCLKSEL	-	OBT656	-	OBLANK0	
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
-	LCM8B	LCM68	LCMDH	TVTYPE	PULSE6	SCANSEL[9:8]	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
FSCTYPE[7:5]			FIX625	LINESEL	CBWIDTH	CBSEL[1:0]	
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:30]	DISPLAY	R/W	0: Disable 1: LCM DMA 2: LCD Enable 3: TV Enable	-
[29]	-	-	Reserved	
[28]	DTYPE	R/W	0: Progressive DMA 1; Interlaced DMA	
[27:26]	-	-	Reserved	
[25:24]	BTYPEn	R/W	0: SINGLE 1: INCR4 2: INCR8 3: INCR16	
[23]	OCLKPOLAR	R/W	LCD Sample Clock Polarity	
[22]	-	-	Reserved	
[21:20]	OCLKSEL	R/W	lcd_dclk = clk/CLKDIV CLKDIV 0: 1 1: 2 2: 4 3: 8	
[19]	-	-	Reserved	
[18]	OBT656	R/W	Insert SAV,EAV into video stream	
[17]	-	-	Reserved	
[16]	OBLANK0	R/W	Fill 0 during blanking interval.	

Bit	Function	Type	Description	Condition
[15]	-	-	Reserved	
[14]	LCM8B	R/W	0: LCM 16bit Interface 1: LCM 8bit Interface	
[13]	LCM68	R/W	0: MCU-8080 Mode 1: MCU-6800 Mode	
[12]	LCMDH	R/W	0: Data Select is logic-0 1: Data Select is logic-1	
[11]	TVTYPE	R/W	0: NTSC 1: PAL	
[10]	PULSE6	R/W	0: 5 Serration Pulse 1: 6 Serration Pulse	
[9:8]	SCANSEL	R/W	0: Non-interlaced 1: Interlaced 2: Progressive	
[7:5]	FSCTYPE	R/W	0: 3.579Mhz @NTSCMJ 1: 4.433Mhz @PALBDGHIN 2: 4.433Mhz @NTSC443, 3: 3.575Mhz @PALM 4: 3.582Mhz @PALNc	
[4]	FIX625	R/W	Add the 1/625 Fh factor to Fsc	-
[3]	LINESEL	R/W	0: 262 LINE or 525 LINE 1: 312 LINE or 625 LINE	-
[2]	CBWIDTH	R/W	0: Color burst width = 2.52us 1: Color burst width = 2.25us	-
[1:0]	CBSEL	R/W	0: Color Burst @NTSCMJ 1: Color Burst @PALM 2: Color Burst @PALBDGHINNc 3: Disable Color Burst	-

27.5.37 TV0 OSDx Palette Pattern Register File

P_TVE0_PALETTEPTR
0x93000800
TV0 OSDx Palette Pattern

Bit	31	30	29	28	27	26	25	24
Function					-			
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-							

0 0 0 0 0 0 0 0

15	14	13	12	11	10	9	8
PALETTE _n [15:8]							

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
PALETTE _n [7:0]							

0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[31:16]	-	-	Reserved	-
[15:0]	PALETTE _n	W	OSD Palette Pattern	-

NOTE:

ADDRESS(PALETTE_n) = PALETTEPTR + n * 4

For example, the address of PALETTE3 is 0x80c

27.6 Recommended Settings

Generalplus highly recommends users take the following table as a quick and optimal reference for TV encoder applications to facilitate the configuration of NTSC and PAL - the two most popular systems used in today's TV industry.

ProGramming Example:

TV Parameter at 480i/576i System

	NTSCM	NTSCJ	NTSC443	PALM	PAL60	PALBDGHI	PALN	PALNc
TVTYPE	0	0	0	1	1	1	1	1
PULSE6	1	1	1	1	1	0	1	0
SCANSEL	1	1	1	1	1	1	1	1
FSCTYPE	0	0	2	3	2	1	1	4
FIX625	0	0	0	0	0	1	1	1
LINESEL	0	0	0	0	0	1	1	1
CBWIDTH	0	0	1	0	1	1	1	0
CBSEL	0	0	0	1	0	2	2	2
BURST_AMP	112	112	112	117	117	117	117	117
BLANKSETUP	42	0	42	42	0	0	42	0
LUM_OFFSET	240	240	240	240	252	252	240	252
HACT_ADJ	32	32	32	32	32	52	52	52
V0ACT_ADJ	26	26	26	26	26	32	32	32
V1ACT_ADJ	26	26	26	26	26	32	32	32
RES_WIDTH	720	720	720	720	720	720	720	720
RES_HEIGHT	480	480	480	480	480	576	576	576



TV Parameter at non-interlaced NTSC/PAL System

	NTSCM	PALM	PALBDGHINNc
TVTYPE	0	1	1
PULSE6	1	1	0
SCANSEL	0	0	0
FSCTYPE	0	3	1
FIX625	0	0	0
LINESEL	0	0	1
CBWIDTH	0	0	1
CBSEL	0	1	2
BURST_AMP	112	117	117
BLANKSETUP	42	42	0
LUM_OFFSET	240	240	252
HACT_ADJ	32	32	52
V0ACT_ADJ	26	26	32
RES_WIDTH	720	720	720
RES_HEIGHT	240	240	288

TV Parameter at 480p/576p System

	480P	576P
PULSE6	1	0
SCANSEL	2	2

LINESEL	0	1
CBSEL	3	3
HACT ADJ	58	68
V0ACT ADJ	32	39
RES WIDTH	720	720
RES HEIGHT	480	576

Input Formats

28 TV Encoder Interface 1

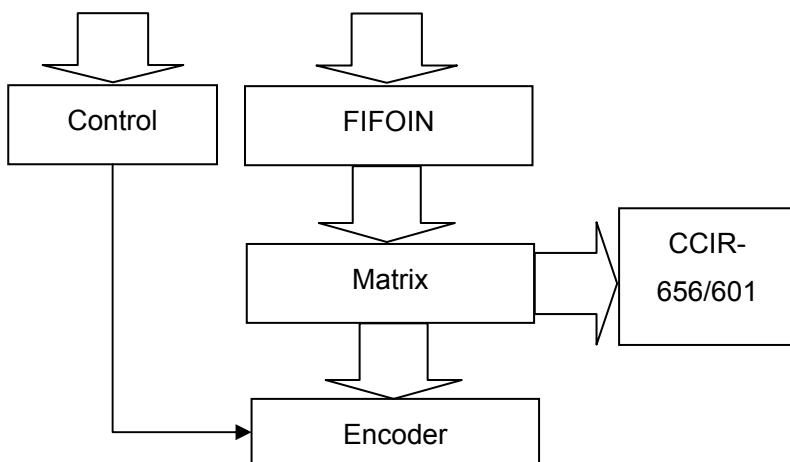
28.1 Introduction

This chapter mainly is intended to describe the TV encoder interface and its major features including the control, saturation, hue, brightness, luminance gain, luminance delay, edge enhancement, horizontal position, and vertical position.

28.2 Features

- Multi-Standard Support : NTSC-(M, N, J)/PAL-(B, D, G, H, I, M, N, Nc)/user-defined
- Composite CVBS output
- 27MHz clock input only
- Interlaced/non-interlaced operation
- Built-in CCIR-601/656 output
- Built-in color bar for color adjustment
- 10-bit DAC
- Saturation/Hue/Brightness/Contrast adjustable
- Edge enhancement
- Horizontal/Vertical position adjustment
- Luminance delay adjustment
- Y/Pb/Pr output

28.3 Block Diagram



28.4 Register Summary

NTSC-M,N,J / PAL-B,D,G,H,I,M,N,Nc

* For TVSTD = 0 ~ 6

Name	Address	Register [15:8]	Register [7:0]
P_TV1_CTRL	0x930200F0	0x00	Control
P_TV1_SATURATION	0x93020200	0x00	Saturation
P_TV1_HUE	0x93020204	0x00	Hue
P_TV1_BRIGHTNESS	0x93020208	0x00	Brightness
P_TV1_SHARPNESS	0x9302020C	0x00	Sharpness
P_TV1_Y_GAIN	0x93020210	0x00	YGain
P_TV1_Y_DELAY	0x93020214	0x00	YDelay
P_TV1_V_POSITION	0x93020218	0x00	VPOS
P_TV1_H_POSITION	0x9302021C	0x00	HPOS
P_TV1_VIDEODAC	0x93020220	0x00	CCIR Output control VDAC control

User-Defined TV standard

* For TVSTD = 7

Name	Address	Register [15:8]	Register [7:0]
P_TV1_CTRL	0x930200F0	0x00	Control 1
P_TV1_CTRL2	0x930200F4	0x00	Control 2
P_TV1_SATURATION	0x93020200	ACC	Saturation
P_TV1_HUE	0x93020204	UV_RATIO	Hue
P_TV1_BRIGHTNESS	0x93020208	BLK_LVL	Brightness
P_TV1_SHARPNESS	0x9302020C	BGP_SOFF	BGP_EOFF
P_TV1_Y_GAIN	0x93020210	AGC YGain	Sharpness
P_TV1_Y_DELAY	0x93020214	SC_FREQ0	H625
P_TV1_V_POSITION	0x93020218	SC_FREQ1	VP OS PAL SC_RST PED YDelay
P_TV1_H_POSITION	0x9302021C	SC_FREQ2	HPOS
P_TV1_VIDEODAC	0x93020220	SC_FREQ3	CCIR Output Control

28.5 Register Definition

28.5.1 Control Register

P_TV1_CTRL 0x930200F0 TV Control Register							
Bit	15	14	13	12	11	10	9
Function	-			HDSEL	VDSEL	EVEN	Sync
Default	0	0	0	0	0	0	0
	7	6	5	4	3	2	1
VINFMT0	Resolution		NonIntl	TVSTD			TVEN
	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:13]	-	-	Reserved	-
12	HDSEL	R/W	Output HD Selection	0: CSYNC 1: HD
11	VDSEL	R/W	Output VD Width Selection (for CCIR-601 DAC)	0: 3.0 or 2.5 H 1: 1.0 Hsync
			BGP_Blank (for video DAC)	0: Standard 1: VD blanking only
10	EVEN	R/W	Even Field Advance Phase	0: odd field advanced phase 1: even field advanced phase
9	Sync	R/W	Synchronize with 1 st Vsync Falling	0: Not Sync with others 1: Sync with other's 1 st Vsync falling
[8:7]	VINFMT	R/W	Video Input Data Format Selection	0: RGB888 1 ~ 3: Reserved
[6:5]	Resolution (*1)	R/W	Display Screen Resolution Selection	0: QVGA (320x240) 1: HVGA (640x240) 2: D1(720X480) 3: Internal Color Bar
4	NonIntl(*3)	R/W	Non-Interlaced Mode Selection	0: Interlaced 1: Non-interlaced
[3:1]	TVSTD	R/W	TV Standard Selection For common use: When TVSTD[1:0] = 00, 1. TVSTD[2] = 0 means NTSC. 2. TVSTD[2] = 1 means PAL.	000 (0): NTSC-M 001 (1): NTSC-J 010 (2): NTSC-N 011 (3): PAL-M 100 (4): PAL-B,D,G,H,I 101 (5): PAL-N 110 (6): PAL-Nc 111 (7): Reserved

Bit	Function	Type	Description	Condition
0	TVEN(*2)	R/W	TV Module Enable It won't control video DAC.	0: TV signal disabled 1: TV signal enabled

Note1: For QVGA output, users must select "QVGA" (Resolution = 0) and "Non-Interlaced" (NonIntl = 1).

For VGA interlaced output, users must select "HVGA" (Resolution = 1) and "Interlaced" (NonIntl = 0).

For VGA non-interlaced output, users must select "HVGA" (Resolution = 1) and "Non-Interlaced" (NonIntl = 1).

Note2: When TVSTD is changed, TVEN must be disabled first.

Note3: For 320x240 and 640x240, enabling NonIntl will increase refreshment rate and reduce flicker.

P_TV1_CTRL2

0x930200F4

TV Control Register

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	-	-	-	-	-	Y/Pb/Pr	576mode	LB_DIS
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]	-	-	Reserved	-
4	PROGEN	R/W	480p or 576p mode enable register, this register must be used with COMPEN = 1 or CCIR's output.	0 : 480i or 576i mode. 1 : 480p or 576p mode.
3	CLKINV	R/W	CCIR clock output invert select register.	0 : Non-invert the CCIR's clock output. 1 : Invert the CCIR's clock output.
2	Y/Pb/Pr	R/W	This bit is for Y/Pb/Pr output	1 = Enable 0 = Disable
1	576mode	R/W	This bit is for 720x576 mode	1 = Enable 0 = Disable
0	LB_DIS	R/W	Horizontal position left boundary disable This bit is valid only when TV resolution is set as D1(720x480). When this bit is '0', the value which is small than '7' filled in P_TV_H_POSITION is invalid. When this bit is '1', the value which is small than '7' filled in P_TV_H_POSITION is valid.	1 = Disable the left boundary 0 = Enable the left boundary

28.5.2 Saturation Register

P_TV1_SATURATION								0x93020200								TV Saturation Register								
Bit	15	14	13	12	11	10	9	8																
Function	-																							
Default	-	-	-	-	-	-	-	-																
7	6	5	4	3	2	1	0	Saturation								1	0	0	0	0	0	0	0	0

Bit	Function	Type	Description								Condition
[15:8]	-	-	Reserved								-
[7:0]	Saturation	R/W	Saturation Adjustment								0: Minimum 128: Default 255: Maximum

28.5.3 Hue Register

P_TV1_HUE								0x93020204								TV Hue Register								
Bit	15	14	13	12	11	10	9	8																
Function	-																							
Default	-	-	-	-	-	-	-	-																
7	6	5	4	3	2	1	0	Hue								0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description								Condition
[15:8]	-	-	Reserved								-
[7:0]	Hue	R/W	Hue Adjustment								0: 0 degree 1: 1/256 degree ... 255: 255/256 degree

28.5.4 Brightness Register

P_TV1_BRIGHTNESS								0x93020208								TV Brightness Register								
Bit	15	14	13	12	11	10	9	8																
Function	-																							
Default	-	-	-	-	-	-	-	-																

7	6	5	4	3	2	1	0
Brightness							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:0]	Brightness	R/W	Brightness adjustment	127: Bright (0x7F) 0: Normal (0x00) -128: Dark (0x80)

28.5.5 Sharpness Register

P_TV1_SHARPNESS		0x9302020C								TV Sharpness Register							
Bit	15	14	13	12	11	10	9	8									
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
Cbar		YLPF			SharpnessSel		Sharpness
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:6]	Cbar	R/W	Color Bar Option	0: 75 color bar 1: invalid 2: 100/75 color bar 3: 100 color bar
[5:4]	YLPF	R/W	Luminance Low Pass Filter Selection	0: LPF OFF 1: LPF ON 2~3: Reserved
[3:2]	SharpnessSel	R/W	Sharpness Time Constant Selection	0: 74ns 1: 148ns 2~3: Reserved
[1:0]	Sharpness	R/W	Sharpness Adjustment	0: 0 (NO sharpness) 1: peak gain 0.5 2: peak gain 1.0 3: peak gain 2.0

28.5.6 Y Gain Register

P_TV1_Y_GAIN		0x93020210								TV Y Gain Register							
Bit	Function	15	14	13	12	11	10	9	8								
Function		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		7	6	5	4	3	2	1	0								
		YGain															
		1	0	0	0	0	0	0	0								

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:0]	YGain	R/W	Luminance Gain Adjustment	0: Minimum 128: Default 255: Maximum

28.5.7 Y Delay Register

P_TV1_Y_DELAY		0x93020214								TV Y Delay Register							
Bit	Function	15	14	13	12	11	10	9	8								
Function		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		7	6	5	4	3	2	1	0								
		YDelay															
		-	-	-	-	-	-	-	-								
		-	-	-	-	-	-	-	-	0	0						

Bit	Function	Type	Description	Condition
[15:2]	-	-	Reserved	-
[1:0]	YDelay	R/W	Luminance delay adjustment	0: No delay 1: 1 luminance delay 2: 2 luminance delay 3: 3 luminance delay

28.5.8 Vertical Position Register

P_TV1_V_POSITION 0x93020218 TV Vertical Position Register								
Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0
	VPOS							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:0]	VPOS	R/W	Vertical Start Position Offset Adjustment It will not handle overflow condition	0: default 255: maximum

28.5.9 Horizontal Position Register

P_TV1_H_POSITION 0x9302021C TV Horizontal Position Register								
Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0
	HPOS							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:0]	HPOS	R/W	Horizontal Start Position Offset Adjustment +127: Maximum 0: Default -128: Minimum (clipped at internal horizontal count = 0)	+127: Maximum 0: Default -128: Minimum (clipped at internal horizontal count = 0)

28.5.10 Video DAC Control Register

P_TV1_VIDEODAC 0x93020220 TV Video DAC Control Register								
Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
CCIR_SYNC	CCIR_656	CBSWAP	CCIR_EN	-	VDAC_UD	VDAC_BGEN	VDAC_EN
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
7	CCIR_SYNC	R/W	CCIR-601 Sync Width Selection	0: Depend on HDSEL/VDSEL 1: 1 dclk (1/27MHz)
6	CCIR_656	R/W	CCIR-656 or CCIR-601 Format	0: CCIR-601 1: CCIR-656
5	CBSWAP	R/W	CCIR-601/656 Cb/Cr Sequence Swap	0: CbYCrY 1: CrYCbY
4	CCIR_EN	R/W	Use External CCIR 8-bit Input DAC (i.e., TV Encoder)	0: Disable 1: Enable
3	-	-	Reserved	-
2	VDAC_UD	R/W	VDAC Count Up/Down	0: Count down 1: Count up
1	VDAC_BGEN	R/W	VDAC Band Gap Enable	0: Disable 1: Enable
0	VDAC_EN	R/W	VDAC Enable	0: Disable 1: Enable

28.6 Recommended Settings

Generalplus highly recommends users take the following table as a quick and optimal reference for TV encoder applications to facilitate the configuration of NTSC and PAL - the two most popular systems used in today's TV industry.

QVGA (Non-interlaced)

Control Register	NTSC-J	PAL_BDGHI
P_TV1_CTRL	0x001F	0x001F
P_TV1_SATURATION	0x815E	0x8161
P_TV1_HUE	0x5A00	0x5A00
P_TV1_BRIGHTNESS	0x7600	0x7600
P_TV1_SHARPNESS	0x0000	0x0000
P_TV1_Y_GAIN	0x6D4E	0x6D5F
P_TV1_Y_DELAY	0xB800	0xCBE0

Control Register	NTSC-J	PAL_BDGHI
P_TV1_V_POSITION	0x1E00	0x8A00
P_TV1_H_POSITION	0xF003	0x0905
P_TV1_VIDEODAC	0x2100	0x2A00

VGA (Interlaced)

Control Register	NTSC-J	PAL_BDGHI
P_TV1_CTRL	0x002F	0x002F
P_TV1_SATURATION	0x815E	0x8161
P_TV1_HUE	0x5A00	0x5A00
P_TV1_BRIGHTNESS	0x7600	0x7600
P_TV1_SHARPNESS	0x0000	0x0000
P_TV1_Y_GAIN	0x6D4E	0x6D4E
P_TV1_Y_DELAY	0xB800	0xCBE0
P_TV1_V_POSITION	0x1E00	0x8A00
P_TV1_H_POSITION	0xF003	0x0905
P_TV1_VIDEODAC	0x2100	0x2A00

VGA (Non-interlaced)

Control Register	NTSC-J	PAL_BDGHI
P_TV1_CTRL	0x003F	0x003F
P_TV1_SATURATION	0x815E	0x8161
P_TV1_HUE	0x5A00	0x5A00
P_TV1_BRIGHTNESS	0x7600	0x7600
P_TV1_SHARPNESS	0x0000	0x0000
P_TV1_Y_GAIN	0x6D4E	0x6D5F
P_TV1_Y_DELAY	0xB800	0xCBE0
P_TV1_V_POSITION	0x1E00	0x8A00
P_TV1_H_POSITION	0xF003	0x0905
P_TV1_VIDEODAC	0x2100	0x2A00

D1 (Progressive)

Control Register	NTSC-J	PAL_BDGHI
P_TV1_CTRL	0x003F	0x003F
P_TV1_CTRL2	0x0001	0x0001
P_TV1_SATURATION	0x815E	0x8161
P_TV1_HUE	0x5A00	0x5A00
P_TV1_BRIGHTNESS	0x7600	0x7600

Control Register	NTSC-J	PAL_BDGHI
P_TV1_SHARPNESS	0x0000	0x0000
P_TV1_Y_GAIN	0x6D4E	0x6D5F
P_TV1_Y_DELAY	0xB800	0xCBE0
P_TV1_V_POSITION	0x1E00	0x8A00
P_TV1_H_POSITION	0xF0F6	0x09F6
P_TV1_VIDEODAC	0x2100	0x2A00

29 SPU Introduction

29.1 General Description

The SPU (Sound Process Unit) inside the GP6 series is designed to emulate various types of musical instruments via programming its tone ROM and controlling the envelope slope for each channel. Each channel can further be defined as a speech channel to produce PCM-format sound effects, e.g. percussion, animal sounds, gun, explosions accompanied with main music rhythm.

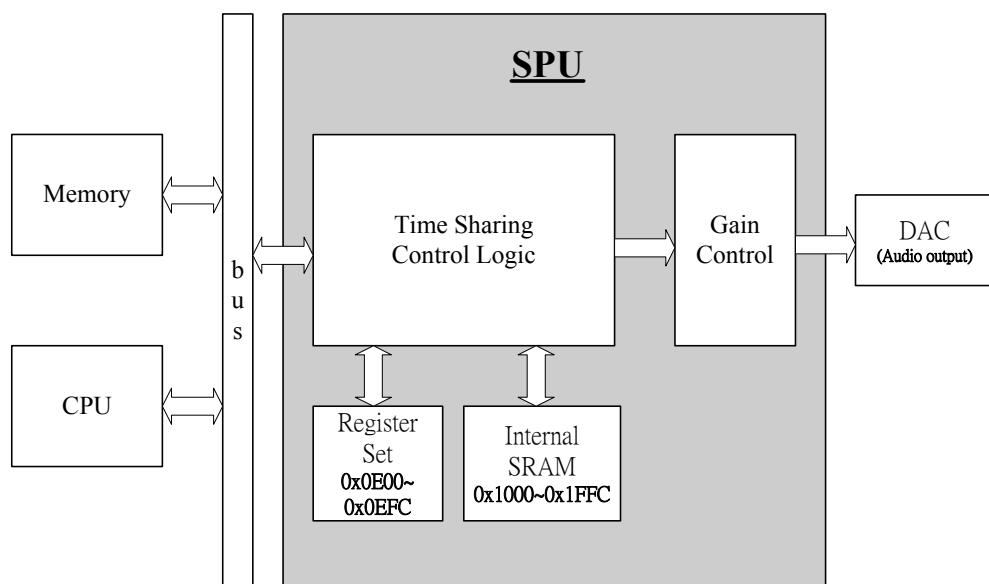


Figure 29-1 Block Diagram

29.2 Feature

- 8-bit /16-bit (software mode) stereo PCM
- 4-bit ADPCM/ADPCM36
- 32 channels
- Up to 256M x 16-bit addressing simultaneously
- 7-bit main volume control
- MIDI format gain control for each R/L channel of 32 channels
- Max. 256 piece-wise envelope with repeat for envelope control
- 32-channel IRQ functions and Beat event IRQ and Envelope IRQ
- Tone-Color & envelope address not limited to bank (64K). The Tone-Color & envelope can be controlled by hardware or manual (software) mode.
- Auto tone-color interpolation function.
- Up to 281.25kHz tone-color sample rate.

- Hardware Release Tone Color function.
- Individual Channel Release function.
- Hardware Pitch Bend function
- Auto volume control (Compressor).

29.3 Internal Memory Mapping

- SPU Base Address: 0xD0400000
- Register Set: 0x0E00 ~ 0x0E7C(Channel 0~15), 0x0E80 ~ 0x0EFC(Channel 16~31)
- Internal SRAM: 32 channels

Figure 29-2 Internal Memory Mapping

Channel	Phase Address Port(Hex)	Attribute Address Port(Hex)
0	1800~183C	1000~103C
1	1840~187C	1040~107C
2	1880~18BC	1080~10BC
3	18C0~18FC	10C0~10FC
4	1900~193C	1100~113C
5	1940~197C	1140~117C
6	1980~19BC	1180~11BC
7	19C0~19FC	11C0~11FC
8	1A00~1A3C	1200~123C
9	1A40~1A7C	1240~127C
10	1A80~1ABC	1280~12BC
11	1AC0~1AFC	12C0~12FC
12	1B00~1B3C	1300~133C
13	1B40~1B7C	1340~137C
14	1B80~1BBC	1380~13BC
15	1BC0~1BFC	13C0~13FC
16	1C00~1C3C	1400~143C
17	1C40~1C7C	1440~147C
18	1C80~1CBC	1480~14BC
19	1CC0~1CFC	14C0~14FC
20	1D00~1D3C	1500~153C
21	1D40~1D7C	1540~157C
22	1D80~1DBC	1580~15BC

Channel	Phase Address Port(Hex)	Attribute Address Port(Hex)
23	1DC0~1DFC	15C0~15FC
24	1E00~1E3C	1600~163C
25	1E40~1E7C	1640~167C
26	1E80~1EBC	1680~16BC
27	1EC0~1EFC	16C0~16FC
28	1F00~1F3C	1700~173C
29	1F40~1F7C	1740~177C
30	1F80~1FBC	1780~17BC
31	1FC0~1FFC	17C0~17FC

29.4 Gain Control

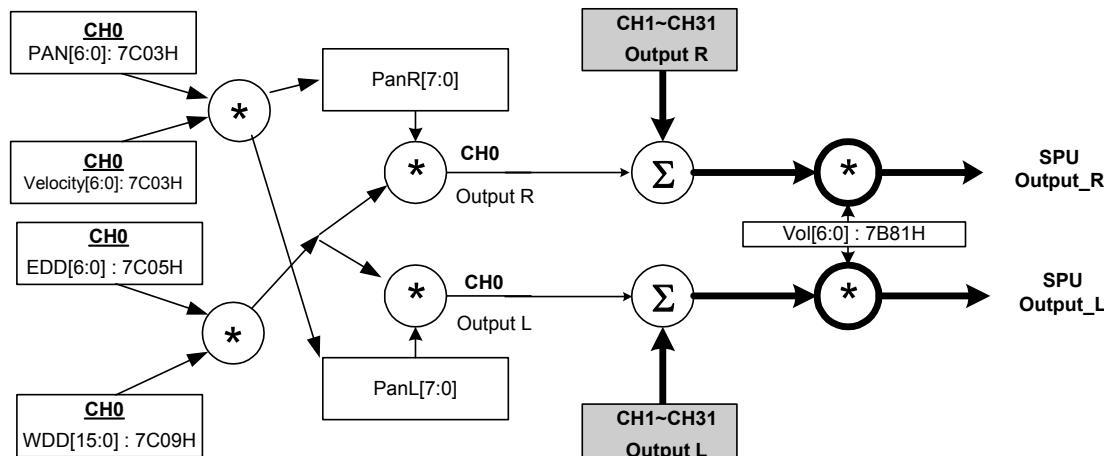


Figure 29-3 SPU structure diagram

The SPU features 32 channels. The above structure diagram only illustrates the Channel 0. Similarities are applied for other channels. In the diagram, the multiplication outcome of tone-color (timber) and envelope is delivered to each R/L channel to complete the panning effect. After all channels are added up, it outputs via the main volume control.

29.5 Fundamentals

Sound is composed of three essential elements: **pitch**, **tone-color (timber)**, and **envelope/ADSR**.

The tone color can be grouped to musical sound and noise in digital music.

Musical Sound: With characteristics of cycles, vibration in certain pattern, e.g. piano, xylophone, violin, etc.

Noise: With the characteristic of vibration in random, e.g. animal sound.

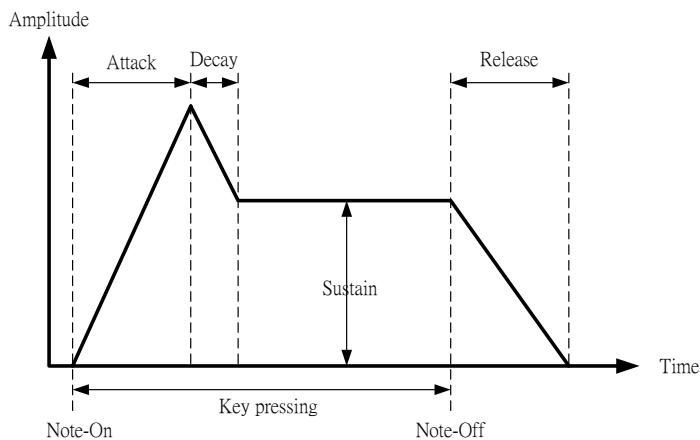


Figure 29-4 ADSR envelope

29.6 Musical Sound

Because musical sound is a cycled wave, it is possible to be synthesized by repetition (must be supported by hardware). The instrument can be expressed by envelope and tone-color. An envelope basically contains four major sections- **Attack**, **Decay**, **Sustain**, and **Release**, or **ADSR**, see the waveform above. The following diagram indicates the modulation outcome from tone-color and envelope.

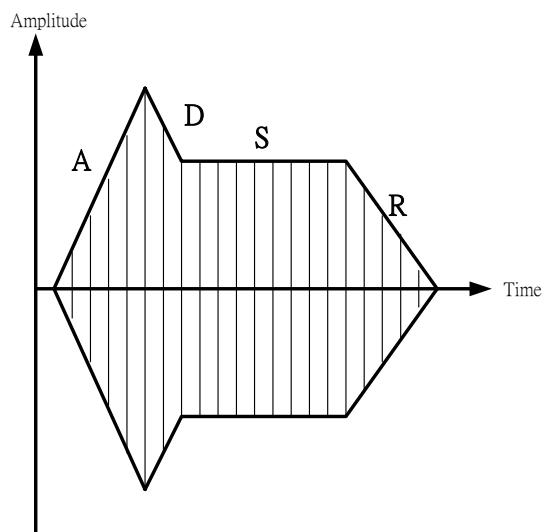
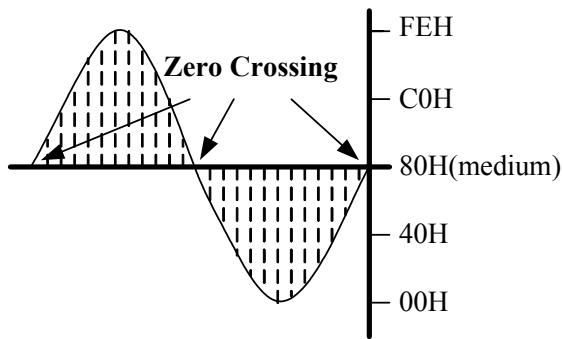


Figure 29-5 ADSR's waveform

An envelope contains a series of repeating tone-color wave. The ADSR illustrated here is only the one of all acoustical models. Not all instruments are produced by this mode.

- 8 bits tone-color code is expressed by unsigned. The medium is "0x80"; max is "0xFE" and min is "0x00". The "0xFF" is defined as the end code of tone-color; therefore, data itself cannot be "0xFF".



Note: FFH defined as the end code of tone-Color

Figure 29-6 Sin Wave of Tone-Color

- The envelope data is expressed by 7 bits unsigned. The maximum value is "0x7F", and minimum is "0x00".
- There are two envelop modes supported, auto and manual modes.
 - a. **Auto Mode:** The CPU writes specific parameters to **register set** and **internal SRAM**. Then, the hardware loads tone-color and envelope data automatically based on the given parameters. The envelope slope of an instrument can be up to 128 types. In addition, it has repeating function, which can be used to synthesize envelope LFO (Low Frequency Oscillation) easily.
 - b. **Envelope Manual Mode:** The envelope data is controlled by software.

29.7 Control Register

Name	Address	Description
P_SPU_CH_EN	0x9300BE00	SPU channel 0~15 enable
P_SPU_MAIN_VOLUME	0x9300BE04	SPU main volume
P_SPU_CH_FIQ_EN	0x9300BE08	SPU channel 0~15 FIQ enable
P_SPU_CH_FIQ_STATUS	0x9300BE0C	SPU channel 0~15 FIQ status
P_SPU_BEAT_BASE_COUNTER	0x9300BE10	SPU beat base counter
P_SPU_BEAT_COUNTER	0x9300BE14	SPU beat counter
P_SPU_ENV_CLK_CH0_3	0x9300BE18	SPU channel 0~3 envelope interval selection
P_SPU_ENV_CLK_CH4_7	0x9300BE1C	SPU channel 4~7 envelope interval selection
P_SPU_ENV_CLK_CH8_11	0x9300BE20	SPU channel 8~11 envelope interval selection

Name	Address	Description
P_SPU_ENV_CLK_CH12_15	0x9300BE24	SPU channel 12~15 envelope interval selection
P_SPU_ENV_RAMP_DOWN	0x9300BE28	SPU channel 0~15 envelope fast ramp down
P_SPU_CH_STOP_STATUS	0x9300BE2C	SPU channel 0~15 stop channel status
P_SPU_CH_ZC_ENABLE	0x9300BE30	SPU channel 0~15 zero crossing enable
P_SPU_CONTROL_FLAG	0x9300BE34	SPU control flags
P_SPU_COMPRESSOR_CONTROL	0x9300BE38	SPU compressor control
P_SPU_CH_STATUS	0x9300BE3C	SPU channel 0~15 status
P_SPU_WAVE_IN_LEFT	0x9300BE40	SPU left channel mixer input
P_SPU_WAVE_IN_RIGHT	0x9300BE44	SPU right channel mixer input
P_SPU_WAVE_OUT_LEFT	0x9300BE48	SPU wave output left of 32 channel + software channel
P_SPU_WAVE_OUT_RIGHT	0x9300BE4C	SPU wave output right of 32 channel + software channel
P_SPU_CH_REPEAT_EN	0x9300BE50	SPU channel 0~15 repeat enable control
P_SPU_CH_ENV_MODE	0x9300BE54	SPU channel 0~15 envelope mode
P_SPU_CH_TONE_RELEASE	0x9300BE58	SPU channel 0~15 tone release control
P_SPU_CH_IRQ_STATUS	0x9300BE5C	SPU channel 0~15 envelope IRQ status
P_SPU_CH_PITCH_BEND_EN	0x9300BE60	SPU channel 0~15 pitch bend enable
P_SPU_ATTACK_RELEASE_TIME	0x9300BE68	SPU attack/release time control
P_SPU_BENK_ADDR	0x9300BE7C	SPU wave table's bank address
P_SPU_CH_EN_HI	0x9300BE80	SPU channel 16~31 enable
P_SPU_CH_FIQ_EN_HI	0x9300BE88	SPU channel 16~31 FIQ enable
P_SPU_CH_FIQ_STATUS_HI	0x9300BE8C	SPU channel 16~31 FIQ status
P_SPU_POST_WAVE_CONTROL	0x9300BE94	SPU post wave counter and control
P_SPU_ENV_CLK_CH16_19	0x9300BE98	SPU channel 16~19 envelope interval selection
P_SPU_ENV_CLK_CH20_23	0x9300BE9C	SPU channel 20~23 envelope interval selection
P_SPU_ENV_CLK_CH24_27	0x9300BEA0	SPU channel 24~27 envelope interval selection
P_SPU_ENV_CLK_CH28_31	0x9300BEA4	SPU channel 28~31 envelope interval selection

Name	Address	Description
P_SPU_ENV_RAMP_DOWN_HI	0x9300BEA8	SPU channel 16~31 envelope fast ramp down
P_SPU_CH_ZC_ENABLE_HI	0x9300BEB0	SPU channel 16~31 zero crossing enable
P_SPU_CH_STOP_STATUS_HI	0x9300BEAC	SPU channel 16~31 stop channel status
P_SPU_CH_STATUS_HI	0x9300BEBC	SPU channel 16~31 status
P_SPU_POST_WAVE_OUT_LEFT	0x9300BEC8	SPU wave output left of 32 channel
P_SPU_POST_WAVE_OUT_RIGHT	0x9300BECC	SPU wave output right of 32 channel
P_SPU_CH_REPEAT_EN_HI	0x9300BED0	SPU channel 16~31 repeat enable control
P_SPU_CH_ENV_MODE_HI	0x9300BED4	SPU channel 16~31 envelope mode
P_SPU_CH_TONE_RELEASE_HI	0x9300BED8	SPU channel 16~31 tone release control
P_SPU_CH_IRQ_STATUS_HI	0x9300BEDC	SPU channel 16~31 envelope IRQ status
P_SPU_CH_PITCH_BEND_EN_HI	0x9300BEE0	SPU channel 16~31 pitch bend enable

29.8 Control Register List

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description																																		
0x0E00	ChEn[15:0]																Channel Enable																																		
0x0E04	--								Vol[6:0]								Main volume																																		
0x0E08	ChFIQEn[15:0]																Channel FIQ Enable																																		
0x0E0C	ChFIQSts[15:0]																Channel FIQ Status																																		
0x0E10	--				BeatBaseCnt[10:0]												Beat base counter																																		
0x0E14	BIE	BIS	BeatCnt[13:0]														Beat counter																																		
0x0E18	EnvClk[15:0], Ch 3~0																Envelope interval select																																		
0x0E1C	EnvClk[31:16], Ch 7~4																Envelope interval select																																		
0x0E20	EnvClk[47:32], Ch 11~8																Envelope interval select																																		
0x0E24	EnvClk[63:48], Ch 15~12																Envelope interval select																																		
0x0E28	EnvRampDown[15:0]																Envelope fast ramp down																																		
0x0E2C	ChStopSts[15:0]																Stop channel status																																		
0x0E30	--																																																		
0x0E34	Saturate	--	ENDAM	--	CompEn	NoHigh	NoInt	--	VolSel	FOF	--	Init	--	--	--	--	Control Flags																																		
0x0E38	Peak	Threshold						AttScale	RelScale	DisZC	Ratio				Compressor Control																																				
0x0E3C	ChSts[15:0]																Channel status																																		
0x0E40	WaveInL[15:0]																Left channel mixer input																																		
0x0E44	WaveInR[15:0]																Right channel mixer input																																		
0x0E48	WaveOutL[15:0]																Left channel mixer output																																		
0x0E4C	WaveOutR[15:0]																Right channel mixer output																																		



GPL32900A Programming Guide

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description								
0x0E50	ChRepeatEn[15:0]																Channel Repeat Enable control								
0x0E54	ChEnvMode[15:0]																Channel Env Mode								
0x0E58	ChToneRelease[15:0]																Channel Tone Release Control								
0x0E5C	ChEnvIrqSts[15:0]																Channel Env Irq Status								
0x0E60	ChPitchBendEn[15:0]																Channel Pitch Bend Enable								
0x0E64	--																								
0x0E68	AttackTime								ReleaseTime								Attack/Release Time Control								
0x0E6C	--																								
0x0E70	--																								
0x0E74	--																								
0x0E78	--																								
0x0E7C									BankAddr[8:0]								Wave Table's Bank Address								

Note: All control registers are active high.



GPL32900A Programming Guide

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description
0x0E80																	ChEn[31:16]
0x0E84																	-
0x0E88																	ChFIQEn[31:16]
0x0E8C																	ChFIQSSts[31:16]
0x0E90																	--
0x0E94	PWEN	PWDS	--	PWCK	PWIE	PWIS	--	PWSE	PWSIL	--	--					PWF0V	Post Wave counter and control
0x0E98																	EnvClk[79:64], Ch 19~16
0x0E9C																	EnvClk[95:80], Ch 23~20
0x0EA0																	EnvClk[111:96], Ch 27~24
0x0EA4																	EnvClk[127:112], Ch 31~28
0x0EA8																	EnvRampDown[31:16]
0x0EAC																	ChStopSts[31:16]
0x0EB0																	--
0x0EB4																	--
0x0EB8																	--
0x0EBC																	ChSts[31:16]
0x0EC0																	--
0x0EC4																	--
0x0EC8																	PWaveOutL[15:0]
																	Post Wave Output



GPL32900A Programming Guide

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description
0x0ECC	PWaveOutR[15:0]																Post Wave Output
0x0ED0	ChRepeatEn[31:16]																Channel Repeat Enable control
0x0ED4	ChEnvMode[31:16]																Channel Env Mode
0x0ED8	ChToneRelease[31:16]																Channel Tone Release Control
0x0EDC	ChEnvIrqSts[31:16]																Channel Env Irq Status
0x0EE0	ChPitchBendEn[31:16]																Channel Pitch Bend Enable

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description
0x0F00	DVB[15:0], Ch 3~0																DVB selection
0x0F04	DVB[31:16], Ch 7~4																DVB selection
0x0F08	DVB[47:32], Ch 11~8																DVB selection
0x0F0C	DVB[63:48], Ch 15~12																DVB selection
0x0F10	DVB[79:64], Ch 19~16																DVB selection
0x0F14	DVB[95:80], Ch 23~20																DVB selection
0x0F18	DVB[111:96], Ch 27~24																DVB selection
0x0F1C	DVB[127:112], Ch 31~28																DVB selection

29.9 Internal SRAM List

Internal Attribute SRAM Format, Channel 0 ~ Channel 31

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0																					
0x1000+0x40*X	Wave Address	Waddr[15:0]																																				
0x1004+0x40*X	Mode	ADPCM	16M	ToneMode	LoopAddr[21:16]						Waddr[21:16]																											
0x1008+0x40*X	Loop Address	LoopAddr[15:0]																																				
0x100C+0x40*X	Pan	-	Pan[6:0]						-	ChVolumn[6:0]																												
0x1010+0x40*X	Envelope0	Repeat Period	EnvTarget[6:0]						EnvSign	EnvInc[6:0]																												
0x1014+0x40*X	Envelope Data	EnvCnt[7:0]						-	EDD[6:0]																													
0x1018+0x40*X	Envelope1	RpCnt				Rpt		EnvLoad[7:0]																														
0x101C+0x40*X	Envelope Address	IrqFireAddress[8:0]						IrqEn	Eaddr[21:16]																													
0x1020+0x40*X	Envelope Address	Eaddr[15:0]																																				
0x1024+0x40*X	Wave Data 0	WDD0[15:0]																																				
0x1028+0x40*X	Envelope Loop Control	RampDownoffset[6:0]						Eaoffset[8:0]																														
0x102C+0x40*X	Wave Data	WDD[15:0]																																				
0x1030+0x40*X	-																																					
0x1034+0x40*X	ADPCM Sel	ADPCM36	PointNumber																																			
0x1038+0x40*X	Wave Loop Address HI	-				LoopAddr[27:22]						Waddr[27:22]																										
0x103C+0x40*X	Envelope Address HI	-																																				



GPL32900A Programming Guide

Internal Phase SRAM Channel Format, Channel 0 ~ Channel 31

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0x1800+0x40*X	Phase High																LFO Counter1[9:0]
0x1804+0x40*X	Phase Accumulator High																LFO Counter2[9:0]
0x1808+0x40*X	TargetPhase High																LFO Setting[9:0]
0x180C+0x40*X	RampDownClk								DAM								RampDownClk
0x1810+0x40*X	Phase																Phase[15:0]
0x1814+0x40*X	Phase Accumulator																PhaseAcc[15:0]
0x1818+0x40*X	Target Phase																Target Phase[15:0]
0x181C+0x40*X	Phase Control			PhaseTimeStep		Sign											PhaseOffset
0x1820+0x40*X	-																
0x1824+0x40*X	-																
0x1828+0x40*X	-																
0x182C+0x40*X	-																
0x1830+0x40*X	-																
0x1834+0x40*X	-																
0x1838+0x40*X	-																
0x183C+0x40*X	-																

29.10 SPU Control Register

29.10.1 SPU Control Flag

P_SPU_CtrFlag								0x9300BE34								SPU Control Flag								
Bit	15	14	13	12	11	10	9	8																
Function	Saturate	--	ENDAM	--	CompEn	NoHigh	NoInt	-																
Default	0	0	0	0	0	0	0	0																
	7	6	5	4	3	2	1	0																
	VolSel		FOF	--	Init	--																0		
	0	0	0	0	0	0	0	0																

Bit	Function	Type	Description	Condition
15	Saturate	R/W	This bit represents whether the output signal is saturated, meaning the output signal will be clipped if it exceeds the maximum range.	Write 1 : Clear Saturate flag Write 0 : No operation Read 1 :Saturate happened Read 0 :Saturate un happened
14	-	-	Reserved	
13	ENDAM	R/W	This bit is the global enable bit of amplitude modulation. After enable this bit, programmer can use DAM setting in the phase ram to enable the amplitude modulation function by channel.	0: Disable amplitude modulation 1: Enable amplitude modulation.
12	-	-	Reserved	
11	CompEn		When CompEn is set as "1", the internal compressor will activate and dynamically control the output volume. Using the parameters stored in 0x7B8E and 0x7B9A. e.g. AttackTime, ReleaseTime, Threshold, Ratio, etc.	0: Compressor off 1: Compressor on
10	NoHigh		When NoHigh is "0", the internal high quality interpolation logic will be used to compensate the error caused by the phase-jitter effect	0:High Quality Interpolation on 1:High Quality Interpolation off

Bit	Function	Type	Description	Condition
9	NoInt		<p>When NoInt is "0", the internal interpolation logic will be used to smooth the output data</p> <p>When the phase is smaller than 0x10000, which causes the total bandwidth lower. If the FOF bit is set when program is running, setting NoInt to "1" may improve the overall performance.</p>	0: Interpolation on 1: Interpolation off
8	-	-	Reserved	
[7:6]	VolSel		When VolSel is set to "0", the volume of a single channel will be 1/32 of max volume. When VolSel is given a non-zero value, the volume of a single channel becomes larger.	00: 1/32 01: 1/8 10: 1/2 11: 1
5	FOF		This flag indicates the SPU is unable to handle such high sample rate and it will cause data lost. The maximum service rate of GP6 is approx. 2500 KHz. It means each channel works at sample rate of approx. 78KHz (2500KHz/32ch=78K/channel) without any data lost. Programmer can determine which channel needs more or few resources. If total bandwidth not exceed 2500 KHz, data lost will not occur.	Write 1 : Clear FOF flag Write 0 : No operation Read 1 : FOF happened Read 0 : FOF un happened
4	-	-	Reserved	
3	Init		Initializes channel's accumulator. The initialization ensures the system is able to operate correctly.	0 : no operation 1 : initializes accumulator
[2:0]	-	-	Reserved	

29.10.2 Channel Enable Control

Before enabling a channel(ChEn [x]=1) , it is a **must** to initialize the attribute and phase RAM. Please refer to **Programming Note** section for initialization. Writing “1” to ChEn[x] means enabling the channel or “0” to disable it. When ChStopSts is set to “1”, SPU starts to function (play music or sound effect). Programmer can inspire whether SPU is playing or stops playing via checking the ChSts flag.

P_SPU_CtrChEn								0x9300BE00								Channel[15:0] Enable Control								
Bit	15	14	13	12	11	10	9	8	ChEn[15:0]								ChEn[15:0]							
Function									ChEn[15:0]								ChEn[15:0]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description								Condition
[15:0]	ChEn[15:0]	R/W	Channel[15:0] Enable Control and Status								0 : Channel Disable 1 : Channel Enable

P_SPU_CtrChEn_H								0x9300BE80								Channel[31:16] Enable Control								
Bit	15	14	13	12	11	10	9	8	ChEn[31:16]								ChEn[31:16]							
Function									ChEn[31:16]								ChEn[31:16]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description								Condition
[15:0]	ChEn[31:16]	R/W	Channel[31:16] Enable Control and Status								0 : Channel Disable 1 : Channel Enable

P_SPU_CtrChStopSts
0x9300BE2C
Channel[15:0] Stop status

Bit	15	14	13	12	11	10	9	8
Function	ChStopSts[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChStopSts[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	ChStopSts[15:0]	R/W	Channel stop status				Write 1 : Clear Stop status Write 0 : No operation Read 1 : Channel is stopped Read 0 : Channel is ready

P_SPU_CtrChStopSts
0x9300BEAC
Channel[31:16] Stop status

Bit	15	14	13	12	11	10	9	8
Function	ChStopSts[31:16]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChStopSts[31:16]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	ChStopSts[31:16]	R/W	Channel stop status				Write 1 : Clear Stop status Write 0 : No operation Read 1 : Channel is stopped Read 0 : Channel is ready

P_SPU_CtrChSts
0x9300BE3C
Channel[15:0] status

Bit	15	14	13	12	11	10	9	8
Function	ChSts[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Chsts[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	Chsts[15:0]	R/W	Channel[15:0] Status				Read 0 : channel is valid Read 1 : channel is busy

P_SPU_CtrChsts 0x9300BEBC Channel[31:16] status

Bit	15	14	13	12	11	10	9	8
Chsts[31:16]								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Chsts[31:16]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	Chsts[31:16]	R/W	Channel[31:16] Status				Read 0 : channel is valid Read 1 : channel is busy

29.10.3 Main Volume Control

The P_SPU_CtrMainVol controls the volume to the entire system. It is to assure that output does not overflow when all 32 channels are turned on with max WDD, EDD, velocity and global volume.

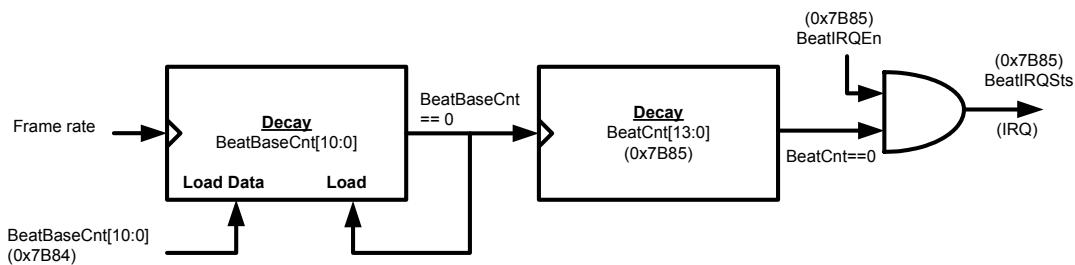
P_SPU_CtrMainVol 0x9300BE04 Main volume control

Bit	15	14	13	12	11	10	9	8
-								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
VOL [6:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]	Reserved	R	Reserved	
[6:0]	SPU_CtrMainVol	R/W	Main volume control to entire system.	0x00~0x7F

29.10.4 Beat Control

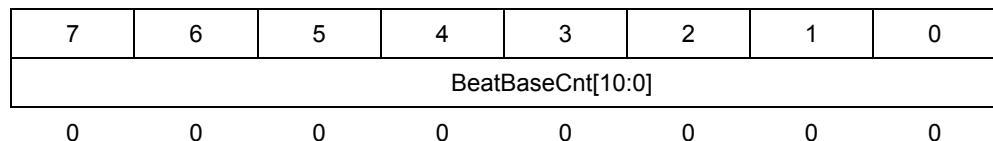


Beat trigger count will be subtracted by one every four frames and beat base count will be loaded into beat trigger count when it reaches zero. Beat count will be subtracted by one every time when beat trigger count reaches zero.

$$\text{BeatIRQ_Period} = (\text{BeatBaseCnt} * \text{BeatCnt}) * 4 * \text{Frame rate} \quad (\text{Frame-Rate} = 281.25\text{kHz}).$$

P_SPU_CtrBeatBase 0x9300BE10 Beat Base Counter

Bit	15	14	13	12	11	10	9	8
Function	--							
Default	0	0	0	0	0	0	0	0



Bit	Function	Type	Description	Condition
[15:11]	Reserved	R	Reserved	
[10:0]	BeatBaseCnt[10:0]	R/W	Beat base count will be subtracted by one every 4 frames.	0x000~0x7FF

P_SPU_CtrBeatCnt 0x9300BE14 Beat Counter

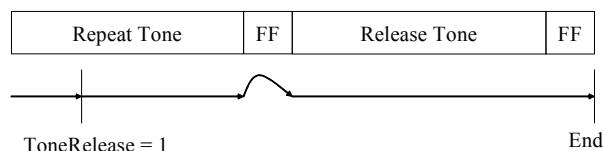
Bit	15	14	13	12	11	10	9	8		
Function	BeatIRQEn	BeatIRQSts	BeatCnt[13:0]							
Default	0	0	0	0	0	0	0	0		

7	6	5	4	3	2	1	0
BeatCnt[13:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	BeatIRQEn	R/W	When Beat event (Beat decay to 0) is enabled, Beat IRQ activates.	Write 1: No operation Write 0: Clear BeatIRQ. Read 1: BeatIRQ enable Read 0: BeatIRQ disable
14	BeatIRQSts	R	Current Beat IRQ (IRQ 4) event status.	0: BeatIRQ no request 1: BeatIRQ request
[13:0]	BeatCnt[13:0]	R/W	BeatIRQ_Period = BeatCnt * BeatBaseCnt * 4 * Frame.	0x0000~0x3FFF

29.10.5 Tone Control

SPU output can be obtained from WaveOut and mixed Waveln with SPU. For example, to achieve Echo effect, programmer shall first acquire SPU's output from waveOut, and then mix Waveln with SPU after a certain period of prolongation. Release tone-color (timber) can be played via the ChToneRelease function. Channel will complete the current tone-color play and begin to play completed release tone.



P_SPU_CtrWavelnL 0x9300BE40 An additional software channel

Bit	15	14	13	12	11	10	9	8
WavelnL[15:0]								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
WavelnL[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	WaveInL[15:0]	R/W	The data is mixed with the left output of SPU.					

P_SPU_CtrWaveInR
0x9300BE44
An additional software channel

Bit	15	14	13	12	11	10	9	8
Function	WaveInR[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
WaveInR[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	WaveInR[15:0]	R/W	The data is mixed with the right output of SPU.					

P_SPU_CtrWaveOutL
0x9300BE48
The 16-bits output of SPU

Bit	15	14	13	12	11	10	9	8
Function	WaveOutL[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
WaveOutL[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	WaveOutL[15:0]	R/W	The data is the final left output outcome of 32-channel + software channel.					

P_SPU_CtrWaveOutR
0x9300BE4C
The 16-bits output of SPU

Bit	15	14	13	12	11	10	9	8
Function	WaveOutR[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
WaveOutR[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	WaveOutR[15:0]	R/W	The data is the final right output outcome of 32-channel + software channel.				

P_SPU_CtrChToneRelease 0x9300BE58 Channel[15:0] I tone release control register

Bit	15	14	13	12	11	10	9	8
Function	ChToneRelease[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChToneRelease[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	ChToneRelease[15:0]	R/W	This register controls the channel tone release. This bit is cleared automatically after channel is stopped.				0: No operation 1: Tone Release

P_SPU_CtrChToneRelease_H 0x9300BED8 Channel[31:16] I tone release control register

Bit	15	14	13	12	11	10	9	8
Function	ChToneRelease[31:16]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChToneRelease[31:16]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ChToneRelease[31:16]	R/W	This register controls the channel tone release. It is cleared automatically after channel is stopped.	0: No operation 1: Tone Release

29.10.6 SoftChannel Control

SoftChannel Control is another route to play acoustic using software approach. When an end code of tone-color is reached, an IRQ interrupt is established. For example, cooperating SACM library using A/B buffer, wave address is set to A and loop address to B at initial state before playing wave address. When A buffer is playing at the end code, it plays loop address (B buffer) and establishes an interrupt. Programmer, at this moment, shall change the loop address to A buffer. When B buffer plays at the end code, it auto play A buffer again and issue an interrupt. Programmer then change the loop address to B buffer till the end of acoustic.

P_SPU_CtrChFiqEn **0x9300BE08**

Channel[15:0] FIQ Enable

Bit	15	14	13	12	11	10	9	8
Function	ChFIQEn[15:0]							

Default

7	6	5	4	3	2	1	0
ChFIQEn[15:0]							
0	0	0	0	0	0	0	0

Bit

15 14 13 12 11 10 9 8

Function	ChFIQEn[15:0]							
----------	---------------	--	--	--	--	--	--	--

Default

0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[15:0]	ChFIQEn[15:0]	R/W	Control FIQ enable. Refer to register 0x0E0C, 0x0E8C	0: Channel FIQ Disable 1: Channel FIQ Enable

P_SPU_CtrChFiqEn_H

0x9300BE88

Channel[31:16] FIQ Enable

Bit

15 14 13 12 11 10 9 8

Function	ChFIQEn[31:16]							
----------	----------------	--	--	--	--	--	--	--

Default

0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
ChFIQEn[31:16]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	ChFIQEn[31:16]	R/W	Control FIQ enable. Refer to register 0x0E0C, 0x0E8C					0: Channel FIQ Disable 1: Channel FIQ Enable

P_SPU_CtrChFiqSts 0x9300BE0C Channel[15:0] FIQ Status

Bit	15	14	13	12	11	10	9	8
ChFIQSts[15:0]								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChFIQSts[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	ChFIQSts[15:0]	R/W	ChFIQSts[x] occurs when END CODE is reached (0xFFFF for 16-bit mode and 0xFF for 8-bit mode). FIQ occurs when the corresponding ChFIQEn[x] and ChFIQSts[x] are activated.					Write 1: Clear FIQ Write 0: No operation Read 1: Channel FIQ active Read 0: Channel FIQ inactive

P_SPU_CtrChFiqSts_H 0x9300BE8C Channel[31:16] FIQ Status

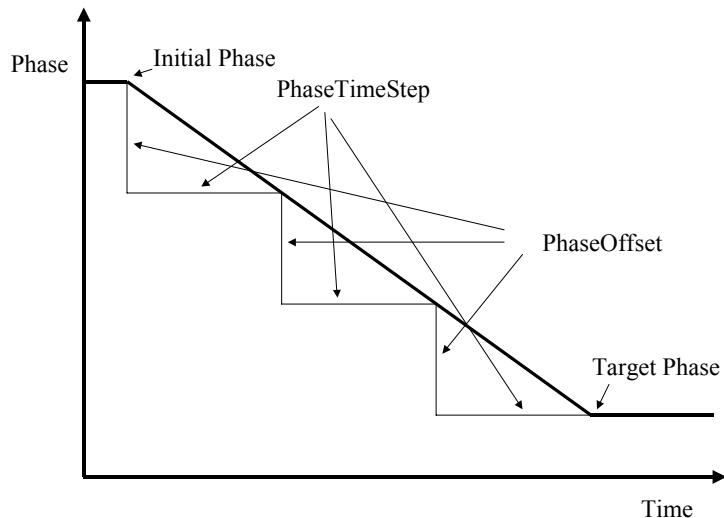
Bit	15	14	13	12	11	10	9	8
ChFIQSts[31:16]								
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChFIQSts[31:16]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ChFIQSts[31:16]	R/W	ChFIQSts[x] occurs when END CODE is reached (0xFFFF for 16-bit mode and 0xFF for 8-bit mode). FIQ occurs when the corresponding ChFIQEn[x] and ChFIQSts[x] are activated.	Write 1: Clear FIQ Write 0: No operation Read 1: Channel FIQ active Read 0: Channel FIQ inactive.

29.10.7 PitchBend Enable Control

The primary function of this register is to control the pitch bend for each channel. The TargetPhase[18:0], PhaseOffset[11:0], PhaseTimeStep[2:0], and PhaseSign will be used to increase/decrease phase. Please refer to **Internal SRAM** for more information.



P_SPU_CtrChPitchBendEn 0x9300BE60 Channel[15:0] | pitch bend enable

Bit	15	14	13	12	11	10	9	8
Function	ChPitchBendEn[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ChPitchBendEn[15:0]	R/W	This register controls the pitch bend for each channel.	0: Pitch bend disabled. 1: Pitch bend enabled.

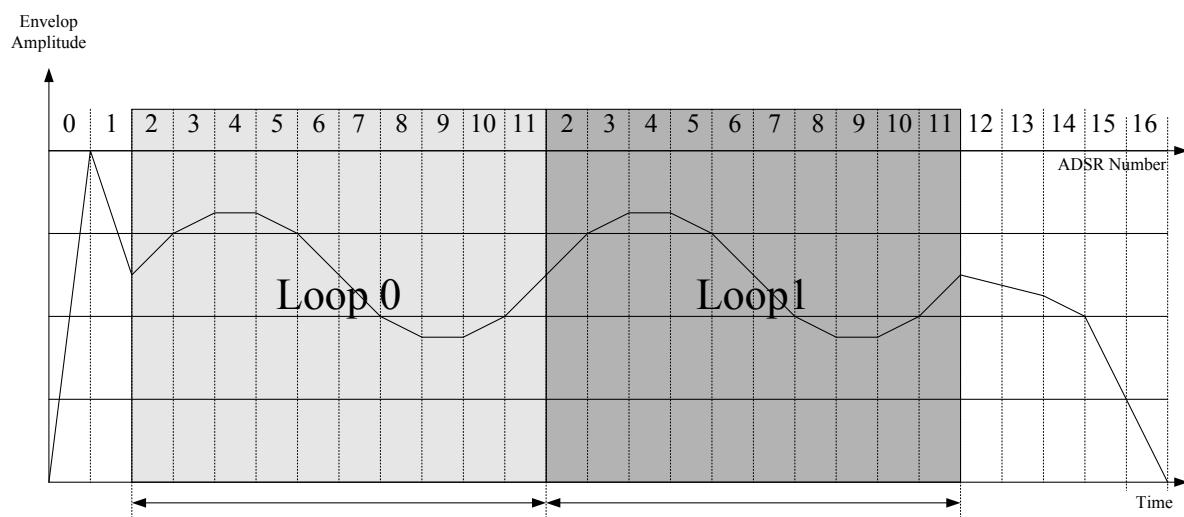
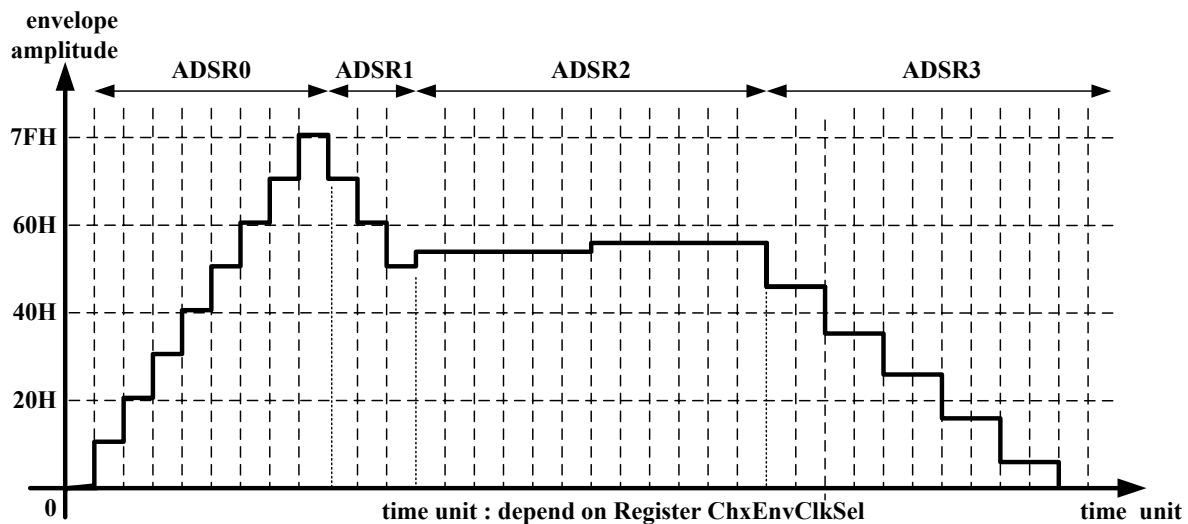
P_SPU_CtrChPitchBendEn_H 0x9300BEE0 Channel[31:16] | pitch bend enable

Bit	15	14	13	12	11	10	9	8
Function	ChPitchBendEn[31:16]							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	ChPitchBendEn[31:16]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ChPitchBendEn[31:16]	R/W	This register controls the pitch bend feature of each channel.	0: Pitch bend disabled. 1: Pitch bend enabled.

29.10.8 Envelope Control

Two envelop modes are supported, auto mode and manual mode. When manual mode is selected, programmer must update the value of envelop using the envelop IRQ. When auto mode is chosen, programmer shall configure every channel's envelop clock to determine the velocity to update envelop. After that, SPU will automatically update envelops; see the diagram below. In addition, it has a repeat function to synthesize the Low Frequency OSC (LFO) envelop easily.



The envelop clock speed is shown as follows:

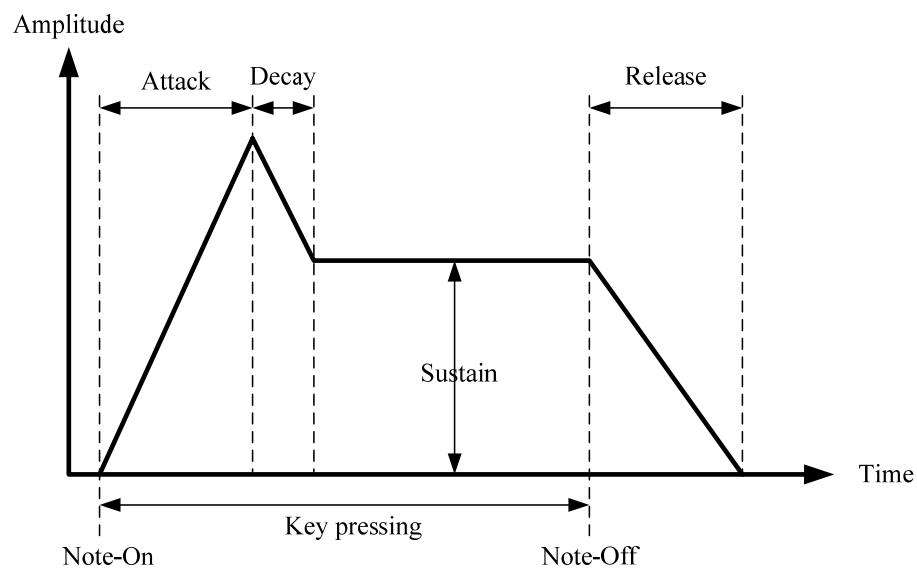
ChxEnvClk where, x is channel 0~31, frame-rate = 281.25KHz

Note: Presently, sunmidar2 supports the envelope clock of 2.2KHz.

Setting	Description	Unit
0000	EnvClk Count once = frame-rate / 4 * 4	17.5 KHz
0001	EnvClk Count once = frame-rate / 8 * 4	8.7 KHz
0010	EnvClk Count once = frame-rate / 16 * 4	4.4 KHz
0011	EnvClk Count once = frame-rate / 32 * 4	2.2 KHz
0100	EnvClk Count once = frame-rate / 64 * 4	1.1 KHz
0101	EnvClk Count once = frame-rate / 128 * 4	540Hz
0110	EnvClk Count once = frame-rate / 256 * 4	270 Hz

Setting	Description	Unit
0111	EnvClk Count once = frame-rate / 512 * 4	130 Hz
1000	EnvClk Count once = frame-rate / 1024 * 4	68 Hz
1001	EnvClk Count once = frame-rate / 2048 * 4	34 Hz
1010	EnvClk Count once = frame-rate / 4096 * 4	17 Hz
1011	EnvClk Count once = frame-rate / 8192 * 4	8 Hz
1100	EnvClk Count once = frame-rate / 8192 * 4	8 Hz
1101	EnvClk Count once = frame-rate / 8192 * 4	8 Hz
1110	EnvClk Count once = frame-rate / 8192 * 4	8 Hz
1111	EnvClk Count once = frame-rate / 8192 * 4	8 Hz

As the diagram shown below, when Note-off is received, envelop must be at release stage (ramp down).



P_SPU_CtrChEnvMode 0x9300BE54 Channel[15:0] envelope mode

Bit	15	14	13	12	11	10	9	8
Function								

Default 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
ChEnvMode[15:0]							

0 0 0 0 0 0 0 0

Bit	Function	Type	Description					Condition
[15:0]	ChEnvMode[15:0]	R/W	This register controls the envelope mode for each channel. Programmer can switch a channel from auto to manual mode any time.					0: auto mode. 1: manual mode.

P_SPU_CtrChEnvMode_H 0x9300BED4 Channel[31:16] envelope mode

Bit	15	14	13	12	11	10	9	8
Function	ChEnvMode[31:16]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChEnvMode[31:16]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	ChEnvMode[31:16]	R/W	This register is to control the envelope mode for each channel. Programmer can switch a channel from auto to manual mode any time.					0: auto mode. 1: manual mode.

P_SPU_CtrChRepeatEn 0x9300BE50 Channel[15:0] repeat enable

Bit	15	14	13	12	11	10	9	8
Function	ChRepeatEn[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChRepeatEn[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	ChRepeatEn[15:0]	R/W	This register is used to control the repeat feature of envelope data. RpEn and RpCnt in P_ChEnvelop1[0x1018+0x40*X] is useful.					0: EnvRepeat disabled. 1: EnvRepeat enabled.

P_SPU_CtrChRepeatEn_H 0x9300BED0 Channel[31:16] repeat enable

Bit	15	14	13	12	11	10	9	8
Function	ChRepeatEn[31:16]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChRepeatEn[31:16]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	ChRepeatEn[31:16]	R/W	This register controls the repeat feature of envelope data. RpEn and RpCnt in P_ChEnvelop1[0x1018+0x40*X] is useful.				0: EnvRepeat disabled. 1: EnvRepeat enabled.

P_SPU_CtrChEnvIRQSts 0x9300BE5C Channel[15:0] envelope IRQ status

Bit	15	14	13	12	11	10	9	8
Function	ChEnvIRQSts[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChEnvIRQSts[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	ChEnvIRQSts[15:0]	R/W	Envelope IRQ will be set when IrqEn is “1” AND Eaoffset matches IrqFireAddress .				0: No Envelope IRQ. 1: Envelope IRQ is set. Write “1” to clear this bit.

P_SPU_CtrChEnvIRQSts_H 0x9300BEDC Channel[31:16] envelope IRQ status

Bit	15	14	13	12	11	10	9	8
Function	ChEnvIRQSts[31:16]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ChEnvIRQSts[31:16]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	ChEnvIRQSts[31:16]	R/W	Envelope IRQ will be set when IrqEn is "1" AND Eaoffset matches IrqFireAddress .					0: No Envelope IRQ. 1: Envelope IRQ is set. Write 1 clear this bit.

P_SPU_CtrEnvClkSel0 0x9300BE18 Envelope Interval selection

Bit	15	14	13	12	11	10	9	8
Function	Ch3EnvClkSel[3:0]					Ch2EnvClkSel[3:0]		
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Ch1EnvClkSel[3:0]					Ch0EnvClkSel[3:0]		
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:12]	Ch3EnvClkSel[3:0]	R/W	Channel 3 envelope Interval selection					Please refer to Envelope clock table
[11:8]	Ch2EnvClkSel[3:0]	R/W	Channel 2 envelope Interval selection					Please refer to Envelope clock table
[7:4]	Ch1EnvClkSel[3:0]	R/W	Channel 1 envelope Interval selection					Please refer to Envelope clock table
[3:0]	Ch0EnvClkSel[3:0]	R/W	Channel 0 envelope Interval selection					Please refer to Envelope clock table

P_SPU_CtrEnvClkSel1 0x9300BE1C Envelope Interval selection

Bit	15	14	13	12	11	10	9	8
Function	Ch7EnvClkSel[3:0]					Ch6EnvClkSel[3:0]		
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Ch5EnvClkSel[3:0]				Ch4EnvClkSel[3:0]			
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	Ch7EnvClkSel[3:0]	R/W	Channel 7 envelope Interval selection				Please refer to Envelope clock table
[11:8]	Ch6EnvClkSel[3:0]	R/W	Channel 6 envelope Interval selection				Please refer to Envelope clock table
[7:4]	Ch5EnvClkSel[3:0]	R/W	Channel 5 envelope Interval selection				Please refer to Envelope clock table
[3:0]	Ch4EnvClkSel[3:0]	R/W	Channel 4 envelope Interval selection				Please refer to Envelope clock table

P_SPU_CtrEnvClkSel2
0x9300BE20
Envelope Interval selection

Bit	15	14	13	12	11	10	9	8
Function	Ch11EnvClkSel[3:0]				Ch10EnvClkSel[3:0]			
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Ch9EnvClkSel[3:0]				Ch8EnvClkSel[3:0]			
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	Ch11EnvClkSel[3:0]	R/W	Channel 11 envelope Interval selection				Please refer to Envelope clock table
[11:8]	Ch10EnvClkSel[3:0]	R/W	Channel 10 envelope Interval selection				Please refer to Envelope clock table
[7:4]	Ch9EnvClkSel[3:0]	R/W	Channel 9 envelope Interval selection				Please refer to Envelope clock table
[3:0]	Ch8EnvClkSel[3:0]	R/W	Channel 8 envelope Interval selection				Please refer to Envelope clock table

P_SPU_CtrEnvClkSel3
0x9300BE24
Envelope Interval selection

Bit	15	14	13	12	11	10	9	8
Function	Ch15EnvClkSel[3:0]				Ch14EnvClkSel[3:0]			
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Ch13EnvClkSel[3:0]				Ch12EnvClkSel[3:0]			
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	Ch15EnvClkSel[3:0]	R/W	Channel 15 envelope Interval selection				Please refer to Envelope clock table
[11:8]	Ch14EnvClkSel[3:0]	R/W	Channel 14 envelope Interval selection				Please refer to Envelope clock table
[7:4]	Ch13EnvClkSel[3:0]	R/W	Channel 13 envelope Interval selection				Please refer to Envelope clock table
[3:0]	Ch12EnvClkSel[3:0]	R/W	Channel 12 envelope Interval selection				Please refer to Envelope clock table

P_SPU_CtrEnvClkSel0_H
0x9300BE98
Envelope Interval selection

Bit	15	14	13	12	11	10	9	8
Function	Ch19EnvClkSel[3:0]				Ch18EnvClkSel[3:0]			
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Ch17EnvClkSel[3:0]				Ch16EnvClkSel[3:0]			
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	Ch19EnvClkSel[3:0]	R/W	Channel 19 envelope Interval selection				Please refer to Envelope clock table
[11:8]	Ch18EnvClkSel[3:0]	R/W	Channel 18 envelope Interval selection				Please refer to Envelope clock table
[7:4]	Ch17EnvClkSel[3:0]	R/W	Channel 17 envelope Interval selection				Please refer to Envelope clock table
[3:0]	Ch16EnvClkSel[3:0]	R/W	Channel 16 envelope Interval selection				Please refer to Envelope clock table

P_SPU_CtrEnvClkSel1_H
0x9300BE9C
Envelope Interval selection

Bit	15	14	13	12	11	10	9	8	
Function	Ch23EnvClkSel[3:0]					Ch22EnvClkSel[3:0]			
Default	0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
Ch21EnvClkSel[3:0]					Ch20EnvClkSel[3:0]		
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	Ch23EnvClkSel[3:0]	R/W	Channel 23 envelope Interval selection				Please refer to Envelope clock table
[11:8]	Ch22EnvClkSel[3:0]	R/W	Channel 22 envelope Interval selection				Please refer to Envelope clock table
[7:4]	Ch21EnvClkSel[3:0]	R/W	Channel 21 envelope Interval selection				Please refer to Envelope clock table
[3:0]	Ch20EnvClkSel[3:0]	R/W	Channel 20 envelope Interval selection				Please refer to Envelope clock table

P_SPU_CtrEnvClkSel2_H
0x9300BEAO
Envelope Interval selection

Bit	15	14	13	12	11	10	9	8	
Function	Ch27EnvClkSel[3:0]					Ch26EnvClkSel[3:0]			
Default	0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
Ch25EnvClkSel[3:0]					Ch24EnvClkSel[3:0]		
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	Ch27EnvClkSel[3:0]	R/W	Channel 27 envelope Interval selection				Please refer to Envelope clock table
[11:8]	Ch26EnvClkSel[3:0]	R/W	Channel 26 envelope Interval selection				Please refer to Envelope clock table
[7:4]	Ch25EnvClkSel[3:0]	R/W	Channel 25 envelope Interval selection				Please refer to Envelope clock table
[3:0]	Ch24EnvClkSel[3:0]	R/W	Channel 24 envelope Interval selection				Please refer to Envelope clock table

P_SPU_CtrEnvClkSel3_H 0x9300BEA4 Envelope Interval selection

Bit	15	14	13	12	11	10	9	8
Function	Ch31EnvClkSel[3:0]				Ch30EnvClkSel[3:0]			
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Ch29EnvClkSel[3:0]				Ch28EnvClkSel[3:0]			
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	Ch31EnvClkSel[3:0]	R/W	Channel 31 envelope Interval selection				Please refer to Envelope clock table
[11:8]	Ch30EnvClkSel[3:0]	R/W	Channel 30 envelope Interval selection				Please refer to Envelope clock table
[7:4]	Ch29EnvClkSel[3:0]	R/W	Channel 29 envelope Interval selection				Please refer to Envelope clock table
[3:0]	Ch28EnvClkSel[3:0]	R/W	Channel 28 envelope Interval selection				Please refer to Envelope clock table

P_SPU_CtrEnvRampDown 0x9300BE28 Ch[15:0] Envelope ramp down

Bit	15	14	13	12	11	10	9	8
Function	EnvRampDown[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
EnvRampDown[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	EnvRampDown[15:0]	R/W	When EnvRampDown is set to "1", EnvRampDownClk will be read from Phase SRAM and Ramp down offset will be read from attribute SRAM then the envelop will ramp down to 0 within 0 ~ 3 seconds, period based on the EnvRampDownClk ($0x180C+0x40*X$) and RampDownOffset ($0x100C+0x40*X$). 0: No operation 1: Start ramp down				

P_SPU_CtrEnvRampDown_H 0x9300BEA8 Ch[31:16]Envelope ramp down

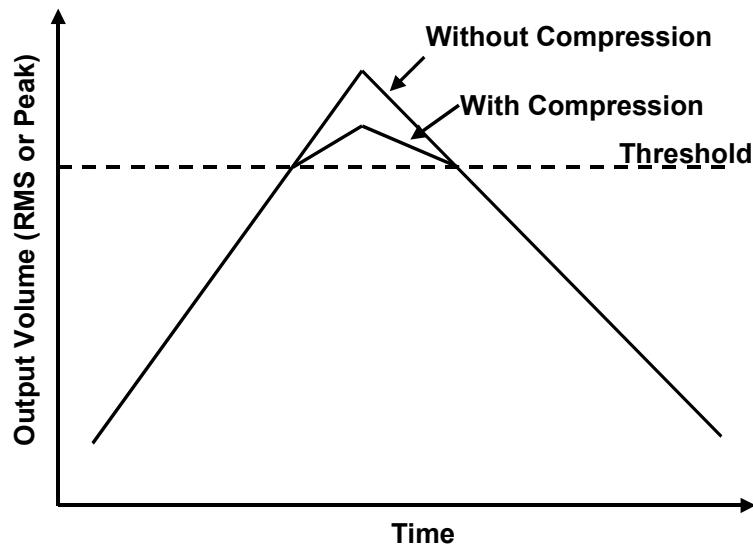
Bit	15	14	13	12	11	10	9	8
Function	EnvRampDown[31:16]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
EnvRampDown[31:16]							
0	0	0	0	0	0	0	0

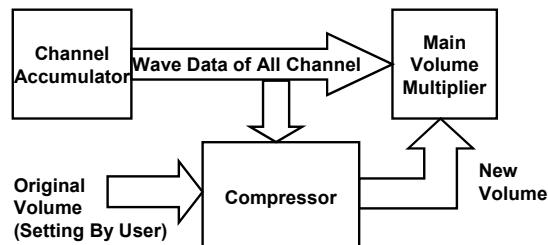
Bit	Function	Type	Description	Condition
[15:0]	EnvRampDown[31:16]	R/W	When EnvRampDown is set to "1", EnvRampDownClk will be read from Phase SRAM and Ramp down offset will be read from attribute SRAM; thus, the envelop will ramp down to 0 in 0 ~ 3 seconds period according to the configuration at EnvRampDownClk ($0x180C+0x40*X$) and RampDownOffset ($0x100C+0x40*X$).	0: No operation 1: Start ramp down

29.10.9 Compressor Control

When CompEn is "1", the internal compressor will be activated and will take the parameter stored in 0x0E38 and 0x0E64 to dynamically control the output volume. When the VolSel is set to a non-zero value, there is a possibility that the output wave will saturate; thus, the compressor can raise the overall output volume without saturation.



The compressor detects the output level **before** the main volume multiplier, and it will control the main volume dynamically. See the following diagram for more details.



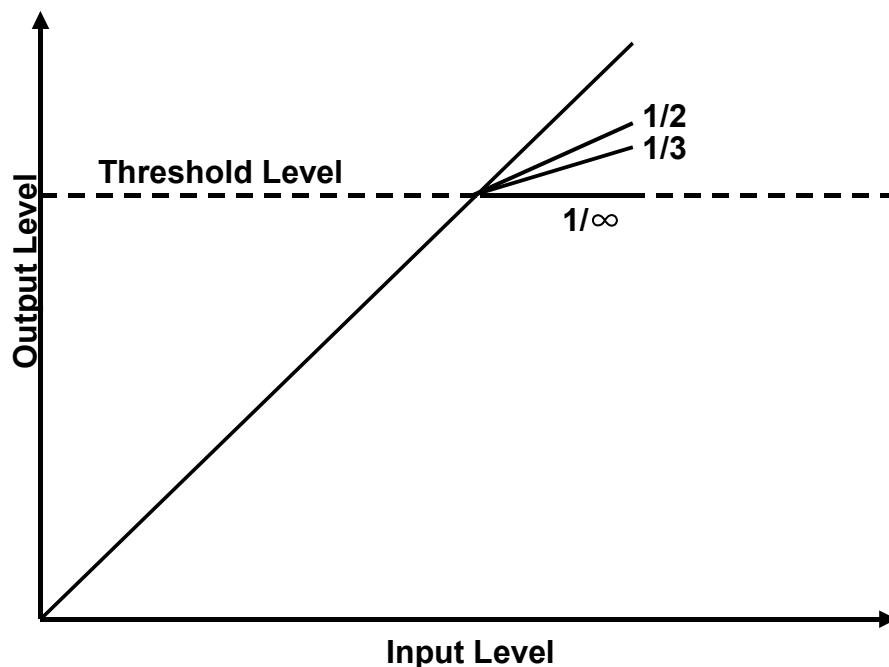
Suppose the maximum value of wave data at all channels is “1”. When peak or RMS of wave data at all channels is larger than the value, (**threshold/0x80**), the compressor will start to activate.

The definition of attack time is how fast the compressor responses to the wave data peak or RMS over the threshold. The fast attack time will result in fast response to the wave data, meaning the compress will start in a short time. This is mainly to prevent saturation, but the transient part in the wave may be lost. So, fast attack time is good for melody, but not for drum. The slow attack time will result in slow response to the wave data, meaning the compress will start after a while. This is to preserve the transient part in the wave, but it's inappropriate to avoid saturation. User can try both settings and find the best combinations for the application. Real attack time of “1” means the attack time is approx. 1.5ms. Real attack time of “100” means the attack time is approx. 150 ms... etc..

The definition of release time is how fast the compressor responses to the wave data peak or RMS lower than the threshold. The fast release time will result in the pump of breathing effect situation, but long

release time lowers the overall volume. A programmer may try both settings and find the best setting for the application. Real release time of "1" means the release time is approx. 1.5 ms. Real release time of "100" means the release time is approx. 150 ms...,etc.

Compress Ratio Setting controls how much it compresses when output-wave level is higher than the threshold level.



Since the compress ratio is highly depended on the setting of VolSel, the following table can be used for suitable ratio selection.

Recommended Threshold Level and Ratio Combination in various VolSel Setting.

VolSel	Minimum Channel Number to Cause Saturation			Effect	Suggested Threshold Level	Suggested Ratio Setting
	32-ch	16-ch	8ch			
00 (x1)	Not possible			-	Turn-off Compressor	Turn-off Compressor
01 (x2)	16	8	4	Compress	0x20 (1/4)	0 (1/2)
				Limit	0x40 (1/2)	7 (1/∞)
10 (x4)	8	4	2	Compress	0x10 (1/8)	0 (1/2)
				Limit	0x20 (1/4)	7 (1/∞)
11 (x16)	2	1	0.5	Compress	0x04 (1/32)	0 (1/2)
				Limit	0x08 (1/16)	7 (1/∞)

User who intends to use compress ratio other than 1/2 and 1/ ∞ can use the threshold level in between these two settings in order to get the best outcome. Larger compress ratio lowers the overall volume but less saturation possibility.

P_SPU_CtrDummy 0x9300BE38 Compressor control								
Bit	15	14	13	12	11	10	9 8	
Function	Peak	Threshold						
Default	0	0	0	0	0	0	0 0	
	7	6	5	4	3	2	1 0	
	AttScale		RelScale		DisZC	Ratio		
	0	0	0	0	0	0	0 0	

Bit	Function	Type	Description	Condition
15	Peak	R/W	Select RMS or Peak mode.	0: RMS mode 1: Peak mode
[14:8]	Threshold	R	The threshold's maximum value is 0x7F and 0x01 in minimum; 0x00 is not allowed.	0x01~0x7F
[7:6]	AttScale	-	Attack Time Scale Control Register	00: AttackTime[7:0] * 1 01: AttackTime[7:0] * 4 10: AttackTime[7:0] * 16 11: AttackTime[7:0] * 64
[5:4]	RelScale	-	Release Time Scale Control Register	00: ReleaseTime [7:0] * 1 01: ReleaseTime [7:0] * 4 10: ReleaseTime [7:0] * 16 11: ReleaseTime [7:0] * 64
3	DisZC	-	Disable Zero Cross Function of Compressor. As described before, the compressor will adjust the main volume automatically. This bit is used to control whether or not the volume change wait to zero cross.	0: Enable zero cross 1: Disable zero cross.

Bit	Function	Type	Description	Condition
[2:0]	Ratio	R	The ratio is used to control how much it compresses when the output-wave level is higher than the threshold level.	0: 1/2 1: 1/3 2: 1/4 3: 1/5 4: 1/6 5: 1/7 6: 1/8 7: 1/∞(Limiter)

P_SPU_CtrAttRelTime 0x9300BE68 Attack/Release Time Control

Bit	15	14	13	12	11	10	9	8
Function	AttackTime[7:0]							
Default	0	0	0	0	0	0	0	0
	ReleaseTime [7:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	AttackTime[7:0]	R/W	The definition of attack time is how fast the compressor responses to the wave data peak or RMS over the threshold.	0x00~0xFF
[7:0]	ReleaseTime [7:0]	R	The definition of release time is how fast the compressor responses to the wave data peak or RMS lower than the threshold.	0x00~0xFF

29.10.10 Post Process Control
P_SPU_CtrPW 0x9300BE94 Post Wave processing (PW) Control

Bit	15	14	13	12	11	10	9	8
Function	PWEN	PWDS	--	PWCK	PWIE	PWIS	--	
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PWSE	PWSIL			--			PWFOV
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	PWEN	R/W	Post Wave Output (to DMA) Enable	0: Disable 1: Enable
14	PWDS	R/W	Post Wave Down-sample by. 6 (valid while PWCK=1)	0: Not down-sample 1: Down-sample by 6
13	Reserved	R/W	Reserved	
12	PWCK	R/W	Post Wave Sampling Clock Setting	0: Post Wave Processing is not valid 1: Post Wave Processing in 288KHz
11	PWIE	R/W	Post Wave IRQ Enable (valid while PWCK=1). When the quantity of down sampled waves is reached at PWCnt, IRQ will launch CPU Writing "0" to the register (0x0E14) to clear the PW IRQ.	0: Disable 1: Enable
10	PWIS	R/W	Post Wave IRQ (IRQ 4) event status	0: PwIRQ inactive 1: PwIRQ active
[9:8]	Reserved	R/W	reserved	
7	PWSE	R/W	Post Wave Output in signed or unsigned format (valid while PWCK=1)	0: UNSIGNED. 1: SIGNED
6	PWSIL	R/W	Post Wave Not Output to DAC	0: Not output to DAC. 1: Output to DAC.
[5:1]	Reserved	R/W	reserved	
0	PWFOV	R/W	Internal Post Wave Output Queue to DMA overflow.	Write 1: Clear PWFOV flag Write 0: No operation Read 1: Overflow Read 0: Not Overflow

P_SPU_CtrPWaveOutL 0x9300BEC8 Wave output port for post-processing.

Bit	15	14	13	12	11	10	9	8
Function	PWaveOutL[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PWaveOutL[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	PWaveOutL[15:0]	R/W	The data is the left output port of 32-channel. The data in this register can be accessed by auto increment or manual mode (0xE94). When DMA mode is used, the address of this register must be set to 0xC0110000.					

P_SPU_CtrPWaveOutR 0x9300BECC Wave output port for post-processing.

Bit	15	14	13	12	11	10	9	8
Function	PWaveOutR[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PWaveOutR[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	PWaveOutR[15:0]	R/W	The data is the right output port of 32-channel. The data in this register can be accessed by auto increment or manual mode (0xE94). When DMA mode is used, the address of this register must be set to 0xC0110004.					

29.10.11 Frequency Vibration Control

The SPU embedded in GPL32900A supports the frequency vibration for each channel. To enable this function, the following register can be used to control the depth of vibration.

P_SPU_DVB0		0x9300BF00								DVB control register 0							
Bit	Function	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default		0	0	0	0	0	0	0	0	Ch3DVB[3:0]				Ch2DVB[3:0]			
		0	0	0	0	0	0	0	0	Ch1DVB[3:0]				Ch0DVB[3:0]			
		0	0	0	0	0	0	0	0						0	0	0

Bit	Function	Type	Description						Condition
[15:12]	Ch3DVB[3:0]	R/W	Channel 3 depth of vibration.						Please refer to DVB table
[11:8]	Ch2DVB[3:0]	R/W	Channel 2 depth of vibration.						Please refer to DVB table
[7:4]	Ch1DVB[3:0]	R/W	Channel 1 depth of vibration.						Please refer to DVB table
[3:0]	Ch0DVB[3:0]	R/W	Channel 0 depth of vibration.						Please refer to DVB table

P_SPU_DVB1		0x9300BF04								DVB control register 1							
Bit	Function	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default		0	0	0	0	0	0	0	0	Ch7DVB[3:0]				Ch6DVB[3:0]			
		0	0	0	0	0	0	0	0	Ch5DVB[3:0]				Ch4DVB[3:0]			
		0	0	0	0	0	0	0	0						0	0	0

Bit	Function	Type	Description						Condition
[15:12]	Ch7DVB[3:0]	R/W	Channel 7 depth of vibration.						Please refer to DVB table
[11:8]	Ch6DVB[3:0]	R/W	Channel 6 depth of vibration.						Please refer to DVB table
[7:4]	Ch5DVB[3:0]	R/W	Channel 5 depth of vibration.						Please refer to DVB table
[3:0]	Ch4DVB[3:0]	R/W	Channel 4 depth of vibration.						Please refer to DVB table

P_SPU_DVB2
0x9300BF08
DVB control register 2

Bit	15	14	13	12	11	10	9	8	
Function	Ch11DVB[3:0]					Ch10DVB[3:0]			
Default	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	Ch9DVB[3:0]					Ch8DVB[3:0]			
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description				Condition
[15:12]	Ch11DVB[3:0]	R/W	Channel 11 depth of vibration.				Please refer to DVB table
[11:8]	Ch10DVB[3:0]	R/W	Channel 10 depth of vibration.				Please refer to DVB table
[7:4]	Ch9DVB[3:0]	R/W	Channel 9 depth of vibration.				Please refer to DVB table
[3:0]	Ch8DVB[3:0]	R/W	Channel 8 depth of vibration.				Please refer to DVB table

P_SPU_DVB3
0x9300BF0C
DVB control register 3

Bit	15	14	13	12	11	10	9	8	
Function	Ch15DVB[3:0]					Ch14DVB[3:0]			
Default	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	Ch13DVB[3:0]					Ch12DVB[3:0]			
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description				Condition
[15:12]	Ch15DVB[3:0]	R/W	Channel 15 depth of vibration.				Please refer to DVB table
[11:8]	Ch14DVB[3:0]	R/W	Channel 14 depth of vibration.				Please refer to DVB table
[7:4]	Ch13DVB[3:0]	R/W	Channel 13 depth of vibration.				Please refer to DVB table
[3:0]	Ch12DVB[3:0]	R/W	Channel 12 depth of vibration.				Please refer to DVB table

P_SPU_DVB4
0x9300BF10
DVB control register 4

Bit	15	14	13	12	11	10	9	8	
Function	Ch19DVB[3:0]					Ch18DVB[3:0]			
Default	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	Ch17DVB[3:0]					Ch16DVB[3:0]			
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description				Condition
[15:12]	Ch19DVB[3:0]	R/W	Channel 19 depth of vibration.				Please refer to DVB table
[11:8]	Ch18DVB[3:0]	R/W	Channel 18 depth of vibration.				Please refer to DVB table
[7:4]	Ch17DVB[3:0]	R/W	Channel 17 depth of vibration.				Please refer to DVB table
[3:0]	Ch16DVB[3:0]	R/W	Channel 16 depth of vibration.				Please refer to DVB table

P_SPU_DVB5
0x9300BF14
DVB control register 5

Bit	15	14	13	12	11	10	9	8	
Function	Ch23DVB[3:0]					Ch22DVB[3:0]			
Default	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	Ch21DVB[3:0]					Ch20DVB[3:0]			
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description				Condition
[15:12]	Ch23DVB[3:0]	R/W	Channel 23 depth of vibration.				Please refer to DVB table
[11:8]	Ch22DVB[3:0]	R/W	Channel 22 depth of vibration.				Please refer to DVB table
[7:4]	Ch21DVB[3:0]	R/W	Channel 21 depth of vibration.				Please refer to DVB table
[3:0]	Ch20DVB[3:0]	R/W	Channel 20 depth of vibration.				Please refer to DVB table

P_SPU_DVB6
0x9300BF18
DVB control register 6

Bit	15	14	13	12	11	10	9	8	
Function	Ch27DVB[3:0]					Ch26DVB[3:0]			
Default	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	Ch25DVB[3:0]					Ch24DVB[3:0]			
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description				Condition
[15:12]	Ch27DVB[3:0]	R/W	Channel 27 depth of vibration.				Please refer to DVB table
[11:8]	Ch26DVB[3:0]	R/W	Channel 26 depth of vibration.				Please refer to DVB table
[7:4]	Ch25DVB[3:0]	R/W	Channel 25 depth of vibration.				Please refer to DVB table
[3:0]	Ch24DVB[3:0]	R/W	Channel 24 depth of vibration.				Please refer to DVB table

P_SPU_DVB7
0x9300BF1C
DVB control register 7

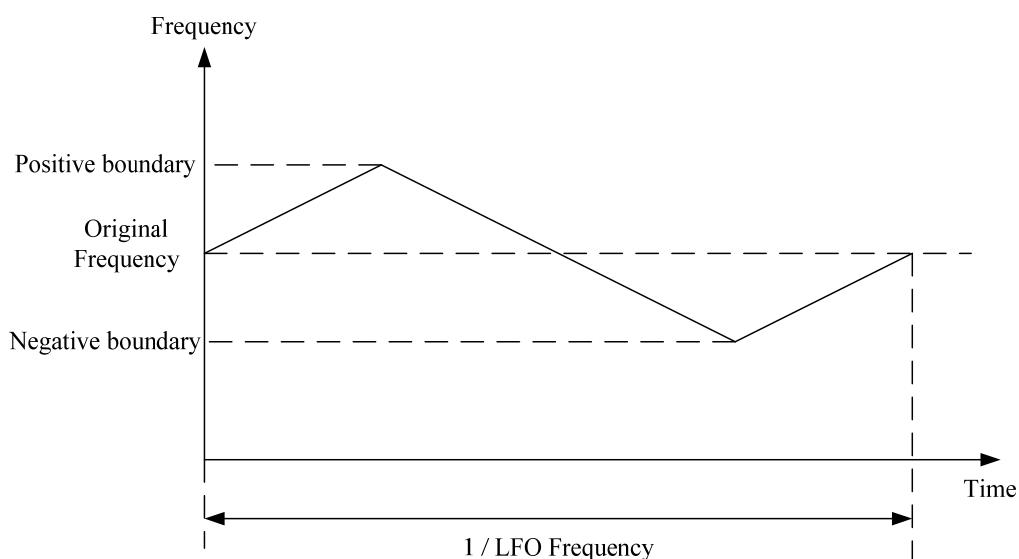
Bit	15	14	13	12	11	10	9	8	
Function	Ch31DVB[3:0]					Ch30DVB[3:0]			
Default	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	Ch29DVB[3:0]					Ch28DVB[3:0]			
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description				Condition
[15:12]	Ch31DVB[3:0]	R/W	Channel 31 depth of vibration.				Please refer to DVB table
[11:8]	Ch30DVB[3:0]	R/W	Channel 30 depth of vibration.				Please refer to DVB table
[7:4]	Ch29DVB[3:0]	R/W	Channel 29 depth of vibration.				Please refer to DVB table
[3:0]	Ch28DVB[3:0]	R/W	Channel 28 depth of vibration.				Please refer to DVB table

The following table shows the definition of ChXDVB[3:0]

ChXDVB[3:0]	Percentage %	Cents
0x0	0 (Disable DVB)	0 (Disable DVB)
0x1	+/-0.20%	3.38
0x2	+/-0.39%	6.75
0x3	+/-0.59%	10.12
0x4	+/-0.78%	13.47
0x5	+/-0.98%	16.83
0x6	+/-1.17%	20.17
0x7	+/-1.37%	23.51
0x8	+/-1.56%	26.84
0x9	+/-1.76%	30.17
0xA	+/-1.95%	33.49
0xB	+/-2.15%	36.80
0xC	+/-2.34%	40.11
0xD	+/-2.54%	43.41
0xE	+/-2.73%	46.71
0xF	+/-2.93%	50.00

The following figure shows the vibration waveform of frequency when doing frequency vibration.



The LFO vibration frequency is controlled by the setting in the phase ram, please refer to the LFO control section for detail.

29.11 Internal SRAM

Internal Attribute SRAM Format, Channel 0 ~ Channel 31

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0x1000+0x40*X	Wave Address																Waddr[15:0]
0x1004+0x40*X	Mode	ADPCM	16M	ToneMode													Waddr[21:16]
0x1008+0x40*X	Loop Address																LoopAddr[15:0]
0x100C+0x40*X	Pan							Pan[6:0]									ChVolumn[6:0]
0x1010+0x40*X	Envelope0	Repeat Period						EnvTarget[6:0]			EnvSign						EnvInc[6:0]
0x1014+0x40*X	Envelope Data						EnvCnt[7:0]										EDD[6:0]
0x1018+0x40*X	Envelope1					RpCnt		Rpt									EnvLoad[7:0]
0x101C+0x40*X	Envelope Address						IrqFireAddress[8:0]			IrqEn							Eaddr[21:16]
0x1020+0x40*X	Envelope Address								Eaddr[15:0]								
0x1024+0x40*X	Wave Data 0								WDD0[15:0]								
0x1028+0x40*X	Envelope Loop Control				RampDownoffset[6:0]								Eaoffset[8:0]				
0x102C+0x40*X	Wave Data								WDD[15:0]								
0x1030+0x40*X	-																
0x1034+0x40*X	ADPCM Sel	ADPCM36			PointNumber												
0x1038+0x40*X	WLaddr High Address							LoopAddr[27:22]									Waddr[27:22]
0x103C+0x40*X	Envelope High Address																Eaddr[27:22]



GPL32900A Programming Guide

Internal Phase SRAM Format, Channel 0 ~ Channel 31

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0x1800+0x40*X	Phase High																Phase[18:16]
0x1804+0x40*X	Phase Accumulator High																PhaseAcc[18:16]
0x1808+0x40*X	TargetPhase High																TargetPhase[18:16]
0x180C+0x40*X	RampDownClk								DAM								RampDownClk
0x1810+0x40*X	Phase																Phase[15:0]
0x1814+0x40*X	Phase Accumulator																PhaseAcc[15:0]
0x1818+0x40*X	Target Phase																Target Phase[15:0]
0x181C+0x40*X	Phase Control			PhaseTimeStep		Sign											PhaseOffset
0x1820+0x40*X	-																
0x1824+0x40*X	-																
0x1828+0x40*X	-																
0x182C+0x40*X	-																
0x1830+0x40*X	-																
0x1834+0x40*X	-																
0x1838+0x40*X	-																
0x183C+0x40*X	-																

29.11.1 Tone-Color Register

Wave data basically has four modes: 8-bitPCM, 16-bit PCM, ADPCM and ADPCM36. Only Auto-End and Auto-Repeat mode are available in ADPCM mode. When the end code (0xFFFF) is reached, this bit is cleared to zero automatically, i.e. back to PCM mode. It means only wave data in the first region will be treated as ADPCM data and all others are assumed as PCM data.

Four ways to play tone-color

00: Software mode (S/W): Tone-Color S/W mode. In S/W mode, the tone color value is the value in WDD(0x1024 + 0x40*X) that is written by CPU.

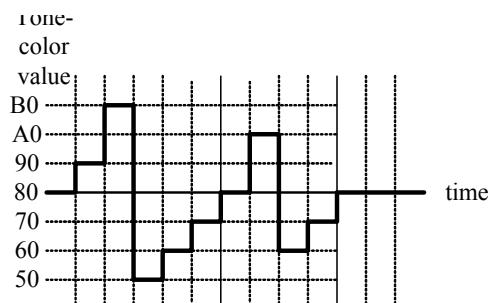
01: Hardware (H/W) auto-end mode: Tone-Color H/W output with auto end. SPU auto loads tone-color and stops playing when tone-color hits the end-code ("0xFF" for 8-bit tone color data mode and "0xFFFF" for 16-bit tone color data mode).

10: Hardware (H/W) auto-repeat mode. Tone-Color H/W output with auto repeat. SPU reloads tone-color automatically when end-code is reached. The data between loop address and end-code is repeated till the channel is off. If ADPCM36 Mode is selected, it means the data in loop area is ADPCM format.

11: Hardware (H/W) auto-repeat mode 1: Tone-Color H/W output with auto repeat. SPU reloads tone-color automatically when end-code is reached. The data between loop address and end-code is repeated till the channel is off. When ADPCM36 Mode is used, it means the data in loop area is PCM format. The mode of PCM (8 or 16) is selected at the ToneColor16 bit.

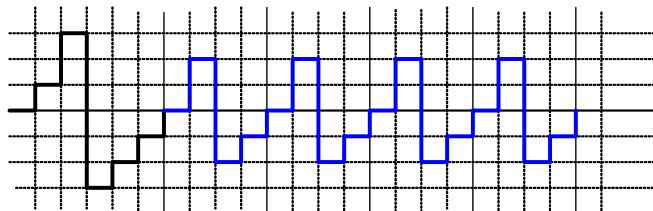
Example: H/W auto-end mode

Address	Data
WAaddr + 0	0x9080
WAaddr + 1	0x50B0
WAaddr + 2	0x7060
WAaddr + 3	0xA080
WAaddr + 4	0x7060
WAaddr + 5	0xFFFF



Example: H/W auto-repeat mode

Address	Data
WAddr + 0	0x9080
WAddr + 1	0x50B0
WAddr + 2	0x7060
WAddr + 3	0xA080
WAddr + 4	0x7060
WAddr + 5	0xFFFF



LoopAddr = WAddr + 3 ;

In ADPCM36 mode, when the ToneMode is set to 10b, the data format is in the following diagram:

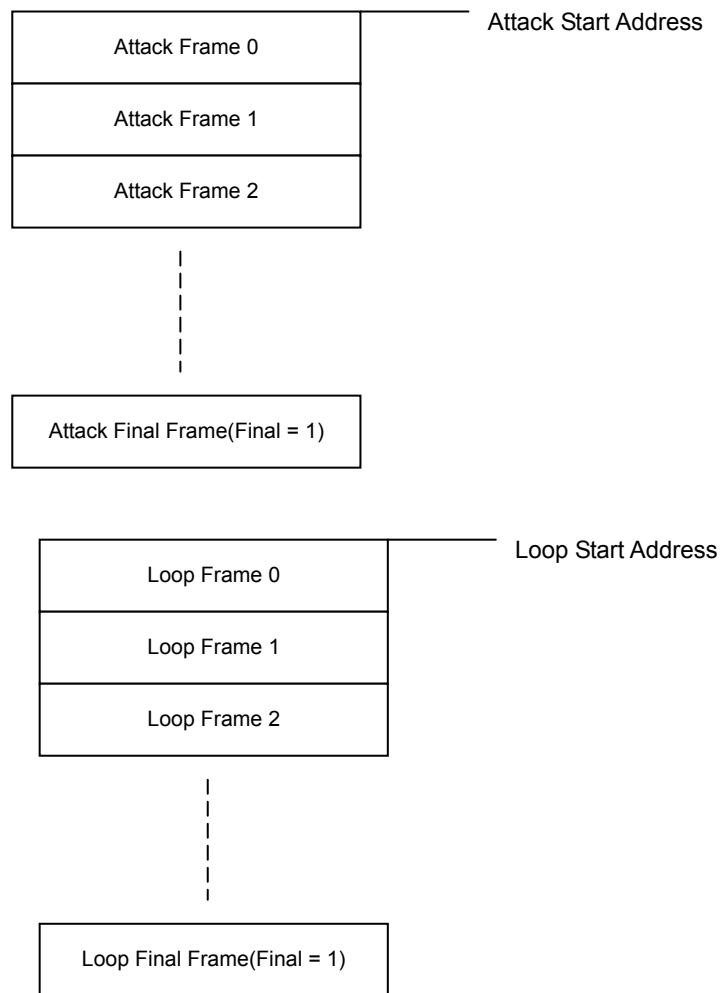
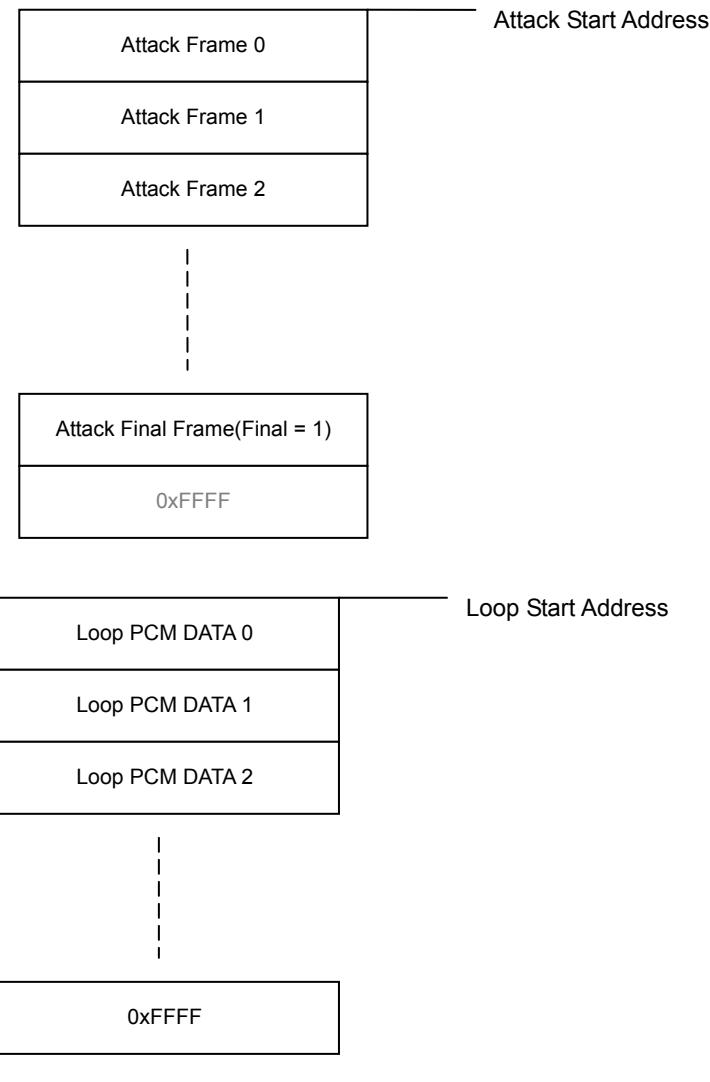


Figure. ToneMode = 10b, ADPCM36 => ADPCM36 mode

In ADPCM36 mode, an end code (0xFFFF) must be appended at the end of attack data when the ToneMode is set to 11b. The data format is given in the following diagram:



ToneMode = 11b, ADPCM36 => PCM mode

P_SPU_ChWaveAddr 0x9300C000+0x40*X Wave (Tone color) data start address

Bit	15	14	13	12	11	10	9	8
Function	Waddr[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
Waddr[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description		Condition
[15:0]	Waddr[15:0]	R/W	Combining P_ChMode with P_ChWaveAddr to get the wave data(tone-color) start address (Waddr[27:0]).		

P_SPU_ChMode 0x9300C004+0x40*X Wave (Tone color) data start address

Bit	15	14	13	12	11	10	9	8
Function	ADPCM	ToneColor16	ToneMode[1:0]				LoopAddr[21:16]	
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0	
LoopAddr[21:16]		Waddr[21:16]						
0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
15	ADPCM	R/W	ADPCM mode on/off	0: PCM data. 1: ADPCM data.
14	ToneColor16		Tone color data resolution control. In 8-bit tone color data mode, wave data is accessed byte by byte. In 16-bit tone color data mode, data is accessed by word.	0: 8-bit data mode 1: 16-bit data mode
[13:12]	ToneMode[1:0]		Tone-color mode control	00: S/W mode. 01: H/W auto-end mode. 10:H/W auto-repeat mode 11: H/W auto-repeat mode
[11:6]	LoopAddr[21:16]		LoopAddr[27:0]; Wave data loop address	
[5:0]	Waddr[21:16]		Waddr[27:0]; Wave (Tone color) data start address	

P_SPU_ChLoopAddr 0x9300C008+0x40*X Wave data loop address

Bit	15	14	13	12	11	10	9	8
Function	LoopAddr[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
LoopAddr[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	LoopAddr[15:0]	R/W	Combining P_ChMode with P_ChLoopAddr to get the Wave data loop address. (LoopAddr[27:0]). This address is used for the tone-color automatic repeat mode.					

P_SPU_ChWLAddrH 0x9300C038+0x40*X Wave/Loop start address Hi

Bit	15	14	13	12	11	10	9	8
Function						LoopAddr[27:22]		
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
LoopAddr[27:22]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[11:6]	LoopAddr[27:22]		LoopAddr[27:0]; Wave data loop address					
[5:0]	Waddr[27:22]		Waddr[27:0]; Wave (Tone color) data start address					

P_SPU_ChWaveData0 0x9300C024+0x40*X Previous Tone-color data value (wave data)

Bit	15	14	13	12	11	10	9	8
Function	WDD1[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
WDD1[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	WDD1[15:0]	R/W	Programmer must write 0x8000 to this register in ADPCM36 mode; otherwise, this register can be ignored.					

P_SPU_ChWaveData1 0x9300C02C+0x40*X Tone-color data value (wave data)

Bit	15	14	13	12	11	10	9	8
Function	WDD[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
WDD[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	WDD[15:0]	R/W	In tone color automatic mode (0x1004+0x40*X), WDD is fetched automatically by hardware. In software mode, WDD is written by CPU. When 8-bit tone color mode (0x1004+0x40*X) is set, the wave data should be placed at WDD[15:8] . And in 16-bit tone color mode, the tone color data is located in WDD[15:0].					

P_SPU_ChAdpcmSel 0x9300C034+0x40*X ADPCM Mode Select

Bit	15	14	13	12	11	10	9	8
Function	ADPCM36	PointNum[4:0]						
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
--							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15]	ADPCM36	R/W	When ADPCM bit in 0x1004+0x40*X is set to "1", configure this bit for ADPCM or ADPCM36.	0: ADPCM 1: ADPCM36
[14:9]	PointNum[4:0]	R/W	The point numbers of first frame in ADPCM36 mode, This field must be filled with the correct value before play an ADPCM36 tone colors. This field must be filled with 0 when the tonecolor is not in the ADPCM36 mode.	0: 1 point. 1: 2 point. 2: 3 point. 3: 4 point. 31 32 point.
[8:0]	reserve	R/W		

29.11.2 Volume Register

When Pan < 64

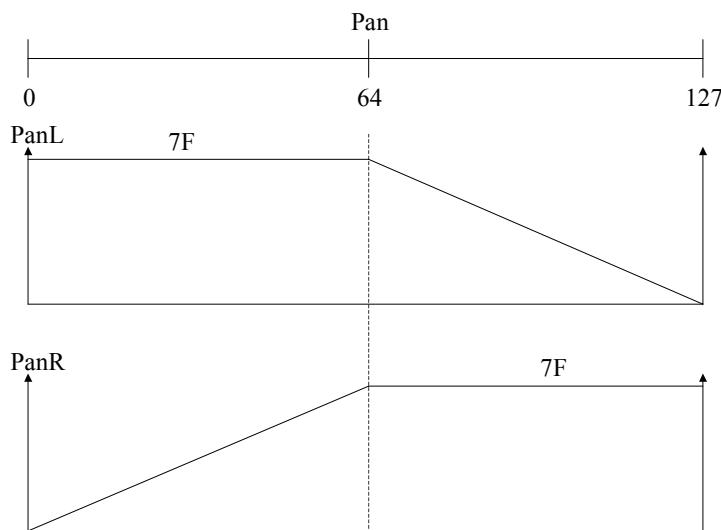
$$\text{PanL} = 0x7F * \text{ChVolume}$$

$$\text{PanR} = \text{Pan} * 2 * \text{ChVolume}$$

When Pan >= 64

$$\text{PanL} = (127 - \text{Pan}) * 2 * \text{ChVolume}$$

$$\text{PanR} = 0x7F * \text{ChVolume}$$



P_SPU_ChPan **0x9300C00C+0x40*X** Volume Setting for right and left channel.

Bit	15	14	13	12	11	10	9	8
Function	--							ChPan[6:0]
Default	0	0	0	0	0	0	0	0

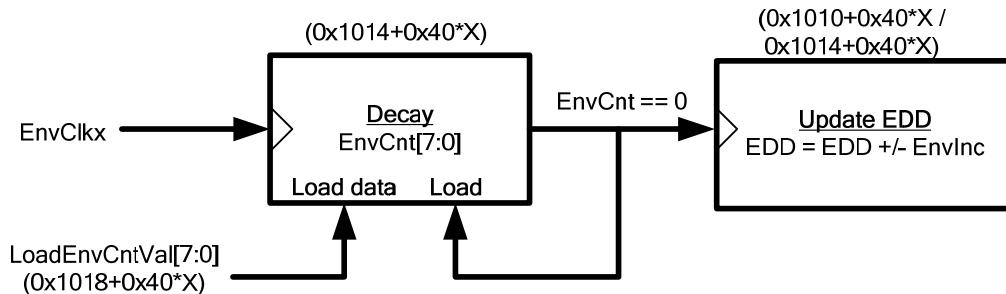
7	6	5	4	3	2	1	0
--							ChVelocity[6:0]
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15]	-	-	reserved	
[14:8]	ChPan[6:0]	R/W	Ratio control for right and left channel.	00~7F
[7]	-	-	reserved	
[6:0]	ChVelocity[6:0]	R/W	Velocity control for right and left channel.	00~7F

29.11.3 Envelope Register

The $0x1010+0x40*X$, $0x1014+0x40*X$ and $0x1018+0x40*X$ are the three registers that control the envelope. In Envelope automatic mode ($0x1004+0x40*X$), these registers are automatically reloaded or updated by hardware from memory. In Envelope manual mode, CPU has to update the envelope data (EDD in $0x1014+0x40*X$). To control the envelope in automatic mode, the envelope is controlled by Envelope0 ($0x1010+0x40*X$), and Envelope1($0x1018+0x40*X$) that is stored in memory.

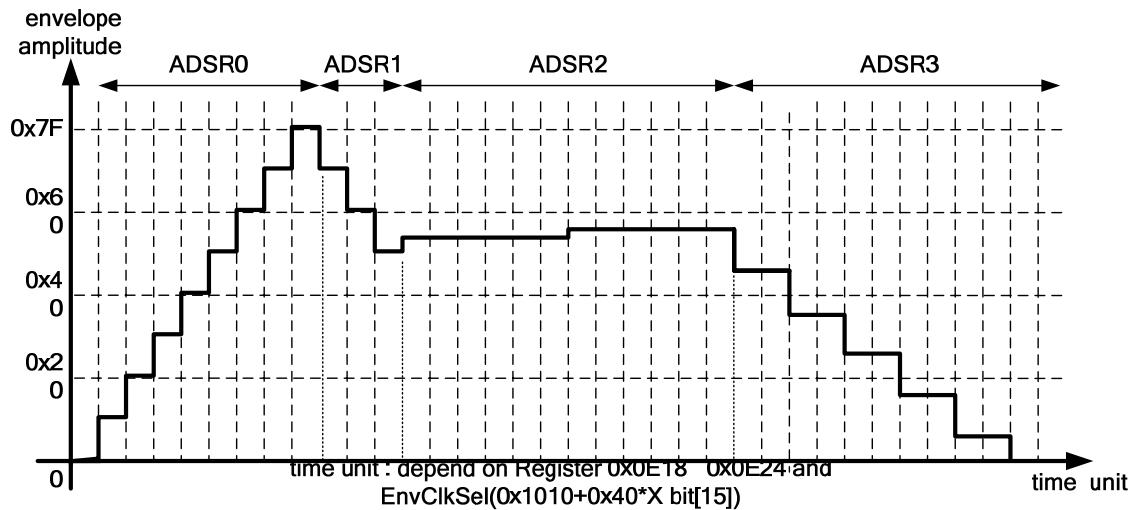
Initially, sound processor auto-loads EnvLoad[7:0] to internal register, **EnvCnt [7:0]**. The processor, according to the EnvClkx, decays EnvCnt. When EnvCnt is decayed to “0”, the processor will calculate the next EDD (envelop) value, $EDD = EDD +/- \text{EnvInc}[6:0]$ based on **EnvSign(0/1)** bit. If EDD value reaches EnvTargetValue[6:0], the processor automatically updates the data in envelope 0/1(Port $0x1010+0x40*X / 0x1018+0x40*X$).



Note : EnvClkx refer $0xE18\sim0xE24$ and $0xE98\sim0xEA4$ register description

- Example for Envelop Format without Envelop Repeat

N _{th} ADSR	Memory Offset[8:0] address	Envelope0/1 ($0x1010+0x40*X / 0x1018+0x40*X$)
0	0	Envelope 0
	1	Envelope 1
1	2	Envelope 0
	3	Envelope 1
2	4	Envelope 0
	5	Envelope 1
...



Suppose the waveform of envelop is shown as above. The start address of the envelope is as follows:

N_{th} ADSR	Memory Offset[8:0] address	Envelop0/1 (0x1010+0x40*X / 0x1018+0x40*X)
0	0	0x7F10
	1	0x0000
1	2	0x5090
	3	0x0000
2	4	0x5804
	5	0x0005
3	6	0x0090
	7	0x0001

1. In ADSR0, it is increased from 0x00 to 0x7F by 8X units. Therefore, the following values are obtained: EnvTargetValue=0x7F, EnvSign=0, EnvInc=(0x7F-0) /8=0x10, EnvCnt=0 (EDD counts each time unit).
2. In ADSR1, it is decreased from 0x7F to 0x50 by 3X units. Therefore, the following values are obtained: EnvTargetValue=0x50, EnvSign=1, EnvInc=(0x7F-0x50)/3=0x10, EnvCnt=0 (EDD counts each time unit).
3. In ADSR2, it is increased from 0x50 to 0x58 by 12X units. Therefore, the following values are obtained: EnvTargetValue=0x58, EnvSign=0, EnvInc=(0x58-0x50)/2=0x04, EnvCnt=5 (EDD counts each 6X units).
4. In ADSR3, it is decreased from 0x58 to 0x00 by 12X units. Therefore, EnvTargetValue=0x00, EnvSign=1, EnvInc=(0x58-0x00)/6=0x10, EnvCnt=1(EDD counts each 2X units).

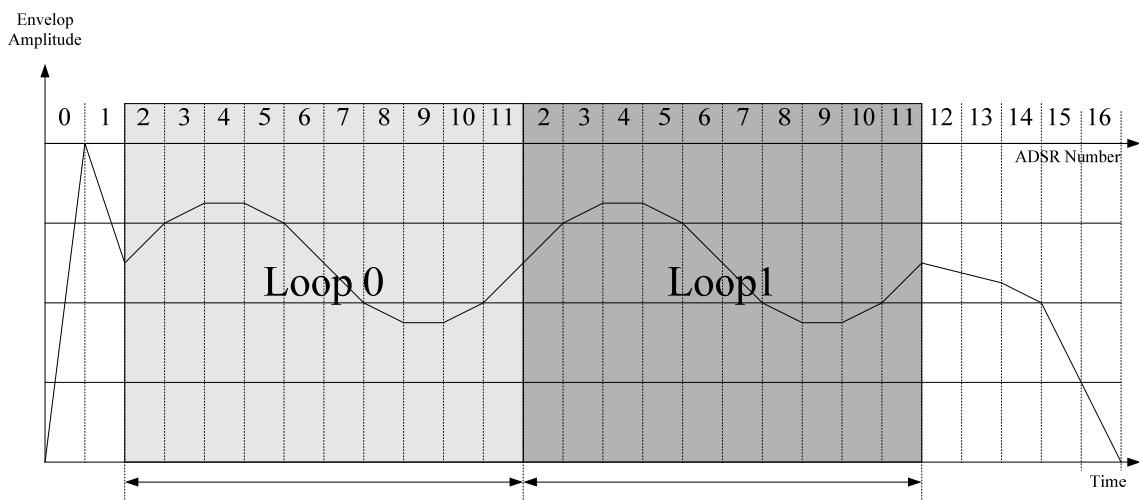
Note: For time unit definition, refer to **Envelope Interval selection** (0x0E18 ~ 0x0E24 & 0x0E98 ~ 0x0EA4).

- Example for Envelop Format with Envelope Repeat

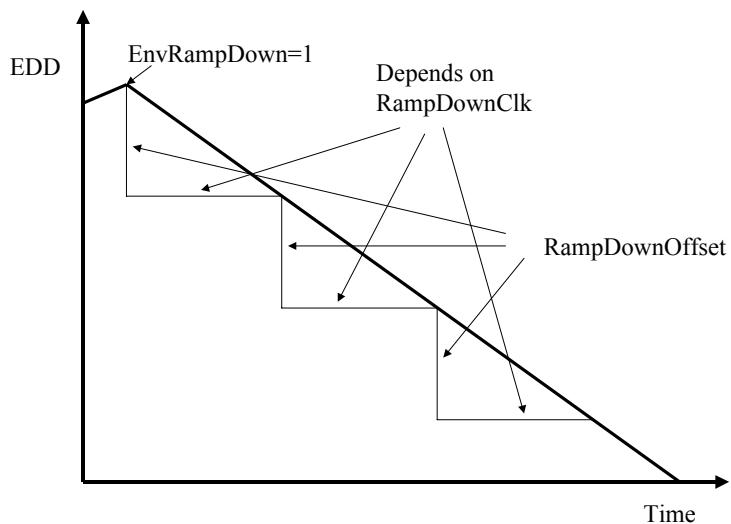
N_{th} ADSR	Memory Offset[8:0] address	Envelope0/1 (0x1010+0x40*X /0x1018+0x40*X)
0	0	Envelope 0
	1	Envelope 1
1	2	Envelope 0
	3	Envelope 1
2	4	Envelope 0
	5	Envelope 1
3	6	Envelope 0
	7	Envelope 1
4	8	Envelope 0
	9	Envelope 1
5	A	Envelope 0
	B	Envelope 1
6	C	Envelope 0
	D	Envelope 1
7	E	Envelope 0
	F	Envelope 1
8	10	Envelope 0
	11	Envelope 1
9	12	Envelope 0
	13	Envelope 1
10	14	Envelope 0
	15	Envelope 1
11	16	Envelope 0
	17	Envelope 1 (*1)
	18	Envelope Loop Control (*2)
12	19	Envelope 0
	1A	Envelope 1
...

*1: Repeat = 1.

*2: In this example, EAoffset[8:0] = 0x04, RpCnt[6:0]=0x01.



When P_CtrEnvRampDown of a channel is set to "1", **EnvRampDownClk** will be read from P_ChRampDownClk, and **Rampdownoffset** will be read from P_ChEnvelopeLoop. After that, envelop will ramp down to 0 in 0 ~ 3 seconds, period based on the **EnvRampDownClk** and **RampDownOffset**.



The decreasing value of EDD at each ramp down step.

EnvRampDownClk	Envelope ramp down step	Envelope ramp down step
000	13 * 4*4 frame	0.738 ms
001	13 * 16*4 frame	2.955 ms
010	13 * 64*4 frame	11.821 ms
011	13 * 256*4 frame	47.284 ms
100	13 * 1024*4 frame	189.137 ms
101	13 * 4096*4 frame	756.548 ms
110	13 * 8192*4 frame	1.513 s
111	13 * 8192*4 frame	1.513 s

P_SPU_ChEnvelop0 0x9300C010+0x40*X Envelope 0 for envelope tracking

Bit	15	14	13	12	11	10	9	8	
Function	--	EnvTargetValue [6:0]							
Default	0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0
Envsign	EnvInc [6:0]						
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15]	Reserved	R/W	reserved	
[14:8]	EnvTargetValue [6:0]	R/W	Envelop Target Value for envelop tracking. When envelop data EDD reaches EnvTargetValue [6:0], processor automatically update the envelope data (0x1010+0x40*X /0x1018+0x40*X) from memory.	
[7]	Envsign	R/W	envelop sign bit	0: envelop increment direction is positive. 1: envelop increment direction is negative.
[6:0]	EnvInc [6:0]	R/W	Envelop increment value for Envelop Tracking. EDD = EDD +/- EnvInc[6:0], based on EnvSign(0/1) bit.	

P_SPU_ChEnvelop1 0x9300C018+0x40*X Envelope 1 for envelope tracking

Bit	15	14	13	12	11	10	9	8
Function	RpCnt[6:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
--	EnvLoad[7:0]						
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:9]	RpCnt[6:0]	R/W	Envelop repeat counter	00~7F
[8]	Repeat	R/W	Envelope repeat control. In repeat mode, an extra word (EnvelopLoop Control in (0x1028+0x40*X)) is accessed from the external memory. The envelope is repeated between the Eaoffset and repeat point RpCnt times.	0 : Envelop normal mode 1 : Envelop repeat mode
[7:0]	EnvLoad[7:0]	R/W	EnvLoad is the value loaded into EnvCnt (0x1014+0x40*X). After EnvCnt reaches "0", the EnvLoad is re-loaded automatically.	

P_SPU_ChEnvelopData 0x9300C014+0x40*X Envelope Data

Bit	15	14	13	12	11	10	9	8	
Function	EnvCnt[7:0]								
Default	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	--	EDD [6:0]							
	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:8]	EnvCnt[7:0]	R/W	Envelop Count is controlled by hardware in Envelope auto mode. The clock source for envelop counter is from EnvClkx. It records the period for counting envelopes data (EDD) once.	
[7]	-	-	reserved	
[6:0]	EDD [6:0]	R/W	If envelope data reaches "0", the corresponding channel will be stopped automatically.	

P_SPU_ChEnvelopSeg 0x9300C01C+0x40*X Envelope Irq Fire Address

Bit	15	14	13	12	11	10	9	8
Function	IRQFireAddress[8:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
IRQFireAddress[8:0]	IRQEn	Eaddr[21:16]					
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]	IRQFireAddress[8:0]	R/W	When EAoffset matches the IrqFireAddress and IrqEn is set to "1", the envelope IRQ will be established and EnvIrqSts[x] will be set.	
[6]	IRQEn	R/W	Envelope IRQ enable bit.	0: EnvelopeIRQ disable 1: EnvelopeIRQ enable
[5:0]	Eaddr[21:16]	R/W	Envelope Address	

P_SPU_ChEnvelopOffset 0x9300C020+0x40*X Envelope Base Address

Bit	15	14	13	12	11	10	9	8
Function	Eaddr[15:0]							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	Eaddr[15:0]	R/W	Envelope Address	

P_SPU_ChEAddrH 0x9300C03C+0x40*X Envelope start address Hi

Bit	15	14	13	12	11	10	9	8
Function								
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[5:0]	Waddr[27:22]		Eaddr[27:0]; Envelope start address				

P_SPU_ChEnvelopLoop 0x9300C028+0x40*X Envelope Loop Control

Bit	15	14	13	12	11	10	9	8
Function	RampDownOffset[6:0]							EAoffset[8:0]
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	EAoffset[8:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:9]	RampDownOffset[6:0]	R/W	Envelop ramp down offset; this value decreases the EDD when EnvRampDown of a channel is set.				
[15:0]	EAoffset[8:0]	R/W	Envelop repeat address offset.				

P_SPU_ChRampDownClk 0x9300C80C+0x40*X EnvRampDownClk

Bit	15	14	13	12	11	10	9	8
Function	--							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	--					EnvRampDownClk		
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:3]	reserve	R/W					
[2:0]	EnvRampDownClk	R/W	Envelope Ramp Down Clock Selection Control. This register is used to control the ramp down period when EnvRampDown[x] is set to "1".				000: 0.738 ms 001: 2.955 ms 010: 11.821 ms 011: 47.284 ms 100: 189.137 ms 101: 756.548 ms 110: 1.513 s 111: 1.513 s

29.11.4 Phase Register

Use the following relation to get the phase of a sample rate.

$$\text{Phase} = \text{sample-rate} * 2^{19} / 281.25 \text{ kHz}$$

P_SPU_ChPhaseHi 0x9300C800+0x40*X control tone-color pitch (similar to sound pitch)

Bit	15	14	13	12	11	10	9	8
Function	--							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	--					Phase[18:16]		
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]	reserve	R/W		
[2:0]	Phase[18:16]	R/W	Phase = sample-rate * 2 ¹⁹ / 281.25 kHz Combining P_ChPhaseHi with P_ChPhase to get the tone-color pitch Phase [18:0]	

P_SPU_ChPhase 0x9300C810+0x40*X control tone-color pitch (similar to sound pitch)

Bit	15	14	13	12	11	10	9	8
Function	Phase[15:0]							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	Phase[15:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	Phase[15:0]	R/W	Phase = sample-rate * 2 ¹⁹ / 281.25 kHz Combining P_ChPhaseHi with P_ChPhase to get the tone-color pitch Phase [18:0]	

P_SPU_ChPhaseAccumHi 0x9300C804+0x40*X Tone-color pitch accumulator

Bit	15	14	13	12	11	10	9	8
Function	--							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
--							PhaseAcc[18:16]
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]	reserve	R/W		
[2:0]	PhaseAcc[18:16]	R/W	Combine P_ChPhaseAccumHi with P_ChPhaseAccum to get the tone-color pitch accumulator PhaseAcc[18:0]. To have the lowest latency between the channel enable(ChEn[x], 0x0E00) and the play of the first tone color, it's recommended setting the PhaseAcc[18:0] to 0x7FFF.	

P_SPU_ChPhaseAccum 0x9300C814+0x40*X Tone-color pitch accumulator

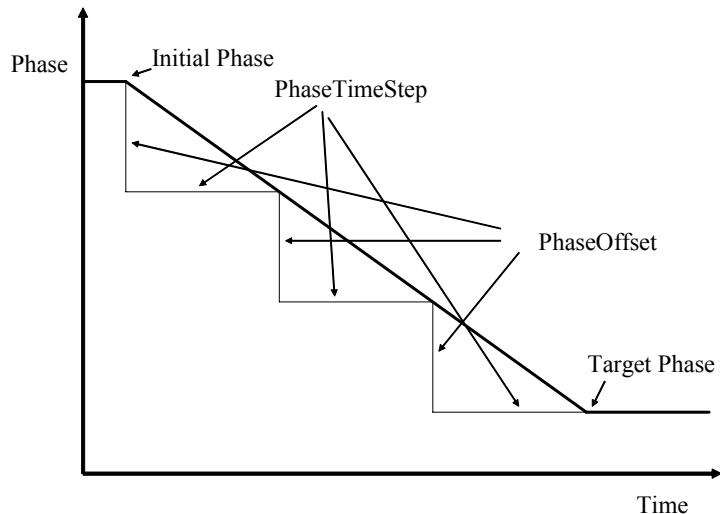
Bit	15	14	13	12	11	10	9	8
Function	PhaseAcc[15:0]							
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PhaseAcc[15:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	PhaseAcc[15:0]	R/W	Combine P_ChPhaseAccumHi with P_ChPhaseAccum to get the tone-color pitch accumulator PhaseAcc[18:0]. To have the lowest latency between the channel enable(ChEn[x], 0x0E00) and the play of the first tone color, it's recommended setting the PhaseAcc[18:0] to 0x7FFF.	

29.11.5 PitchBend Register

If PitchBendEn[x] is “1”, the channel will use PhaseOffset, PhaseSign, and PhaseTimeStep to increase/decrease Phase to the TargetPhase.



P_SPU_ChPhaseControl 0x9300C81C+0x40*X Pitch Bend Control

Bit	15	14	13	12	11	10	9	8
Function	PhaseTimeStep[2:0]		Sign		PhaseOffset[11:0]			
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PhaseOffset[11:0]							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:13]	PhaseTimeStep[2:0]	R/W	This register is used to control the period of phase change	000: 0.114ms 001: 0.227ms 010: 0.455ms 011: 0.909ms 100: 1.819ms 101: 3.637ms 110: 7.274ms 111: 14.549ms

Bit	Function	Type	Description	Condition
12	PhaseSign	-	This is the increase or decrease selection of phase.	0 : Increase phase. 1 : Decrease phase.
[11:0]	PhaseOffset[11:0]	R/W	This is the phase offset each time PhaseTimeStep is reached.	

P_SPU_ChTargetPhaseHi 0x9300C808+0x40*X Pitch Bend Target Phase

Bit	15	14	13	12	11	10	9	8
Function	--							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	--							
	0	0	0	0	0	0	0	0

TargetPhase[18:16]

Bit	Function	Type	Description	Condition
[15:3]	reserve	R/W		
[2:0]	TargetPhase[18:16]	R/W	The pitch bend target phase. Combine P_ChTargetPhaseHi with P_ChTargetPhase to get the pitch bend target phase TargetPhase[18:0]	

P_SPU_ChTargetPhase 0x9300C818+0x40*X Pitch Bend Target Phase

Bit	15	14	13	12	11	10	9	8
Function	TargetPhase[15:0]							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	TargetPhase[15:0]							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TargetPhase[15:0]	R/W	The pitch bend target phase. Combine P_ChTargetPhaseHi with P_ChTargetPhase to get the pitch bend target phase TargetPhase[18:0]	

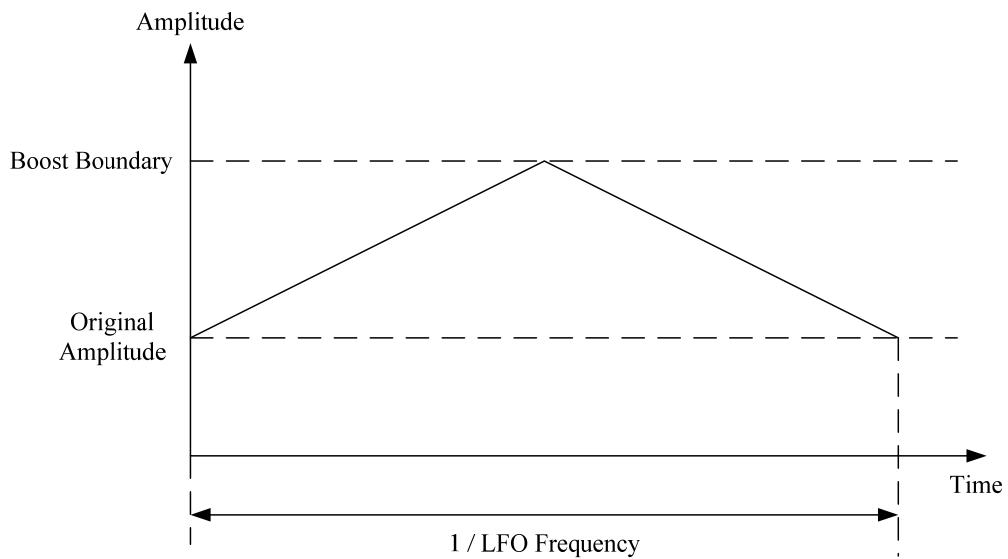
29.11.6DAM Register

The DAM register is used to control the depth of amplitude modulation.

The following table shows the definition of the DAM register.

DAM[4:0]	Percentage(%)	Boost dB	DAM[4:0]	Percentage(%)	Boost dB
0x00	0 (no DAM)	0 (no DAM)	0x10	100.00%	6.02
0x01	6.25%	0.53	0x11	106.25%	6.29
0x02	12.50%	1.02	0x12	112.50%	6.55
0x03	18.75%	1.49	0x13	118.75%	6.80
0x04	25.00%	1.94	0x14	125.00%	7.04
0x05	31.25%	2.36	0x15	131.25%	7.28
0x06	37.50%	2.77	0x16	137.50%	7.51
0x07	43.75%	3.15	0x17	143.75%	7.74
0x08	50.00%	3.52	0x18	150.00%	7.96
0x09	56.25%	3.88	0x19	156.25%	8.17
0x0A	62.50%	4.22	0x1A	162.50%	8.38
0x0B	68.75%	4.54	0x1B	168.75%	8.59
0x0C	75.00%	4.86	0x1C	175.00%	8.79
0x0D	81.25%	5.17	0x1D	181.25%	8.98
0x0E	87.50%	5.46	0x1E	187.5%	9.17
0x0F	93.75%	5.74	0x1F	194%	9.36

The following figure shows the amplitude vibration waveform when doing amplitude vibration.



The LFO vibration frequency is controlled by the setting in the phase ram, please refer to the LFO control section for detail.

29.11.7LFO Register

The LFO register is used to control the LFO frequency for both amplitude modulation and frequency modulation. The following registers are used to control the LFO frequency.

P_SPU_ChPhaseHi 0x9300C800+0x40*X control tone-color pitch (similar to sound pitch)

Bit	15	14	13	12	11	10	9	8
Function	--				LFO Counter 1[9:5]			
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
LFO Counter 1[4:0]					Phase[18:16]		
0	0	0	0	0	0	0	0

P_SPU_ChPhaseAccumHi 0x9300C804+0x40*X Tone-color pitch accumulator

Bit	15	14	13	12	11	10	9	8
Function	--				LFO Counter 2[9:5]			
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
LFO Counter 2[4:0]					PhaseAcc[18:16]		
0	0	0	0	0	0	0	0

P_SPU_ChTargetPhaseHi 0x9300C808+0x40*X Pitch Bend Target Phase

Bit	15	14	13	12	11	10	9	8
Function	--				LFO Setting[9:5]			
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
LFO Setting[4:0]					TargetPhase[18:16]		
0	0	0	0	0	0	0	0

Both LFO Counter 1 and LFO Counter 2 and the internal counter register, and they should be clear to zero when initial a SPU channel with DVB or DAM function. The LFO frequency is controlled by the LFO setting register with the following equation.

$$\text{LFO Frequency} = 281.25\text{kHz} / (2048 * (\text{LFO Setting} + 1))$$

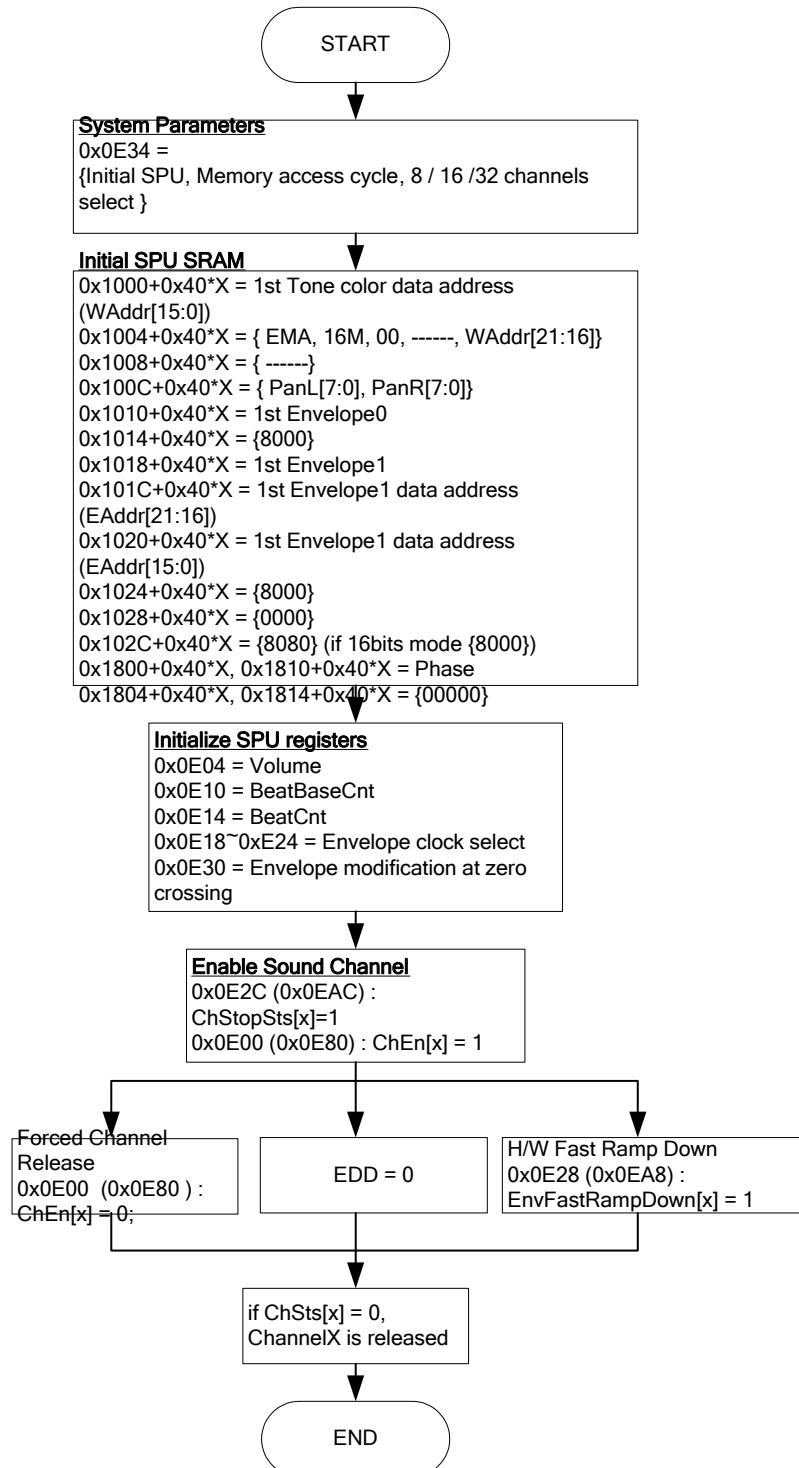
For example, when LFO setting is 1, then the LFO frequency will be $281.25\text{kHz}/(2048*2) = 68.66\text{Hz}$.

29.12 Programming Tips

- While initializing SPU SRAM, data must be written into Attribute SRAM and Phase SRAM. The Port[0x1014+0x40*X] and Port[0x1028+0x40*X] should be set to zero. For 8-bit tone color applications, Port[0x1024+0x40*X]=0x8000 (middle level), and Port[0x102C+0x40*X]=0x8080 or the first tone-color. For 16-bit tone color applications, Port[0x1024+0x40*X] is equal to 0x8000 (middle level), and Port[0x102C+0x40*X] is 0x8000 or the first tone-color. Besides, “1” should be written into Port[0xE34].b3 to set up the related SRAM initial value.

- Note: When updating data during beat event, make sure the corresponding bit is “0” in 0xE3C (ChSts). **Otherwise, hardware control is unpredictable.**
There are four ways to disable ChSts[x]:
 1. Disable ChEn[x] and the channel will be disabled in the next tone-color zero-cross.
 2. When the envelop data EDD(0x1014+0x40*X) is 0.
 3. S/W writes EnvRampDown[x]=1. H/W will ramp down in speed defined in EmvRampDownClk[2:0] and offset in EnvRampDownOffset[6:0] until envelope is equal to zero.
 4. When tone-color reaches to the end code (0xFF) in the H/W auto-end mode.

29.13 Programming Flow Example



30 SAR ADC (ADC)

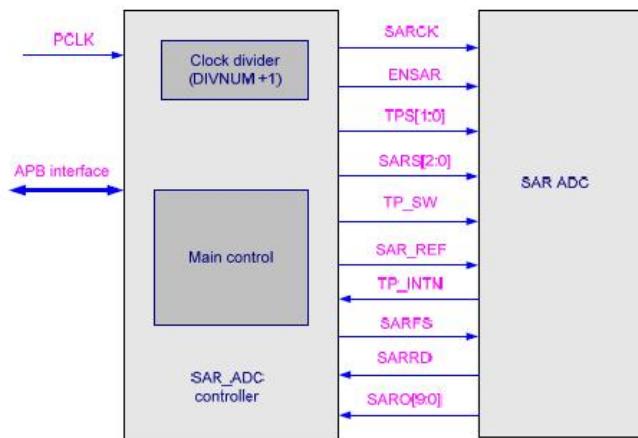
30.1 Introduction

Fifth channels of 10-bit SAR ADC are built-in GPL32900A. Four channels: TPXP, TPXN, TPYP, TPYN for touch panel input or normal ADC input, and another one channels: AUX1 for normal ADC input. These fifth channels are very suitable for system voltage detection and other general-purpose usages.

30.2 Features

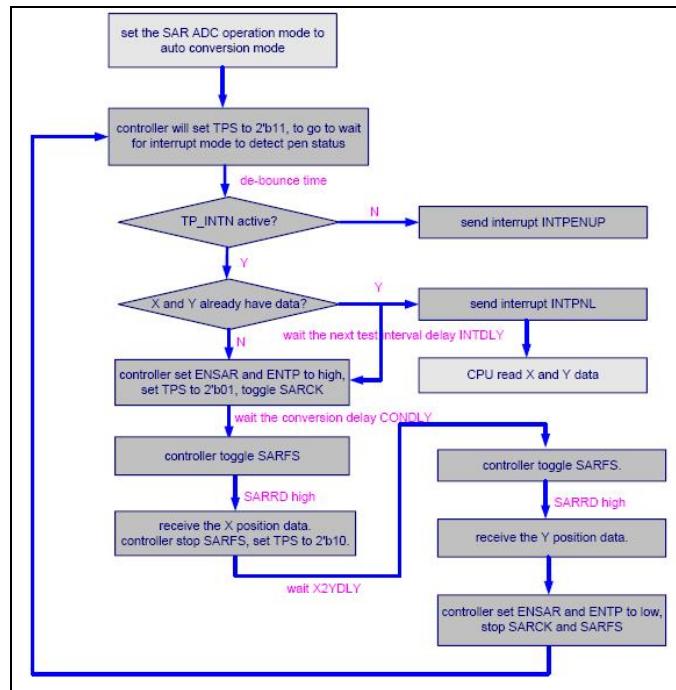
- Supports touch panel data sampling by auto measurement mode or manual setting mode.
- Supports AUX1 ADC data sampling by manual setting mode.
- Supports low power mode for SAR ADC.
- Supports pen down wake up interrupt.

30.3 Block Diagram

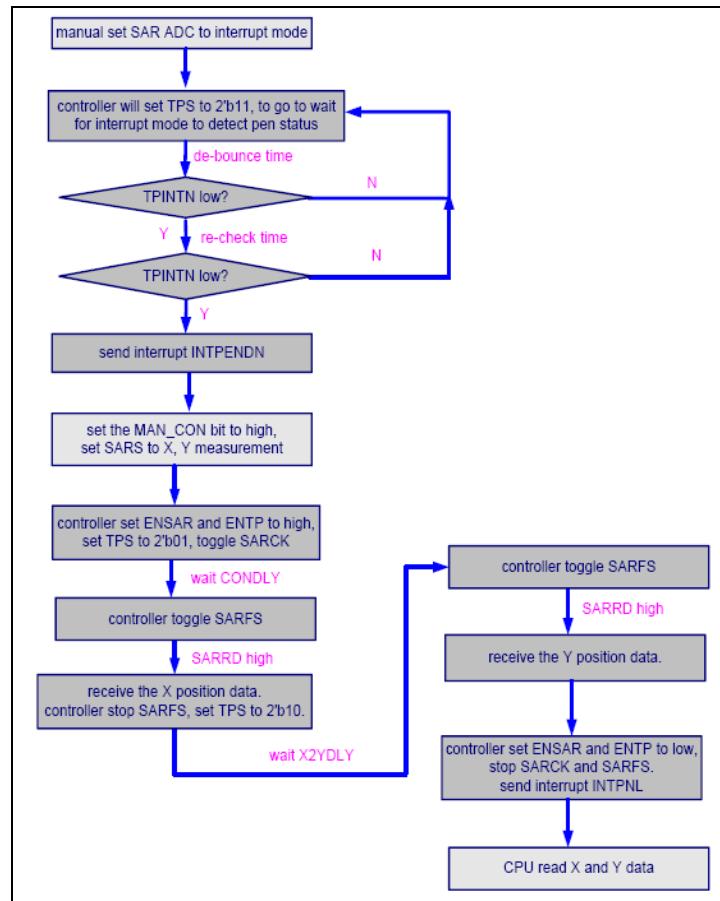


30.4 SAR ADC Measurement Flow

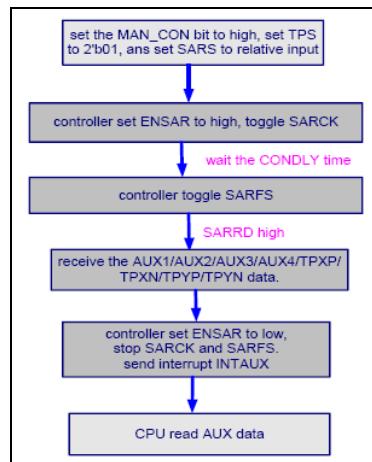
Auto Conversion mode for touch panel



Manual mode for touch panel



Manual conversion mode for AUX



30.5 Register Summary

Name	Address	Description
P_ADC_CTRL	0x9301F000	SAR ADC control register.
P_ADC_CONVDLY	0x9301F004	SAR ADC conversion delay value.
P_ADC_AUTODLY	0x9301F008	Auto conversion mode time interval setting.
P_ADC_DEBTIME	0x9301F00C	Pen down interrupt de-bounce time.
P_ADC_TP_POS	0x9301F010	Indicate the X, Y position on touch panel.
P_ADC_AUX_DATA	0x9301F014	Indicate the AUX measurement data on the touch panel.
P_ADC_INT_EN	0x9301F018	SAR ADC controller's interrupt enable.
P_ADC_INT_STATUS	0x9301F01C	SAR ADC controller's interrupt flag.

30.6 Register Definition

P_ADC_CTRL 0x9301F000 SAR control register																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	BUSY	MODE		REFVOL	TOGEN	
Default	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																
Function	DIVNUM								BACK	TPS		INSEL			MANU	AUTO
Default	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:22]	Reserve	-	-	-
21	BUSY	R	Low active SAR ADC conversion flag	0: in conversion process 1: conversion finish
[20:18]	MODE	R	Current SAR ADC operation mode	000: no operation 001: wait for interrupt 010: measure X-position 011: measure Y-position 100: measure AUX data, refer bit [4:2] to decide which input is measured.
17	REFVOL	R/W	Audio ADC reference voltage select	0: internal 1: external

Bit	Function	Type	Description	Condition
16	TOGEN	R/W	Select when to toggle SAR clock to SAR ADC	0: only toggle when SAR ADC measurement is on, this can save unnecessary power. 1: always toggling, even it is in interrupt or no operation mode
[15:8]	DIVNUM	R/W	SAR ADC clock divider number. SARCK = PCLK / (2 * (DIVNUM + 1)). SARCK must be larger than 384KHz, and lower than 2MHz.	Reset value: 0xFF
7	BACK	R/W	The mode after manual conversion end. It is invalid when AUTO (bit0) is high.	0: back to no operation mode after manual conversion. 1: back to interrupt mode after manual conversion.
[6:5]	TPS	R/W	Touch panel control interface.	00: normal test ADC 01: touch panel X-axis measurement 10: touch panel Y-axis measurement 11: interrupt mode. It is auto set in auto conversion mode.
[4:2]	INSEL	R/W	SAR ADC input selection	000: TPXP 001: TPXN 010: TPYP 011: TPYN 100: AUX1
1	MANU	R/W	High active manual measurement enable bit. When the conversion is end, this bit will be cleared. When both MANU (bit1) and AUTO (bit0) are high, controller will add a manual conversion, according to TPS and INSEL setting, and then back to auto conversion mode.	0: Disable 1: Enable
0	AUTO	R/W	High active auto conversion enable bit. SAR ADC will auto start measuring touch panel X and Y position after their data has been read.	0: Disable 1: Enable

P_ADC_CONVDLY

0x9301F004

SAR ADC conversion delay value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	CONVDLY							
Default	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]	Reserve			
[7:0]	CONVDLY	R/W	SAR ADC conversion delay value, measured with SARCK cycle. For both the auto conversion mode and manual conversion mode. A single measurement time for manual conversion mode is: CONDLY + measurement time.	Reset value = 0x10

P_ADC_AUTODYL

0x9301F008

Auto conversion mode time interval setting

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	INTDLY			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	INTDLY												X2YDLY				
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	

Bit	Function	Type	Description	Condition
[31:19]	Reserve			
[18:5]	INTDLY	R/W	In auto conversion mode, it indicates the time interval from one X and Y measurement end to a new measurement beginning. Measured with SARCK cycle.	Reset value = 0x800 (about 1ms)
[7:0]	CONVDLY	R/W	In auto conversion mode, it indicates the time interval from X-measurement end to Y-measurement frame sync beginning. Measured with SARCK cycle.	Reset value = 0x10 (a frame time)

P_ADC_DEBTIME

0x9301F00C

SAR ADC conversion delay value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Function	-	-	-	-	-	-	-	-	CHKDLY								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DEBDLY															
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:24]	Reserve			
[23:16]	CHKDLY		The pen status re-checks time value, measured with SARCK cycle. It only activates in manual conversion mode. In manual mode and SAR ADC is in interrupt mode, if the TP_INTN low level has been checked at the DEBDLY time, controller will re-check the TP_INTN in CHKDLY time, if TP_INTN is always low in the CHKDLY time, pen is down; if TP_INTN has high pulse in CHKDLY time, pen is up.	Reset value = 0x0
[15:0]	DEBDLY	R/W	SAR ADC debounce delay value, measured with SARCK cycle, each X/Y measurement should ensure the pen is down. In auto conversion mode, controller will check pen status automatically, when SAR ADC is in interrupt mode, when the delay comes, if TP_INTN is still high, it indicates pen is up; if TP_INTN is low, it indicates pen is down, the X/Y data is valid. In the manual conversion mode, if user set SAR ADC to interrupt mode, controller will check the pen status after every DEBDLY time.	Reset value = 0x1000 (2ms)

P_ADC_TP_POS

0x9301F010

SAR ADC conversion delay value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	TPPOSY															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TPPOSX															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description												Condition
[31:16]	TPPOSY	R	Touch panel Y position												Reset value: 0x0
[15:0]	TPPOSX	R	Touch panel X position												Reset value: 0x0

P_ADC_AUX_DATA
0x9301F014
SAR ADC conversion delay value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	AUX															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description												Condition
[31:16]	Reserve														
[15:0]	AUX	R	AUX measurement data, it can be read from TPXP, TPXN, TPYP, TPYN, AUX1 data.												Reset value: 0x0

P_ADC_INT_EN
0x9301F018
SAR ADC controller's interrupt enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	INTAUX	INXPNL	INTPENUP	INTPENDN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:4]	Reserve			
3	INTAUX	R/W	If the touch panel AUX register updated, the controller will insert an interrupt. When AUX data is read, auto-clear the interrupt.	0: disable 1: enable
2	INTPNL	R/W	If the touch panel X position and Y position register updated, the controller will insert an interrupt. In auto conversion mode, the INTPNL also indicates the X and Y data is valid (pen down). When PNL data is read, auto-clear the interrupt.	0: disable 1: enable
1	INTPENUP	R/W	In auto conversion mode, if controller checks TPINTN low when the DEBDLY time comes, pen up interrupt will be sent. The received touch panel data is not valid, so INTPNL will not be high. When the interrupt is read, auto-clear the interrupt.	0: disable 1: enable
0	INTPENDN	R/W	In manual mode, if the touch panel's pen down status has been checked, the controller sends this interrupt. In auto conversion mode, if the touch panel's pen down status has been checked, this interrupt will NOT be sent. Instead, the INTPNL will be sent, which indicates the received X and Y position data is valid. When pen down data is read, auto-clear the interrupt.	0: disable 1: enable

P_ADC_INT_STATUS **0x9301F01C** **SAR ADC controller's interrupt status**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	INTAUX	INXPUL	INTPENUP	INTPENDN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:4]	Reserve			
3	INTAUX	R/W	The interrupt status of INTAUX	Reset value: 0x0
2	INTPNL	R/W	The interrupt status of INTPNL	Reset value: 0x0
1	INTPENUP	R/W	The interrupt status of INTPENUP	Reset value: 0x0
0	INTPENDN	R/W	The interrupt status of INTPENDN	Reset value: 0x0

31 I2S

31.1 Introduction

The bus has only to handle audio data, while the other signals, such as sub-coding and control, are transferred separately. To minimize the number of pins required and to keep wiring simple, a 3-line serial bus is used consisting of a line for two time-multiplexed data channels, a word select line and a clock line. Since the transmitter and receiver have the same clock signal for data transmission, the transmitter as the master, has to generate the bit clock, word-select signal and data. In complex systems however, there may be several transmitters and receivers, which makes it difficult to define the master. In such systems, there is usually a system master controlling digital audio data-flow between the various ICs. Transmitters then, have to generate data under the control of an external clock, and so act as a slave. Figure 1 illustrates some simple system configurations and the basic interface timing. Note that the system master can be combined with a transmitter or receiver, and it may be enabled or disabled under software control or by pin programming.

31.2 Features

- Full-duplex data transfer between CODEC and this interface
- TX FIFO is supported for I2S controller, and FIFO size is configurable
- Support configurable settings of I2S controller for different frame sizes, word length, frame synchronization mode, data alignment, MSB/LSB first send mode, rising/falling sending edge mode, frame polarity, and the polarity of first transmitted frame.

31.3 Block Diagram

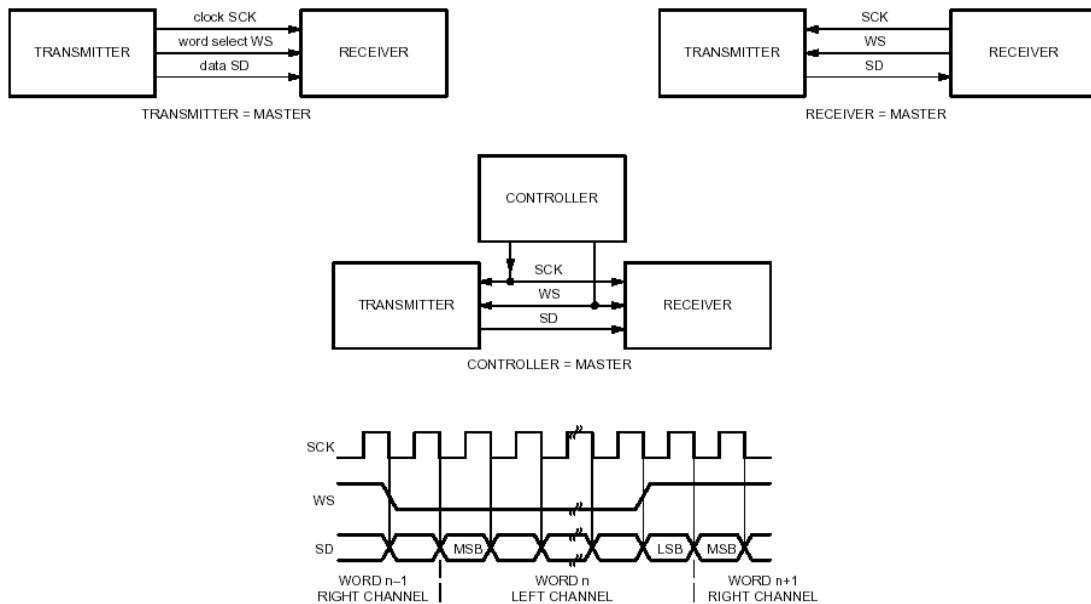


Fig. 1 Simple System Configurations and Basic Interface Timing

■ The I2S Bus

As shown in Fig. 1, the bus has three lines:

- continuous serial clock (SCK);
- word select (WS);
- serial data (SD);

and the device generating SCK and WS is the master.

- Serial Data

Serial data is transmitted in two's complement with the MSB first. The MSB is transmitted first because the transmitter and receiver may have different word lengths. It isn't necessary for the transmitter to know how many bits the receiver can handle, nor does the receiver need to know how many bits are being transmitted.

When the system word length is greater than the transmitter word length, the word is truncated (least significant data bits are set to '0') for data transmission. If the receiver is sent more bits than its word length, the bits after the LSB are ignored. On the other hand, if the receiver is sent fewer bits than its word length, the missing bits are set to zero internally. And so, the MSB has a fixed position, whereas the position of the LSB depends on the word length. The transmitter always sends the MSB of the next word one clock period after the WS changes.

Serial data sent by the transmitter may be synchronized with either the trailing (HIGH-to-LOW) or the leading (LOW-to-HIGH) edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and so there are some restrictions when transmitting data that is synchronized with the leading edge.

Word Select

The word select line indicates the channel being transmitted:

- WS = 0; channel 1 (left);
- WS = 1; channel 2 (right).

WS may change either on a trailing or leading edge of the serial clock, but it doesn't need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The WS line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clear the input for the next word.

31.4 I2S Tx

- Overwrite: When TX FIFO is full and APB master still sends data to it, "overwrite" will be asserted. At the same time, the earliest data will be discarded. This function can be enable/disable by "EN_OVWR_n" of ISCR register.
- Interrupt: When data stored in TX FIFO is less than half of its size, "interrupt" will be active. The polarity of interrupt can be configured by "IRT_Polarity" of ISCR register. User should be careful enough to distinguish the output signal "interrupt" and the interrupt flag in ISCR register, which is high active. The function can be enable/disable by "EN_IRT_n" of ISCR register.
- TX FIFO: TX FIFO is a 2-port FIFO and can read and write data at the same time. The maximum word number this FIFO can store is configurable as 2, 4, 8, or 16 (in i2s_apb_param.vh).

31.5 Control Register

Name	Address	Description
P_I2S_Ctrl	0x93012000	I2S control register
P_I2S_Data	0x93012004	I2S data register
P_I2S_Status	0x93012008	I2S status register

P_I2S_Ctrl
0x93012000
I2S control register

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
-	Mono Mode	R_LSB	MERGE	UndFlow	ClrFIFO	IRT_FLAG	EN_I2T
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
EN_OVWR_n	IRT_Polarity	SLVMode	I2SMode		FrameSizeMode	ValidDataMode	
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ValidDataMode	NormalModeAlign	SendMode	EdgeMode	FramePolarity	FirstFrameLR	EN_I2S_TX	
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]	-	-	Reserved	-
22	Mono Mode	R/W	Select channel data entry to TX Register (TX_Fifo)	0:one channel data (L channel or R channel) 1:two channel data (L channel and R channel)
21	R_LSB	R/W	Right channel data in LSB of TX_Data or not, only valid when word length is 16-bit wide, and the MERGE function is on.	0:TX_Data[15:0]:left channel data, TX_Data[31:16]:right channel data. 1:TX_Data[15:0]:right channel data, TX_Data[31:16]:left channel data.

Bit	Function	Type	Description	Condition
20	MERGE	R/W	<p>The 32-bit data which the APB bus transferred is made up of two 16-bit data or not, only valid when IIS word length is 16-bit wide.</p> <p>When word length is not 16-bit wide, the LSB of the TX_Data is the transferred data.</p> <p>0: when word length is 16-bit wide, only TX_Data[15:0] is the transferred data, TX_Data[31:16] is reserved.</p> <p>1: when word length is 16-bit wide, both TX_Data[15:0] and TX_Data[31:16] are transferred data, whether it is the right or left channel data is depended on signal R_LSB bit.</p> <p>If the APB bus transferred is made up of four 8-bit data or not , only valid when IIS word length is 8-bit wide,</p> <p>0: when word length is 8-bit wide, only TX_Data[7:0] is the transferred data , TX_DATA[31:8] is reserved .</p> <p>1: when word length is 8-bit wide , TX_Data[31:24] , TX_Data[23:16] , TX_Data[15:8] , TX_Data[7:0] are Transferred data .</p>	
19	UndFlow	R/WC	High active I2S TX FIFO underflow bit, write 1 clear.	
18	ClrFIFO	R/W	High active clear Tx FIFO bit, automatically clear to 0 after the FIFO pointer is cleared.	
17	IRT_FLAG	R/WC	High active I2S-bus Tx interrupt flag bit. When FIFO is half empty, it will be high. If FIFO's data count is larger than half count, it will be auto cleared.	
16	EN_IRT	R/W	I2S-bus Tx interrupt enable bit	0:Disable 1:Enable

Bit	Function	Type	Description	Condition
15	EN_OVWR_n	R/W	Low active Tx FIFO overwrite enable bit	0:Enable 1:Disable
14	IRT_Polarity	R/W	I2S-bus Tx interrupt polarity	0: interrupt is low active 1: interrupt is high active
13	SLVMode	R/W	Master/slave mode select signal	0:master,drive LRCK_OUT 1: slave, receive LRCK_IN
[12:11]	I2SMode	R/W	Framing Mode	00:I2S mode 01: Normal Mode , not supported in slave mode. 10:DSP mode 11:DSP mode
[10:9]	FrameSize Mode	R/W	Frame size	00:Each frame (right/left) is 16 bits length 01:Each frame (right/left) is 24 bits length 10:Each frame (right/left) is 32 bits length 11: Each frame (right/left) size is not predictable, only used in slave mode.
[8:6]	ValidData Mode	R/W	Transmitted data wordlength	000: Transmitted data wordlength is 16 bits length 001: Transmitted data wordlength is 18 bits length 010: Transmitted data wordlength is 20 bits length 011: Transmitted data wordlength is 22 bits length 100: Transmitted data wordlength is 24 bits length 101: Transmitted data wordlength is 32 bits length 110: Transmitted data wordlength is 8 bits length 111: Transmitted data wordlength is 32 bits length

Bit	Function	Type	Description	Condition
5	NormalMode Align	R/W	Transmitted data right/left alignment selection	0:Right align 1:Left align
4	SendMode	R/W	Bit priority sending mode	0:MSB sending first (Default) 1:LSB sending first
3	EdgeMode	R/W	Sending edge mode	0:Serial clock falling edge (Default) 1:Serial clock rising edge
2	FramePolarity	R/W	Frame polarity selection	0:LRCK=0 is right frame (Default) 1:LRCK=0 is left frame
1	FirstFrameLR	R/W	First transmitted frame polarity	0:Left frame (Default) 1:Right frame
0	EN_I2S_TX	R/W	EN_I2S_TX is high active enable signal for transmitting data. It should be asserted to transmit after setting all the required configurations.	0:Disable 1:Enable

P_I2S_Data
0x93012004
I2S data register

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	TX_Data															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TX_Data															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:0]	TX_Data	R	The data that will be transmitted				

P_I2S_Status
0x93012008
I2S status register

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	word_no
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0	
word_no						Overwrite	Full	Empty
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:9]	-	-	Reserved.	
[8:3]	word_no	R	The number of word stored in the Tx FIFO NOTE : The MSB of word_no depends on the size of FIFO, which is configurable	
2	Overwrite	R	Overwrite is HIGH when FIFO is overwrite	
1	Full	R	Full is HIGH to indicate Tx FIFO full	
0	Empty	R	Empty is HIGH to indicate Tx FIFO empty	

32 Audio ADC

32.1 Introduction

GPL32900A Audio ADC has two kinds of input pins: MIC, LINE-IN. Since GPL32900A provides two A/D multiplexing channels but only one physical A/D converter, user can select MIC input or LINE-IN input. And user must set I2S to receive ADC output and save it to RAM.

32.2 Register Summary

Name	Address	Description
P_AUD_PWCTL	0x9301F020	Audio ADC and DAC setting register
P_AUD_ADCCTL	0x9301F024	Audio ADC control register
P_AUD_LININCTL	0x9301F028	Audio ADC line in control register

32.3 Register Definition

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PGAG				BOOST	ENMIC	ENMICBIAS	VREFSM	ENVREF	ENZCD	
Default	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:11]	Reserve			
[10:6]	PGAG	R/W	To adjust the gain of P.G.A	0x00: 33dB 0x01: 31.5dB 0x02: 30dB 0x03: 28.5dB ... 0x16: 0dB ... 0x1D: -10.5dB 0x1E: -12dB 0x1F: -∞dB

Bit	Function	Type	Description	Condition
5	BOOST	R/W	Boost amplifier gain control.	0: 0dB 1: 20dB
4	ENMIC	R/W	MIC power control, high active	0: disable 1: enable
3	ENMICBIAS	R/W	MIC bias-voltage output power control, high active	0: disable 1: enable
2	VREFSM	R/W	VREF fast setup mode, high active	0: disable 1: enable
1	ENVREF	R/W	VREF power control, high active	0: disable 1: enable
0	ENZCD	R/W	Gain is update while signal crossing zero, high active	0: disable 1: enable

P_AUD_ADCCTL

0x9301F028

Audio ADC line in control register

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	ADCVOL	ADROVN	ADROVP	ADLOVN	ADLOVP	ADOVRS	ADHP	ENADR	ENADL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	

Bit	Function	Type	Description	Condition
[31:10]	Reserve			
9	ADCVOL	R/W	ADC volume control enable, high active	0: disable 1: enable
8	ADROVN	RO	When ADROVN is high, it means right ADC input is over ADC's negative input range	
7	ADROVP	RO	When ADROVP is high, it means right ADC input is over ADC's positive input range	
6	ADLOVN	RO	When ADLOVN is high, it means left ADC input is over ADC's negative input range	
5	ADLOVP	RO	When ADLOVP is high, it means left ADC input is over ADC's positive input range	

Bit	Function	Type	Description	Condition
[4:3]	ADOVRS	R/W	Select ADC input limit range.	00: 0.84*Full range 01: 0.71*Full range 10: 0.60*Full range 11: 0.50*Full range
2	ADHP	R/W	Audio ADC high pass filter control, high active	0: disable 1: enable
1	ENADR	R/W	Right channel ADC power control, high active	0: disable 1: enable
0	ENADL	R/W	Left channel ADC power control, high active	0: disable 1: enable

P_AUD_LININCTL **0x9301F028** **Audio ADC line in control register**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	LININRG					LININLG					-	ENLININ
Default	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:12]	Reserve			
[11:7]	LININRG	R/W	Right channel line-in gain	0x00: 12dB 0x01: 10.5dB 0x02: 9dB ... 0x08: 0dB ... 0x1E: -33dB 0x1F: -∞dB

Bit	Function	Type	Description	Condition
[6:2]	LININLG	RO	Left channel line-in gain	0x00: 12dB 0x01: 10.5dB 0x02: 9dB ... 0x08: 0dB ... 0x1E: -33dB 0x1F: -∞dB
0	ENLININ	R/W	Line in power control, high active	0: disable 1: enable

33 USB Interface

33.1 Introduction

The USB Controller includes USB-Host and USB-Device functions. It is compatible with the Universal Serial Bus Specification Revision 2.0. In host mode, it supports USB high-speed(480M bits/s) and full-speed(12M bits/s) data rates. The host interface is compatible with OHCI(Open Host Controller Interface Specification for USB, Release 1.0a) and EHCI(Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0). In device mode, it supports USB high-speed and full-speed data rates

33.2 Features

- USB2.0 HighSpeed/FullSpeed
- Host is compatible with OHCI and EHCI
- UTMI interface to USB PHY
- Embedded 8Byte Setup FIFO, 64Bytes EP0 FIFO, Duplex 512Bytes EP1/EP2 FIFO
- Embedded DMA for Endpoint1/Endpoint2 Bulk transfer
- AXI Interface
- There are 4 endpoints when USB device is enable
 - EP0 – Control Endpoint with 8 bytes setup buffer and 64bytes IN/OUT Data Buffer
 - EP1 – Bulk In Endpoint with 64bytes in Full Speed or 512 bytes in High Speed Data Buffer
 - EP2 – Bulk Out Endpoint with 64bytes in Full Speed or 512 bytes in High Speed Data Buffer
 - EP3 – Interrupt In Endpoint with 64bytes Data Buffer

34 SD/MMC Card controller

34.1 Introduction

Secure Digital (SD) card and Multimedia card (MMC) are flash-based memory card specifically designed to meet the security, capacity, performance and environment requirements inherent in newly emerging audio and video consumer electronic device. SD and MMC card communicate based on an advanced 9-pins and 14-pins interface (Clock, Command, 4xData for SD, 8xData for MMC and 3xPower lines), respectively.

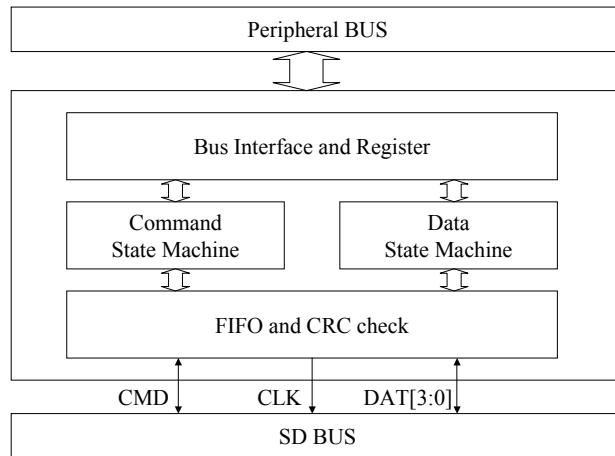
SD IO card is based on and compatible with the SD memory card. The intent of the SD IO card is to provide high-speed data I/O with low power consumption for mobile electronic devices.

The SD/MMC card controller built in this chip is designed to have high performance transfer rate by means of DMA access which can achieve the best performance/cost ratio.

34.2 Features

- Fully compatible with SD/MMC (1, 4 bit data width only) memory card specification
- Accept SD/MMC command directly which improve the compatibility
- Programmable clock speed on the clock line
- Bus clock control while buffer is full
- Interrupt generation
- DMA R/W operation
- 1-bit, 4-bits data width mode supported
- SD IO card interrupt detection
- 2 controllers support

34.3 Block Diagram



34.4 Control Architecture

34.4.1 Command Line Control

There are three major types of response which SD card will send, such as no response, 6 bytes response and 17 bytes response. Programmer needs to specific response type to tell the controller how many data controller needs to receive. Description of each response is shown as following.

1. Response type 3'b000(R0) : No response controller need to receive.
2. Response type 3'b001(R1) : Normal 6-bytes long response.
3. Response type 3'b010(R2) : 17 bytes response type.
4. Response type 3'b011(R3) : 6-bytes long response with command index and CRC7 field equal to 6'b111111.
5. Response type 3'b111(R1b) : Normal 6-bytes long response with busy signal on the DAT0. Controller keeps the clock running until busy signal is cleared.
6. All other response type will be interpreted as normal 6 bytes long response.

In order to receive all 16 bytes (except the first byte) response via the response register, host need to poll the RESPBUFFULL bit in **P_SDCX_STATUS**¹ and read 4 bytes one by one. There are two 32 bits buffers when receive the response. The RESPBUFFULL bit in **P_SDCX_STATUS** will be set when one of the buffers is full. If both of the receive buffer is full, the controller will stop the clock then wait until host read the response register.

NOTE 1 P_SDCX_STATUS: X means 0 or 1 to indicate controller 0 or 1.

The response will be time out after 64 clocks cycle when the host transfer the last bit of the command.

If the card do not response in this period, TIMEOUT bit in **P_SDCX_STATUS** will be set.

34.4.2 Data Line Control

If a command will have data transferred on the data line, the host needs to set the CMDWD bit in **P_SDCX_CMD** and then set TRANDATA bit in **P_SDCX_CMD** to indicate the data direction. The DATLEN is also necessary for the controller to determine how many bytes need to be transferred.

Another thing need to note is about the data length, the data length can be give by bytes, but it is necessary to let it be word aligned to prevent data lose.

In transmit mode, the data will start to transmit 2 clocks after the final bit of the response. After all bits and CRC16 transmit, host will wait for 2 clocks and start to receive the CRC status from the card. DATCRCERR bit in **P_SDCX_STATUS** will be set while data CRC check fail.

In receive mode, host will start to wait for data after the final bit of the command send. This wait will be timeout after 150ms. TIMEOUT bit in **P_SDCX_STATUS** will be set in such condition. If the card transmit the data and CRC16 correctly, the transaction complete smoothly, else the DATCRCERR bit in **P_SDCX_STATUS** will be set.

34.4.3 Card Busy Detection

CARDBUSY bit in **P_SDCX_STATUS** simply reflects the status of the SD0DAT[0] line.

34.4.4 Card Insertion Detection

When both the CMD and DAT State machine is idle and DAT3 on the bus is pulled high, the CARDPRE bit in **P_SDCX_STATUS** will be set, else if the DAT3 is pulled low, the CARDPRE bit in **P_SDCX_STATUS** will be cleared. A de-bounce circuit is used here to prevent the noise on the bus.

34.4.5 Multi-Block Read/Write

The multi-block read/write mode is enabled by set MULBLK bit in **P_SDCX_CMD** to 1. In this mode, host can read/write multi-block in one command. The read/write method is the same as single block mode. The only difference is host need to stop controller manually by set STPCMD bit in **P_SDCX_CMD** when all data is received / transmitted. Host also needs to start CMD12 on the bus to stop the card.

34.4.6 Transmit / Receive Data via APBDMA Channel

Data TX/RX between controller and SD card can be speed up by using APBDMA channel. Following describes the setting of APBDMA channel 0,

Buffer Address must be word-alignment

Read Data from SD Card:

```
* P_APBDMA0_SA0A = BufferAddress;  
* P_APBDMA0_EA0A = BufferAddress + DataNumber - 4  
* P_APBDMA0_SA0 = P_SDCX_DATRX ;  
* P_APBDMA0_CTRL0
```

Bit0: DIR = APB to MIU

Bit1: DMAMODE = Polling Mode

Bit2: APBMODE = Regular mode

Bit3: MIUMODE = Single Buffer

Bit[5:4]: TXMODE = 32bit or 32 bits burst transfer

Bit6: IRQEN = Depends on Programmer

Bit7: CHEN = Enable DMA Channel

Write Data to SD Card:

```
* P_APBDMA0_SA0A = BufferAddress;  
* P_APBDMA0_EA0A = BufferAddress + DataNumber - 4  
* P_APBDMA0_SA0 = P_SDCX_DATTX;  
* P_APBDMA0_CTRL0
```

Bit0: DIR = MIU to APB

Bit1: DMAMODE = Polling Mode

Bit2: APBMODE = Regular mode

Bit3: MIUMODE = Single Buffer

Bit[5:4]: TXMODE = 32bit or 32 bits burst transfer

Bit6: IRQEN = Depends on Programmer

Bit7: CHEN = Enable DMA Channel

34.5 Control Pin Configuration

34.5.1 SD 0 Pin Configuration

I: input; O: output; PP: push-pull

Name	I/O	Description	I/O Port
SD0CLK	O	Clock Pin	
SD0CMD	I/O/PP	Command / Response transfer on this pin	
SD0Data0	I/O/PP	Data Transfer Pin	
SD0Data1	I/O/PP	Data Transfer Pin	
SD0Data2	I/O/PP	Data Transfer Pin	
SD0Data3	I/O/PP	Data Transfer Pin	

34.5.2 SD 1 Pin Configuration

Name	I/O	Description	I/O Port
SD1CLK	O	Clock Pin	
SD1CMD	I/O/PP	Command / Response transfer on this pin	
SD1Data0	I/O/PP	Data Transfer Pin	
SD1Data1	I/O/PP	Data Transfer Pin	
SD1Data2	I/O/PP	Data Transfer Pin	
SD1Data3	I/O/PP	Data Transfer Pin	

34.6 Register Summary

Name	Address	Description
For GPIO setting		
For GPIO setting		
P_SDC0_DATTX	0x92B0B000	SD card data transmit register.
P_SDC1_DATTX	0x92B0C000	
P_SDC0_DATRX	0x92B0B004	SD card data receive register.
P_SDC1_DATRX	0x92B0C004	
P_SDC0_CMD	0x92B0B008	SD card command register.
P_SDC1_CMD	0x92B0C008	
P_SDC0_ARG	0x92B0B00C	SD card argument register.
P_SDC1_ARG	0x92B0C00C	

Name	Address	Description
P_SDC0_RESP	0x92B0B010	SD card response register.
P_SDC1_RESP	0x92B0C010	
P_SDC0_STATUS	0x92B0B014	SD card status register.
P_SDC1_STATUS	0x92B0C014	
P_SDC0_CTRL	0x92B0B018	SD card control register.
P_SDC1_CTRL	0x92B0C018	
P_SDC0_INTEN	0x92B0B01C	SD card interrupt enable register.
P_SDC1_INTEN	0x92B0C01C	

34.7 Register Definition

P_SDC0_DATTX **0x92B0B000** **SD card data transmit register.**

P_SDC1_DATTX **0x92B0C000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	DATA TX															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DATA TX															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	DATA TX	R/W	SD data transmit register. Write data to SD card, data can be written to this register only when DATBUFEMPTY is 1.	

Data transmit register, host writes 32-bits data to this register and the controller will transmit it to SD card. When the data stored in the buffer is transmitted, DATBUFEMPTY bit in status register will be set or the DMA request will be issued. It should be noted data could be written to this register only when **DATBUFEMPTY** is 1.

P_SDC0_DATRX
0x92B0B004
SD card data receive register.
P_SDC1_DATRX
0x92B0C004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	DATA RX															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DATA RX															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	DATA RX	R/W	SD data receive register. Read data from SD card, read data from this register will only valid when the DATBUFFULL is set, otherwise, it will return zeros.	

Data receive register. This register is used to store the read data from the SD card. When 32-bits data is received, DATBUFFULL bit in status register will be set or the DMA request will be issued. It should be noted data could be read from this register only when **DATBUFFULL** is 1.

P_SDC0_CMD
0x92B0B008
SD card command register.
P_SDC1_CMD
0x92B0C008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RESPTYPE INICARD MULBLK TRANDAT CMDWD RUNCMD STPCMD CMDCODE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:15]	Reserved.			
[14:12]	RESPTYPE	R/W	Response type selection. Indicate the response type of the command. Currently, only the response type R2 has response length 128 bits, all other response will have 32 bits length. Response type R1b will keep the controller to wait for busy signal on the SD bus.	000b = No response. 001b = Response type R1. 010b = Response type R2. 011b = Response type R3. 110b = Response type R6. 111b = Response type R1b.

Bit	Function	Type	Description	Condition
11	INICARD	R/W	Initial card. Write This Bit to 1 will start a 74 clock cycles on the clock line.	
10	MULBLK	R/W	Multi-block transfer bit. If this bit is set to "1", it will start multiple block transfer.	0 = Single block transfer 1 = Multiple block transfer
9	TRANDAT	R/W	Transmit / receive data. Indicate if this command transmit or receive data	0 = Receive data (read) 1 = Transfer data (write)
8	CMDWD	R/W	Command with data. Indicate if this command with data or not.	0 = Command without data 1 = Command with data
7	RUNCMD	R/W	Run command. Write "1" to this bit will initiate the SD command on the SD bus according to current configuration of the controller. This bit will be cleared to "0" after the transaction start. Programmer can start a new transaction only when BUSY bit is 0.	
6	STPCMD	R/W	Stop command. Write "1" to this bit will force the controller back to IDLE state. This bit will be cleared to "0" after the controller back to IDLE state.	
[5:0]	CMDCODE	R/W	Command code The command code host wishes to transfer.	

P_SDC0_ARG
0x92B0B00C
SD card argument register.
P_SDC1_ARG
0x92B0C00C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	ARGUMENT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ARGUMENT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	ARGUMENT	R/W	ARGUMENT [31:0] transfer to SD card. Host writes the argument need to be transferred to the card in this register. The SD command needs a 32-bits long argument.	

P_SDC0_RESP
0x92B0B010
SD card response register.
P_SDC1_RESP
0x92B0C010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	RESPONSE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RESPONSE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	RESPONSE	R	Response data from SD card. Response data from SD card, read data from this register will be valid only if the RESPBUFFULL is set.	

This register is used to store the response from the SD card. Commands with response R1, R1b, R3, R6 have 6-bits command index and 32-bits response length. The response will be stored in this register. Command with response R2 will have response length 128 bits. Host need to poll the **RESPBUFFULL** bit in the status register to determine when to read this register. The data in this register is valid only when **RESPBUFFULL** bit is '1'.

P_SDC0_STATUS
0x92B0B014
SD card status register.
P_SDC1_STATUS
0x92B0C014

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	CARDINT	CARDPRE	CARDWP	DATCRCERR	TIMEOUT	DATBUFEMPTY
Default	0	0	0	1	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Function	DATBUFFULL	RESPBUFFULL	RSPCRCERR	RSPIDXERR	DATCOM	CMDCOM	CARDBUSY	BUSY
Default	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:14]			Reserved.	
13	CARDINT	R	Card interrupt. Indicate a SDIO card interrupt is pending. This bit will be set only when IOEN in control register is 1. Host need to clear the interrupt using device specific command. Write 1 to this register will have no effect.	Read 0 = Not occurred Read 1 = Occurred
12	CARDPRE	R/W	Card present. Indicate the card is present. This bit only detects the DAT3 on the SD interface when the controller is idle. Controller's behavior will not be affected by this bit. Host can initial a transaction no matter what this bit is. Write 1 to this register will clear the pending interrupt of card present.	Read 0 = Not occurred Read 1 = Occurred Write 0 = No effect Write 1 = Clear
11	CARDWP	R	Card write protect. Indicate the card is write protect. This bit only detects the write protect pin on the interface. Controller's behavior will not be affected by this bit. Host need to take the responsibility to protect the card.	Read 0 = Not protect Read 1 = Protect
10	DATCRCERR	R/W	Data CRC error. Indicate read data CRC error or write data with CRC error response.	Read 0 = No Error Read 1 = Error Write 0 = No effect Write 1 = Clear
9	TIMEOUT	R/W	Time out. Indicate command response time out or read data response time out.	Read 0 = Not occurred Read 1 = Timeout Write 0 = No effect Write 1 = Clear
8	DATBUFEMPTY	R	Data buffer empty. This bit will be set when data buffer is empty. This bit will be cleared after appropriate number of data had been written to the P_SDCX_DATTX register or write 1 to STPCMD bit in P_SDCX_CMD register.	Read 0 = Buffer Not Empty Read 1 = Buffer Empty

Bit	Function	Type	Description	Condition
7	DATBUFFULL	R	Data buffer full. This bit will be set when data buffer is full. This bit will be cleared after appropriate number of data had been read from the P_SDCX_DATRX register or write 1 to STPCMD bit in P_SDCX_CMD register.	Read 0 = Buffer Not Full Read 1 = Buffer Full
6	RESPBUFFULL	R	Response buffer full. Indicate the RESP register is full. Read to RESP register or start a new transaction or set STPCMD in command register will clear this bit.	Read 0 = Buffer Not Full Read 1 = Buffer Full
5	RSPCRCERR	R/W	Response CRC error. Indicate the CRC bits in the response is failed, this bit will be set if the CRC received is not 6'b111111 in the case of response R3.	Read 0 = No Error Read 1 = Error Write 0 = No effect Write 1 = Clear
4	RSPIDXERR	R/W	Command index in response error. Indicate the command index in the response is failed.	Read 0 = No Error Read 1 = Error Write 0 = No effect Write 1 = Clear
3	DATCOM	R/W	Data complete. Indicate data transfer/receive have completed.	Read 0 = Not occurred Read 1 = Occurred Write 0 = No effect Write 1 = Clear
2	CMDCOM	R/W	Command complete. Indicate corresponding response is received or timeout happened after send a command.	Read 0 = Not occurred Read 1 = Occurred Write 0 = No effect Write 1 = Clear
1	CARDBUSY	R	Card busy. Indicate the SD card is busy (drive the DAT0 low). Host needs to poll this bit after a write command is issued.	Read 0 = Card is not Busy Read 1 = Card is Busy
0	BUSY	R	Controller busy. Indicate the controller is busy.	Read 0 = Controller is idle Read 1 = Controller is B=busy

P_SDC0_CTRL
0x92B0B018
SD card control register.
P_SDC1_CTRL
0x92B0C018

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-		BLKLEN											
Default	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	DUMMYCLKEN	IOEN	DMAMODE	BUSWIDTH										CLKDIV
Default	1	1	1	1	0	0	0	0	0	1	0	1	0	1	0	0

Bit	Function	Type	Description	Condition
[31:28]			Reserved.	
[27:16]	BLKLEN	R/W	Data block length. Data block length to be transferred, in the unit of bytes. The value in this register should be equal to the block length of the SD card.	
[15:12]			Reserved.	
11	DUMMYCLKEN	R/W	SD dummy clock enable. When this bit is set, controller will start clock even if in idle state.	0 = Disable SD dummy clock 1 = Enable SD dummy clock
10	IOEN	R/W	SDIO card interrupt enable. If this bit is set to "1", SD IO Card interrupt detection is enabled, else it is disabled.	0 = Disable 1 = Enable
9	DMAMODE	R/W	DMA mode enable. If this bit is set to "1", it will use DMA channel to transfer data.	0 = Not using DMA mode 1 = Using DMA mode
8	BUSWIDTH	R/W	Bus Width Selection Indicate the data bus width during transfer.	0 = 1 bit data bus 1 = 4 bits data bus
[7:0]	CLKDIV	R/W	Clock Division The Clock Speed on the SD bus is calculated from this register. $F_{SDCLK} = F_{APBCLK}/2(CLKDIV+1)$.	

This register is used to control the clock speed on the SD bus and data block length when transfer or receive data.

This register is changeable only when BUSY bit in **SD Card Status Register** is '0'.

P_SDC0_INTEN
0x92B0B01C
SD card interrupt enable register.
P_SDC1_INTEN
0x92B0C01C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	IPOINT	INSINT	DATBUFEMPTYINT	DATBUFFULLINT	RESPBUFFULLINT	DATCOMINT	CMDCOMINT
Default	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:7]			Reserved.	
6	IPOINT	R/W	SDIO card interrupt enable. Write "1" to this bit will enable the SDIO card interrupt.	0 = Disable 1 = Enable
5	INSINT	R/W	Card insert interrupt enable. Write "1" to this bit will enable the card insert interrupt. Write "1" to the P_SDCX_STATUS[12] will clear this interrupt.	0 = Disable 1 = Enable
4	DATBUFEMPTYINT	R/W	Data buffer empty interrupt enable. Write "1" to this bit will enable the data buffer empty interrupt. This interrupt will be cleared after data had been written to the P_SDCX_DATTX .	0 = Disable 1 = Enable
3	DATBUFFULLINT	R/W	Data buffer full interrupt enable Write "1" to the bit will enable the data buffer full interrupt. This interrupt will be cleared after data had been read from the P_SDCX_DATRX .	0 = Disable 1 = Enable
2	RESPBUFFULLINT	R/W	Response buffer full interrupt enable. Write "1" to this bit will enable the command buffer full interrupt. This interrupt will be cleared when read data from P_SDCX_RESP register or start a new transaction or set STPCMD in P_SDCX_CMD register.	0 = Disable 1 = Enable

Bit	Function	Type	Description	Condition
1	DATCOMINT	R/W	Data complete interrupt enable. Write “1” to this bit will enable the data complete interrupt. Write “1” to P_SDCX_Status[3] will clear this interrupt.	0 = Disable 1 = Enable
0	CMDCOMINT	R/W	Command complete interrupt enable. Write “1” to this bit will enable the command complete interrupt. Write “1” to P_SDCX_STATUS[2] will clear this interrupt.	0 = Disable 1 = Enable

35 Memory Stick

35.1 Introduction

Memory Stick is a high speed memory card. Memory Stick uses long data size (512 bytes) or short data size (32, 64, 128 or 256 bytes) as block size and CRC16 as transaction protection code. Memory Stick uses 10 kinds of TPC to complete data transfer on the bus. To read or write page data, programmer need to set up the card and wait its interrupt to start a read/write TPC (Transfer Protocol Command).

35.2 Features

- Memory Stick v1.xx and Memory Stick Pro compatible
- MSCLK control in buffer overrun/underrun condition
- An 8x32 bits bi-direction FIFO when using page read/write TPC command
- MSINT(CED, ERR, BREQ,CMDNK) detection available
- Multi-byte register read/write available
- Dithering function inside to generate gray-level patterns
- Card insertion/remove detection
- Programmable clock rate

36 UART

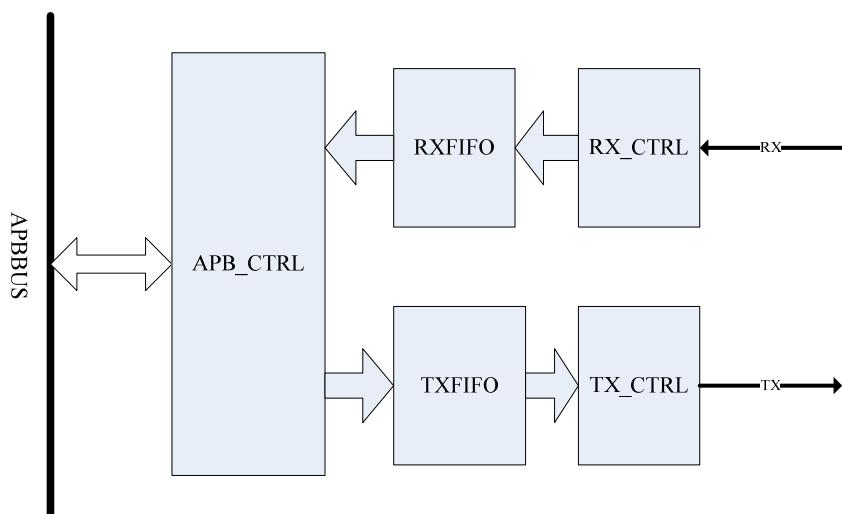
36.1 Introduction

The universal asynchronous receiver/transmitter (UART) performs serial-to-parallel conversion on data characters received from a peripheral device, and parallel-to serial conversion on data characters received from the control unit. General UART supports UART interface (UART_TX and UART_RX). The transmission and reception paths are individually buffered with internal 16 bytes transmit and 16 bytes receive FIFO memories, respectively. This module provides the following feature.

36.2 Features

- Separate 16 bytes transmit and 16 bytes receive FIFO.
- Programmable baud rate generator.
- Support auto-hardware flow control (auto-RTS, auto-CTS).
- False start bit detection.
- Link break generation and detection.
- 5-, 6-, 7- or 8-bit characters.
- Even, odd, no-parity bit generation and detection.
- 1-, 1¹/₂-, or 2-stop bit generation

36.3 Block Diagram



36.4 Transmit / Receive Data via APBDMA Channel

UART data TX/RX between controller and device can be speed up by using APBDMA channel.

Following describes the setting of APBDMA0 channel.

Send data by UART:

```
* P_APBDMA0_SA0A = BufferAddress;  
* P_APBDMA0_EA0A = BufferAddress + DataNumber - 1  
* P_APBDMA0_SA0 = P_UART0_DAT;  
* P_APBDMA0_CTRL0  
    Bit0: DIR = APB to MIU  
    Bit1: DMAMODE = Polling Mode  
    Bit2: APBMODE = Regular mode  
    Bit3: MIUMODE = Single Buffer  
    Bit[5:4]: TXMODE = 8 bits burst transfer  
    Bit6: IRQEN = Depends on Programmer  
    Bit7: CHEN = Enable DMA Channel
```

Receive Data by UART:

```
* P_APBDMA0_SA0A = BufferAddress;  
* P_APBDMA0_EA0A = BufferAddress + DataNumber - 1  
* P_APBDMA0_SA0 = P_UART0_DAT;  
* P_APBDMA0_CTRL0  
    Bit0: DIR = MIU to APB  
    Bit1: DMAMODE = Polling Mode  
    Bit2: APBMODE = Regular mode  
    Bit3: MIUMODE = Single Buffer  
    Bit[5:4]: TXMODE = 8 bits burst transfer  
    Bit6: IRQEN = Depends on Programmer  
    Bit7: CHEN = Enable DMA Channel
```

36.5 Control Pin Configuration

36.5.1 UART 0 Pin Configuration

I: input; O: output;

Name	I/O	Description	I/O Port
UART0_TX	O	Data transfer pin.	
UART0_RX	I	Data receiver pin.	

36.5.2 UART 1 Pin Configuration

Name	I/O	Description	I/O Port
UART1_TX	O	Data transfer pin.	
UART1_RX	I	Data receiver pin.	

36.6 Register Summary

Name	Address	Description
P_SCUA_UART_CFG	0x93007094	UART clock configure. For IO setting
P_UART0_DLL	0x92B04000	Divisor latch LSB
P_UART1_DLL	0x92B06000	
P_UART0_DLM	0x92B04004	Divisor latch MSB
P_UART1_DLM	0x92B06004	
P_UART0_DAT	0x92B04000	UART data port
P_UART1_DAT	0x92B06000	
P_UART0_INTEN	0x92B04004	Interrupt enable register
P_UART1_INTEN	0x92B06004	
P_UART0_INT	0x92B04008	Interrupt status register
P_UART1_INT	0x92B06008	
P_UART0_FCR	0x92B04008	FIFO Control Register
P_UART1_FCR	0x92B06008	
P_UART0_LCR	0x92B0400C	Line Control Register
P_UART1_LCR	0x92B0600C	
P_UART0_MCR	0x92B04010	MODEM Control Register
P_UART1_MCR	0x92B06010	

Name	Address	Description
P_UART0_LSR	0x92B04014	Line Status Register
P_UART1_LSR	0x92B06014	
P_UART0_FSR	0x92B04020	FIFO Status Register
P_UART1_FSR	0x92B06020	
P_UART0_SPR	0x92B04024	Sample Point Register
P_UART1_SPR	0x92B06024	

36.7 Register Definition

The P_UARTX_DLM, P_UARTX_DLL, and P_UARTX_SPR registers specify the UART baud rates.

The baud rate can be calculated with the following equation:

$$baud_rate = \frac{\frac{UART_CLOCK}{(Bit\ time\ length + 1)}}{\{P_UARTX_DLM, P_UARTX_DLL\} + 1}$$

P_UART0_DLL

0x92B04000

Divisor latch LSB

P_UART1_DLL

0x92B06000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	DLL							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]			Reserved.	
[7:0]	DLL	R/W	Divisor latch LSB. After setting DLAB bit in P_UARTX_LCR, user can set UART baud rate by this register.	

P_UART0_DLM
0x92B04004
Divisor latch MSB
P_UART1_DLM
0x92B06004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	DLM							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description														Condition
[31:8]			Reserved.														
[7:0]	DLM	R/W	Divisor latch MSB. After setting DLAB bit in P_UARTX_LCR, user can set UART baud rate by this register.														

P_UART0_DAT
0x92B04000
UART data port
P_UART1_DAT
0x92B06000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	DATA							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description														Condition
[31:8]			Reserved.														
[7:0]	DATA	R/W	UART data port.														

P_UART0_INTEN
0x92B04004
Interrupt enable register
P_UART1_INTEN
0x92B06004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Bit	Function	Type	Description	Condition
[31:3]			Reserved.	
2	RIS	R/W	Receiver line status interrupt enable.	
1	TX	R/W	Transmitter FIFO empty interrupt enable.	
0	RX	R/W	Receiver FIFO available interrupt enable.	

P_UART0_INT

0x92B04008

Interrupt status register

P_UART1_INT

0x92B06008

Bit	Function	Type	Description	Condition
[31:4]			Reserved.	
[3:1]	ID	R	Interrupt ID. When ID set as line status occurred, user must check P_UARTX_LSR for detail status.	000b = Reserved 001b = None. 010b = Tx FIFO empty. 100b = Rx FIFO available. 110b = Line status.
0	IRQ	R	Interrupt pending status.	0 = interrupt is pending. 1 = No interrupt pending.

PUART0 FCR

0x92B04008

FIFO Control Register

P UART1 FCR

0x92B06008

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	RFTRG	-	-	DMA	TFRST	RFRST	FIFOEN	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]			Reserved.	
[7:6]	RFTRG	W	Receiver FIFO trigger level. 00b = 1 byte. 01b = 4 bytes. 10b = 8 bytes. 11b = 14 bytes.	
[5:4]			Reserved.	
3	DMA	W	DMA mode select. These two DMA modes only work when FIFO enabled. In mode 0, controller transmits or receives one byte by DMA every transaction. In mode 1, controller will receive or send data continuously until FIFO full or empty, respectively.	0 = Mode 0. 1 = Mode 1.
2	TFRST	W	Transmitter FIFO reset. It will auto clear.	0 = No effect. 1 = Reset FIFO.
1	RFRST	W	Receiver FIFO reset. It will auto clear.	0 = No effect. 1 = Reset FIFO.
0	FIFOEN	W	Transmitter and receiver FIFO enable.	0 = Disable FIFO. 1 = Enable FIFO.

P_UART0_LCR

0x92B0400C

Line Control Register

P_UART1_LCR

0x92B0600C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	DLAB	B	PB			SB	CL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]			Reserved.	
7	DLAB	R/W	Divisor latch accessing enable. Setting this bit will enable divisor accessing enable by means of P_UARTX_DLL, P_UARTX_DLM and P_UART0_SPR.	0 = Disable. 1 = Enable.
6	B	R/W	Set Break.	
[5:3]	PB	R/W	Parity bit setting.	000b = No parity. 100b = Odd parity. 101b = Even parity. 110b = Logic 1. 111b = Logic 0.
2	SB	R/W	Stop bit length.	0 = 1 bit. 1 = 1½, or 2 bits.
[1:0]	CL	R/W	Character length.	00b = 5 bits. 01b = 6 bits. 10b = 7 bits. 11b = 8 bits.

P_UART0_MCR
0x92B04010
MODEM Control Register
P_UART1_MCR
0x92B06010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	AFC	L	-	-	RTS	DTR	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:6]			Reserved.	
5	AFC	R/W	Auto-hardware flow control enable. For receiver, RTS goes inactive when the trigger level of receiver FIFO is reached, and restores if the receiver FIFO is empty. For transmitter, the serial output would be halted when the CTS bit is inactive.	0 = Disable. 1 = Enable.

Bit	Function	Type	Description	Condition
4	L	R/W	Loopback diagnostic. Provide a local loopback diagnostic feature for testing of the UART. MODEM control signals are inactive	0 = Disable. 1 = Enable.
[3:2]			Reserved.	
1	RTS	R/W	Request to Send. Programmable 1 informs that the UART is ready to exchange data. (If AFC is 1, this bit is useless)	
0	DTR	R/W	Data Terminal Ready. Programming 1 informs that the UART is ready to establish a communications link.	

P_UART0_LSR

0x92B04014

Line Status Register

P_UART1_LSR

0x92B06014

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	RE	TEMT	THRE	BI	FE	PE	OE	DR
Default	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]			Reserved.	
7	RE	R	Receiver FIFO error. This bit indicates that there is at least one parity error, framing error or break interrupt in the RCVR FIFO.	Read 0 = Not occurred. Read 1 = Occurred.
6	TEMT	R	Transmitter FIFO empty. This bit indicates all buffers in the transmitter are empty.	Read 0 = Not occurred. Read 1 = Occurred.
5	THRE	R	Transmitter holding register empty. This bit indicates that the UART is ready to accept new character for transmission.	Read 0 = Not occurred. Read 1 = Occurred.
4	BI	R	Break interrupt. This bit is set to 1 whenever the received data input is held in the logic 0 state for longer than a full word transmission time (START + data + parity + STOP).	Read 0 = Not occurred. Read 1 = Occurred.

Bit	Function	Type	Description	Condition
3	FE	R	Framing error. This bit indicates that the received character does not have a valid STOP bit.	Read 0 = Not occurred. Read 1 = Occurred.
2	PE	R	Parity error. This bit indicates that the received data character does not have the correct parity bit if the PE bit is set to logic 1.	Read 0 = Not occurred. Read 1 = Occurred.
1	OE	R	Overrun error. This bit indicates that the data in the receiver buffer register was not read by the CPU before the next character was transferred into the receiver buffer register. Thus, the previous data will be destroyed.	Read 0 = Not occurred. Read 1 = Occurred.
0	DR	R	Received data ready. When this is set, a complete incoming character has been received and transferred to P_UARTX_DAT.	Read 0 = Not occurred. Read 1 = Occurred.

P_UART0_FSR

0x92B04020

FIFO Status Register

P_UART1_FSR

0x92B06020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	RXF	RXHF	RXE	TXF	TXHE	TXE	
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Bit	Function	Type	Description	Condition
[31:6]			Reserved.	
5	RXF	R	Receiver FIFO full.	Read 0 = Not occurred. Read 1 = Occurred.
4	RXHF	R	Receiver FIFO half full. The data number in receiver FIFO is more than 8 bytes.	Read 0 = Not occurred. Read 1 = Occurred.
3	RXE	R	Receiver FIFO empty.	Read 0 = Not occurred. Read 1 = Occurred.
2	TXF	R	Transmitter FIFO full.	Read 0 = Not occurred. Read 1 = Occurred.

Bit	Function	Type	Description	Condition
1	TXHE	R	Transmitter FIFO half empty. The data number in transmitter FIFO is less than 8 bytes.	Read 0 = Not occurred. Read 1 = Occurred.
0	TXE	R	Transmitter FIFO empty.	Read 0 = Not occurred. Read 1 = Occurred.

P_UART0_SPR
0x92B04024
Sample Point Register
P_UART1_SPR
0x92B06024

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	SP				BT			
Default	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[31:7]			Reserved.	
[7:4]	SP	R/W	Sample point.	
[3:0]	BT	R/W	Bit time length.	

37 Serial Peripheral Interface (SPI)

37.1 Introduction

The SPI/SSI interface device is a master-only interface that enables synchronous serial communication with slave peripherals. It includes the following functions:

Motorola SPI 4 wire interface

Motorola SSI 4 wire interface

It performs parallel-to-serial conversion on data written to an internal 8-bit wide, 8-locations deep transmit FIFO. It also performs serial-to-parallel conversion on the serial input data, and buffers it in a receive FIFO which is also 8-bits wide and 8-locations deep. The module includes a programmable bit rate clock divider to generate the serial output clock SCLK from the input clock PCLK.

37.2 Features

- SPI 4 wire interface
- SSI 4 wire interface
- 4 slave select output
- Independently 8-bit wide, 8-locations transmitter and receiver FIFO
- Programmable polarity and phase
- Programmable bit rate clock divider
- Programmable bytes per timeslot/per frame
- Programmable periodic frame sync pulse

37.3 Block Diagram

The following diagram is a functional block diagram of the SPI module.

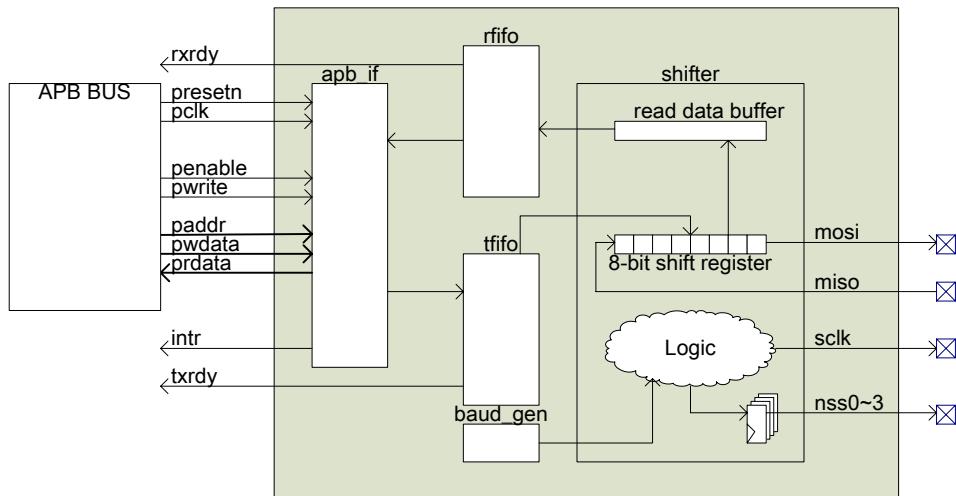


Fig. xx SPI/SSP controller block diagram

Four I/O pin signals are associated with SPI transfer. The SCLK clock line, the MOSI data line, the MISO data line, and active low nSS chip select pin. When a transfer occurs, an 8-bit character is shifted out MOSI data pin while a received 8-bit character is simultaneously shifted in MISO data pin. Thus, the characters in the master and slave are exchanged.

For SSI mode, the SCLK clock line, the MOSI data line, the MISO data line, and active high SYNC frame sync pulse are used. Similarly, when a transfer occurs, an 8-bit shift register in the master and another 8-bit shift register in the slave are shifted to exchange data.

37.4 Register Summary

Name	Type	Address	Description
P_SSP_RDR	R	0x92B08000	Receive Data Register
P_SSP_TDR	W	0x92B08000	Transmit Data Register
P_SSP_IER	RW	0x92B08004	Interrupt Enable Register
P_SSP_IIR	RWC	0x92B08008	Interrupt Identification Register
P_SSP_FCR	RW	0x92B08010	FIFO Control Register
P_SSP_MCR	RW	0x92B08014	Mode Control Register
P_SSP_SSR	RW	0x92B08018	Slave Select Register

Name	Type	Address	Description
P_SSP_FSR	R	0x92B0801C	FIFO Status Register
P_SSP_DLL	RW	0x92B08020	Divisor Latch LSB
P_SSP_DLM	RW	0x92B08024	Divisor Latch MSB
P_SSP_LCR	RW	0x92B08030	Line Control Register
P_SSP_TSR	RW	0x92B08034	Time Slot Register
P_SSP_SDR	RW	0x92B08038	Frame Sync divisor Register

37.5 Register Definition

P_SSP_RDR/P_SSP_TDR 0x92B08000 Receive/Transmit Data Register							
Bit	15	14	13	12	11	10	9
Function	-						
Default	0						
P_SSP_RDR/P_SSP_TDR							
	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:8]	-	-	Reserved					-
[7:0]	P_SSP_RDR	R	Receive Data Register					-
[7:0]	P_SSP_TDR	W	Transmit Data Register					-

P_SSP_IER 0x92B08004 Interrupt Enable Register								
Bit	15	14	13	12	11	10	9	
Function	-							
Default	0							
P_SSP_IER								
	7	6	5	4	3	2	1	0
	-				OURNE	RFTGE	RDRRE	TDREE
	0				0	0	0	0

Bit	Function	Type	Description					Condition
[15:4]	-	-	Reserved					-
[3]	OURNE	R/W	FIFO Overrun or Underrun interrupt enable					-
[2]	RFTGE	R/W	RCVR FIFO Trigger level reach interrupt enable					-
[1]	RDRRE	R/W	Receive data register ready interrupt enable					-

Bit	Function	Type	Description					Condition
[0]	TDREE	R/W	Transmit data register empty interrupt enable					-

P_SSP_IIR **0x92B08008** **Interrupt Identification Register**

Bit	15	14	13	12	11	10	9	8
Function							-	

Default 0

7	6	5	4	3	2	1	0
				OURN	RFTG	RDRR	TDRE
0				0	0	0	0

Bit	Function	Type	Description					Condition
[15:4]	-	-	Reserved					-
[3]	OURN	RWC	FIFO Overrun or Underrun interrupt					-
[2]	RFTG	RWC	RCVR FIFO Trigger level reach interrupt					-
[1]	RDRR	RWC	Receive data register ready interrupt					-
[0]	TDRE	RWC	Transmit data register empty interrupt					-

P_SSP_FCR **0x92B08010** **FIFO Control Register**

Bit	15	14	13	12	11	10	9	8
Function							-	

Default 0

7	6	5	4	3	2	1	0
RFTRG			-	DMA	TFRST	RFRST	FFE
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:8]	-	-	Reserved					-
[7:6]	RFTRG	R/W	RCVR FIFO Trigger Level (byte)					2'b00 1 2'b01 3 2'b10 5 2'b11 7
[5:4]	-	-	Reserved					-
[3]	DMA	R/W	DMA Mode Select					-
[2]	TFRST	R/W	XMIT FIFO Reset (this bit is self-clearing)					-

Bit	Function	Type	Description	Condition
[1]	RFRST	R/W	RCVR FIFO Reset (this bit is self-clearing)	-
[0]	FFE	R/W	XMIT, RCVR FIFO enable	-

Transmitter and receiver DMA signaling is available through two pins (txrdy and rxrdy). When operating in the FIFO mode, one of two type of DMA mode can be selected via DMA register. In DMA mode 0, it supports the single transfer. And in the DMA mode 1, multiple transfers are made continuously until the RCVR FIFO has been emptied or XMIT FIFO has been filled.

	DMA Mode 0	DMA Mode 1
Txrdy	The txrdy will be active when there are no characters in the XMIT FIFO or XMIT holding register. And txrdy will go inactive after a character is loaded into the XMIT FIFO or holding register.	The txrdy will be active when there are no characters in the XMIT FIFO. And txrdy will go inactive after the XMIT FIFO is complete full.
Rxrdy	The rxrdy will be active when there is at least 1 character in the RVCR FIFO or RVCR buffer register. And rxrdy will go inactive when there are no characters in the FIFO or buffer register.	The rxrdy will be active when the trigger level has been reach. And rxrdy will go inactive when there are no characters in the FIFO or buffer register.

P_SSP_MCR **0x92B08014** **Mode Control Register**

Bit	15	14	13	12	11	10	9	8
Function						-		
Default						0		

7	6	5	4	3	2	1	0
-			LOOP	CPOL	CPHA	MNSS	LSBF
0			0	0	0	0	0

Bit	Function	Type	Description	Condition	
[15:5]	-	-	Reserved	-	
[4]	LOOP	R/W	Provide a local loopback feature for diagnostic testing	-	
[3]	CPOL	R/W	Clock Polarity (SPI mode only)	1'b0	Active high clocks SCLK idles at low state
				1'b1	Active low clocks SCLK idles at high state

Bit	Function	Type	Description	Condition	
[2]	CPHA	R/W	Clock Phase (SPI mode only)	1'b0	Sampling of data occur at odd edge of SCLK clock
				1'b1	Sampling of data occur at even edge of SCLK clock
[1]	MNSS	R/W	Manual Slave Select Assertion	1'b0	Slave Select output asserted by master state machine
				1'b1	Slave Select output follows value in slave select register (SPI_SSR)
[0]	LSBF	R/W	This bit does not affect the position of the MSB, LSB in the data registers. Reads and writes of the data registers always have the MSB in bit 7. The bit controls the MOSI, MISO to transfer LSB bit first.		

Software can selected any of four combinations of serial clock phase and polarity using two bits in the mode control register. The clock phase and polarity should be identical for the master SPI device and the communicating slave device.

Fig. 1 is a timing diagram of an SPI transfer where CPHA equals zero. When CPHA is 0, the nSS line must be deasserted and reasserted between each successive serial byte (Fig. 2).

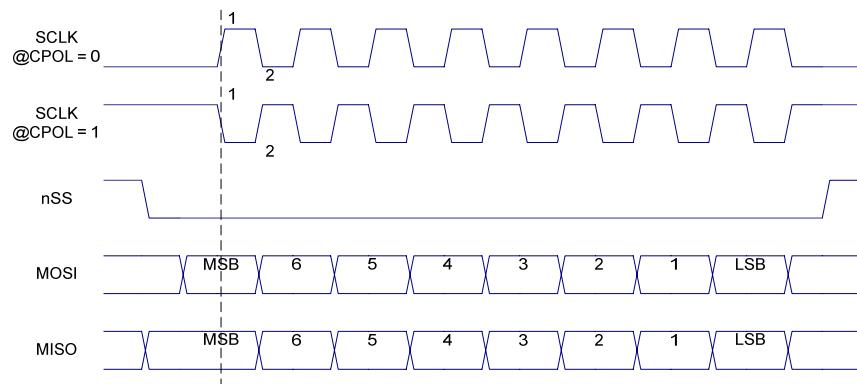


Fig. 1 CPHA equals zero SPI transfer format

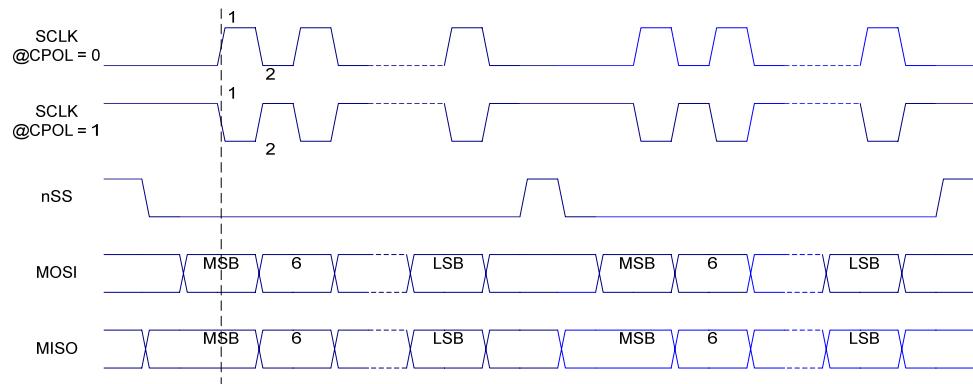


Fig. 2 CPHA equals zero SPI continuously transfer format

Fig. 3 is a timing diagram of an SPI transfer where CPHA equals one. When CPHA = 1, the nSS line may remain active low between successive transfers (Fig. 4).

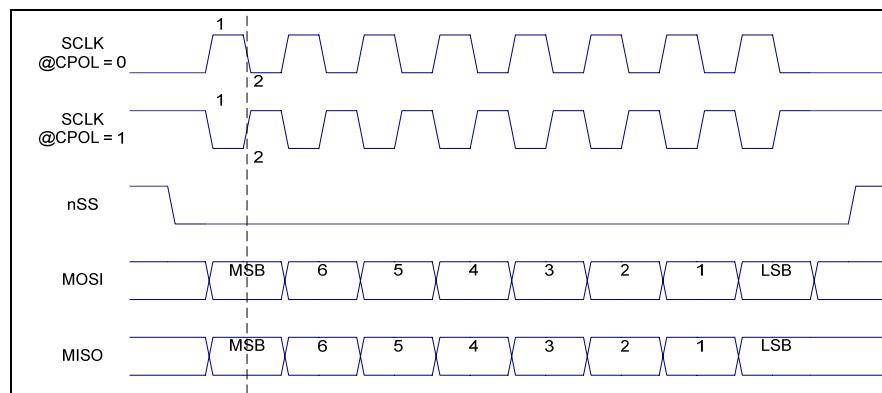


Fig. 3 CPHA equals one SPI transfer format

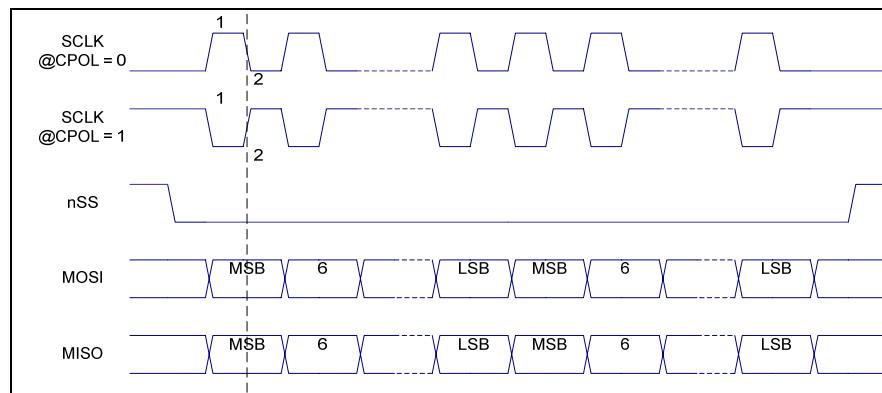


Fig. 4. CPHA equals one SPI continuously transfer format

P_SSP_SSR

0x92B08018

Slave Select Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0							

7	6	5	4	3	2	1	0
nSS3E	nSS2E	nSS1E	nSS0E	MNSS3	MNSS2	MNSS1	MNSS0
0	0	0	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7]	nSS3E	R/W	Slave Select n Enable:	-
[6]	nSS2E	R/W	Slave Select n Enable is valid when MNSS is disabled.	-
[5]	nSS1E	R/W	The master machine would control the active slave select register automatically.	-
[4]	nSS0E	R/W		-
[3]	MNSS3	R/W	Manual Slave Select n:	-
[2]	MNSS2	R/W	Manual Slave Select n is valid when MNSS is enabled.	-
[1]	MNSS1	R/W	The value must be set the one-cold encoded vector or all one value. Written into it.	-
[0]	MNSS0	R/W	Transmit FIFO Empty Flag	-

P_SSP_FSR

0x92B0801C

FIFO Status Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0							

7	6	5	4	3	2	1	0
RFORN	RFURN	RFFUL	RFEMT	TFORN	TFURN	TFFUL	TFEMT
0	0	0	1	0	0	0	1

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7]	RFORN	R	RCVR FIFO Overrun Issue RFRST will clear this flag	-
[6]	RFURN	R	RCVR FIFO Underrun Issue RFRST will clear this flag	-
[5]	RFFUL	R	RCVR FIFO Full Read the receive data register will clear this flag	-

Bit	Function	Type	Description	Condition
[4]	RFEMT	R	RCVR FIFO Empty Master state machine will clear this flag if the data is received	-
[3]	TFORN	R	XMIT FIFO Overrun Issue TFRST will clear this flag	-
[2]	TFURN	R	XMIT FIFO Underrun Issue TFRST will clear this flag	-
[1]	TFFUL	R	XMIT FIFO Full Master state machine will clear this flag if the data is transmitted	-
[0]	TFEMT	R	XMIT FIFO Empty Write the transmit data register will clear this flag	-

P_SSP_DLL / P_SSP_DLM 0x92B08020/0x92B08024 Divisor Latch (LSB)/(MSB)

Bit	15	14	13	12	11	10	9	8
Function								-
Default								0
P_SSP_DLL / P_SSP_DLM								
0 0 0 0 0 0 0 0 0								

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:0]	P_SSP_DLL	R/W	Divisor Latch (LSB)	-
[7:0]	P_SSP_DLM	R/W	Divisor Latch (MSB)	-

These two registers specify the SPI/SSI baud rates. The baud rate can be calculated with the following equation:

$$\text{baud_rate} = \frac{\frac{\text{PCLK}}{2}}{\{\text{DLM}, \text{DLL}\} + 1}$$

The SPI/SSI baud rate Example:

Baud rate	PCLK (Hz)	Divisor Latch	Error (%)
20000000	40500000	0	-1.25%
4000000	40500000	4	-1.25%
512000	40500000	39	1.12%
256000	40500000	78	-0.13%
128000	40500000	157	-0.13%

P_SSP_LCR 0x92B08030 Line Control Register							
Bit	15	14	13	12	11	10	9
Function	-						
Default	0						
	7	6	5	4	3	2	1
	-			SSIEN	SBLEN	CONT	MODE
	0			0	0	0	0

Bit	Function	Type	Description	Condition	
[15:5]	-	-	Reserved	-	
[4]	SSIEN	R/W	SSI Normal/Network Mode Enable	-	
[3]	SBLEN	R/W	Frame Sync Type (SSI mode only)	1'b0	Word Length Frame Sync
				1'b1	Bit Length Frame Sync
[2]	CONT	R/W	Clock Type (SSI mode only)	1'b0	Gated Clock
				1'b1	Continuously running clock
[1:0]	MODE	R/W	Manual Slave Select Assertion	2'00	SPI Mode
				2'01	SSI Normal Mode
				2'10	SSI On-demand Mode
				2'11	SSI Network Mode

P_SSP_TSR 0x92B08034 Time Slot Register								
Bit	15	14	13	12	11	10	9	
Function	-							
Default	0							
	7	6	5	4	3	2	1	
	TSLOT				FLEN			
	0				0			

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:4]	TSLOT	R/W	Time Slot (SSI-network Mode only) These bits are used to select the length of the data bytes being transferred per timeslot. Value 0 means one byte per timeslot, 1 represents two bytes per timeslot, ... etc.	-
[3:0]	FLEN	R/W	Frame Length (SSI-network Mode only): These bits are used to select the length of the data bytes being transferred per frame. Value 0 means one byte per frame, 1 represents two bytes per frame, ... etc.	-

P_SSP_FSD
0x92B08038
Frame Sync divisor Register

Bit	15	14	13	12	11	10	9	8								
Function	-															
Default	0															
FSD																
0																

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:0]	FSD	R/W	Frame Sync Divisor Latch The registers are valid when the operation mode is SSI-normal mode, or SSI-network mode.	-

The periodic frame sync rate can be calculated with the following equation:

$$\text{frame_sync} = \frac{\text{baud_rate}}{\text{FSD} + 1}$$

The following timing diagram shows the difference between the bit length frame sync and the word length frame sync. If the bit length frame sync is selected, this indicates the frame sync was high during the last bit period prior to the current timeslot. If word length frame sync is selected, this indicates that the frame sync was high at least at the beginning of the timeslot.

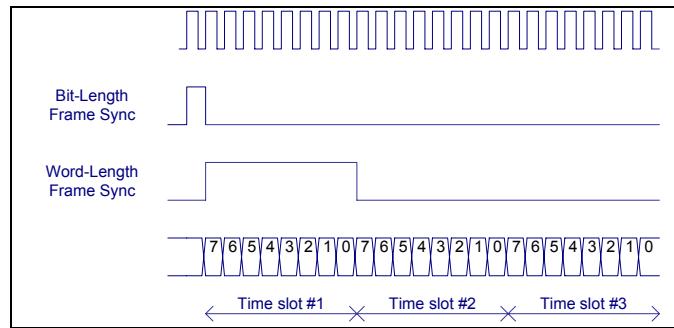


Fig. 5 Frame Sync Timing Diagram

A continuous clock is required in applications such as communicating with some codecs where the clock is used for more than just data transfer. The following figure illustrates the difference between continuous clock and gated clock system.

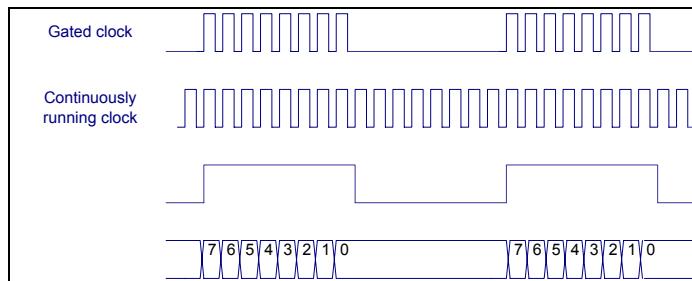


Fig. 6 Clock Type bit Operation

The SSI has three basic operating modes. For normal mode, the SSI functions with one byte per frame. For network mode, 1 to 16 bytes may be used per frame. In either case, the transfers are periodic. The data-driven on-demand mode is intended to be used to communication with device on a non-periodic base transfer.

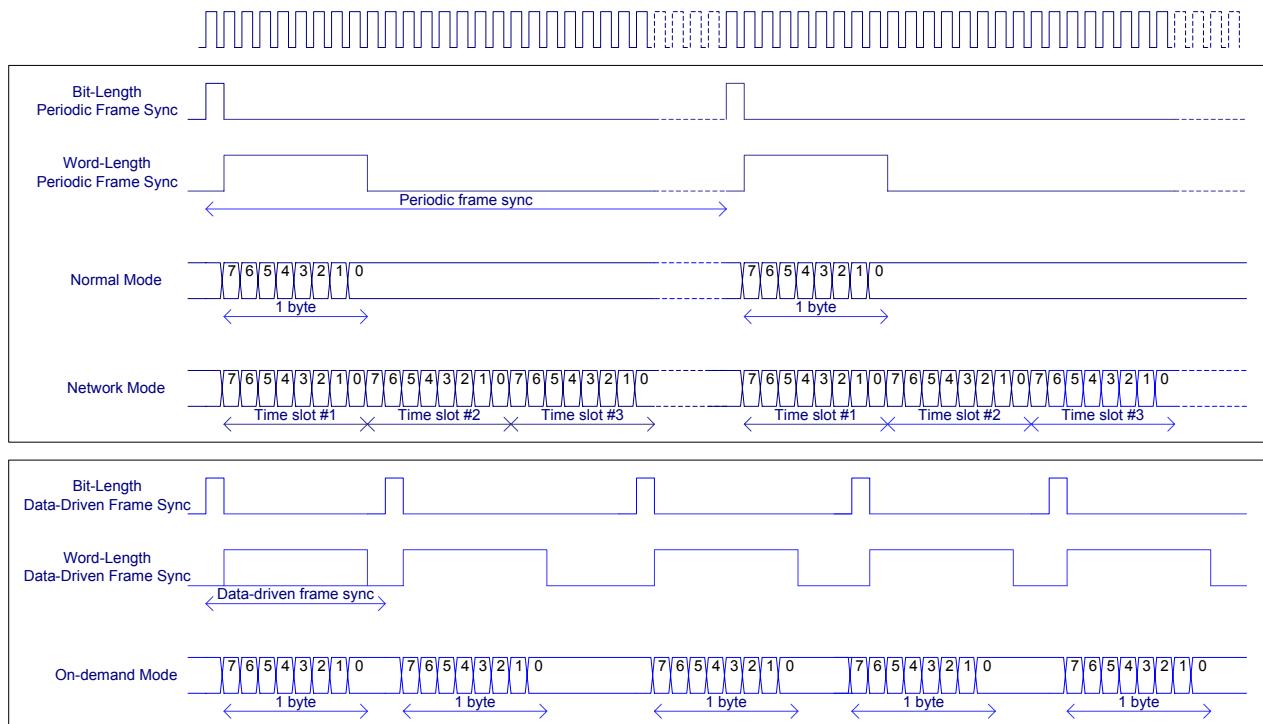


Fig. 7 Operation Mode

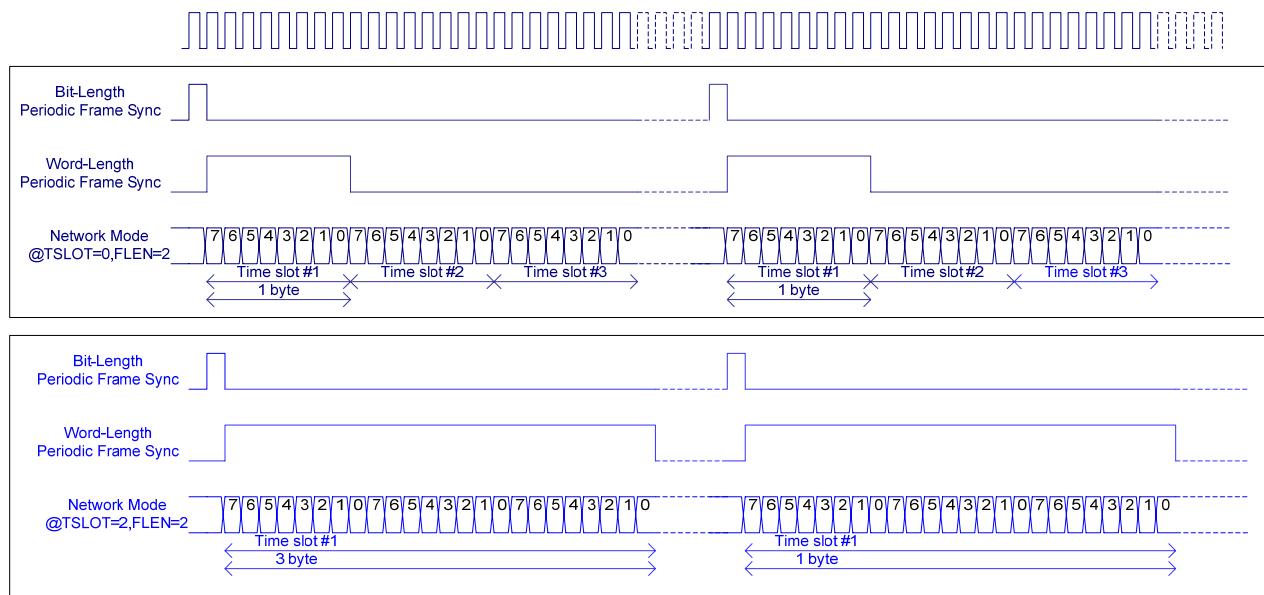


Fig. 8 TimeSlot on SSI-network mode

38 Serial Peripheral Interface (I2C)

38.1 Introduction

The multi-master I2C-bus controller provides a mechanism to communicate between bus masters and peripheral devices by using two signals, a serial data line (SDA) and a serial clock line (SCL). To avoid all possibilities of confusion, data loss and blockage of information, the master and slave devices must have a defined protocol.

In multi-master I2C-bus mode, multiple microprocessor can receive or transmit serial data to or from slave devices. Fig1 is an example of two micro-controllers using I2C-bus to do system configuration and data communication.

The master that initiates a data transfer over the I2C-bus is responsible for terminating the transfer. It is possible to combine several masters, in addition to several slaves onto an I2C-bus to form a multi-master system.

If more than one master simultaneously tries to control the line, an arbitration procedure decides which master gets priority. The maximum number of devices connected to the bus is dictated by the maximum allowable capacitance on the lines, 400 pF, and the protocol's addressing limit of 16 K.

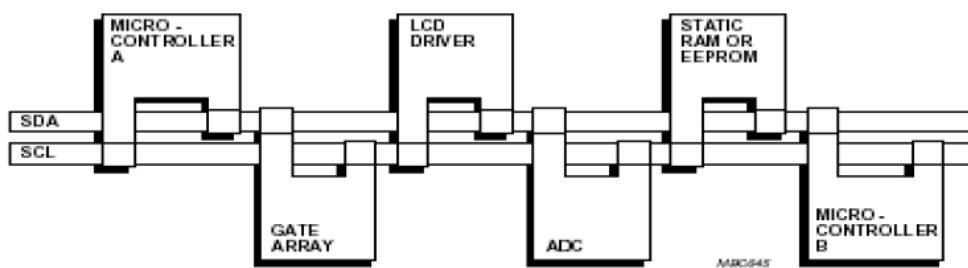


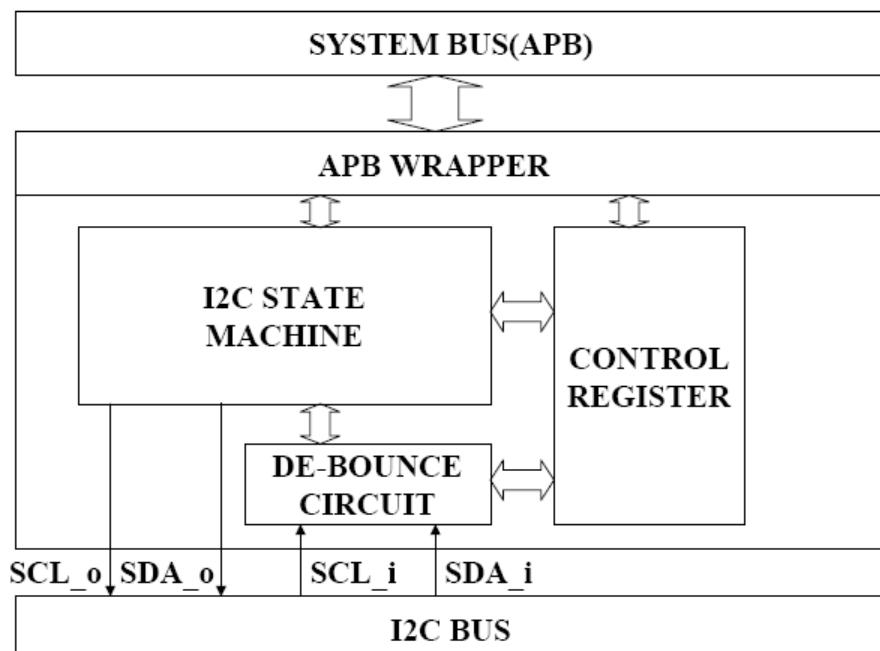
Fig1. Example of I2C-Bus configuration

38.2 Features

- Master transmit/receive mode supported.
- Slave transmit/receive mode supported.
- Detection of bus arbitration fail.
- Interrupt generation.
- Programmable ACK generation.
- Programmable clock speed in master mode.
- Input de-bounce circuit.
- 7 Bit addressing.

38.3 Block Diagram

The following diagram is a functional block diagram of the I2C module.



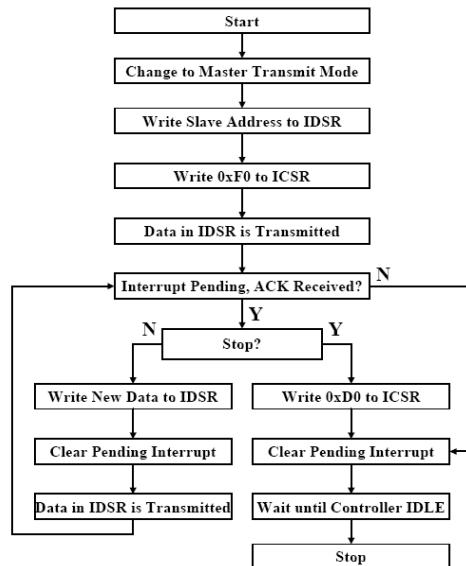
38.4 I2C Control Pin Configuration

Name	I/O	Description
I2C SCL	O	I2C Clock signal.
I2C SDA	O/I	I2C data signal

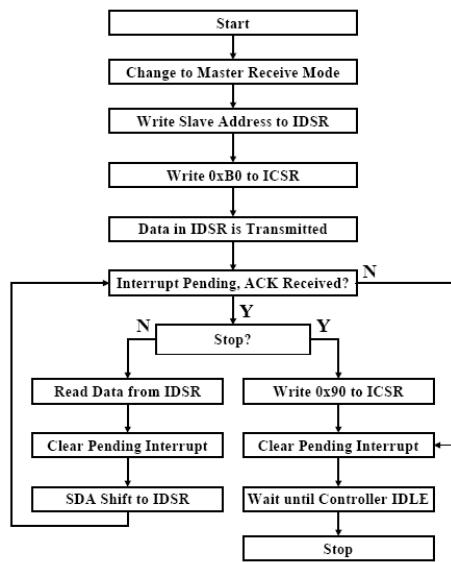
38.5 I2C Transfer and Receiver Mode

I2C supports master/slave transmit/receiver mode. The follow is the operation state.

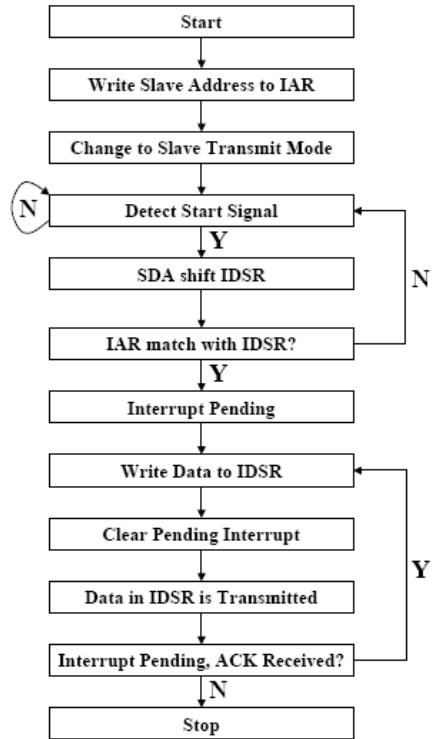
- Master Transmit Mode



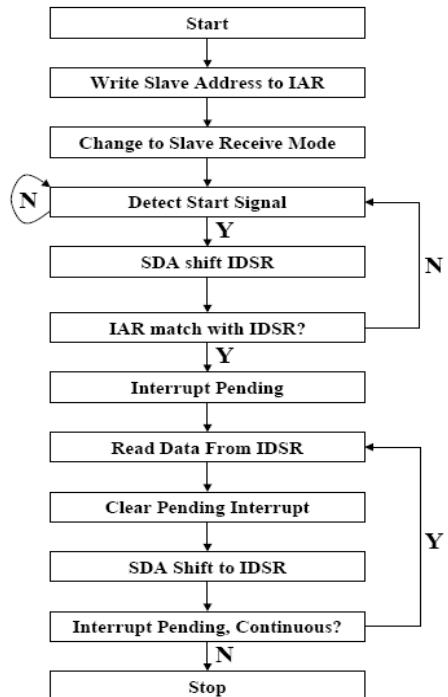
- Master Receiver Mode



■ Slave Transmit Mode



■ Slave Receive Mode



38.6 Register Summary

Name	Address	Description
P_I2C_ICCR	0x90002000/0x92B02000	I2C Control Register
P_I2C_ICSR	0x90002004/0x92B02004	I2C Transmit/Receive Status Register
P_I2C_IAR	0x90002008/0x92B02008	I2C address Register
P_I2C_IDSR	0x9000200C/0x92B0200C	I2C data Register
P_I2C_IDEBCLK	00x90002010/0x92B02010	I2C deblock register

38.7 Register Definition

P_I2C_ICCR 0x90002000/0x92B02000 I2C Control Register								
Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	ACKEN	CLKSELPRE	INTREN	INTRPEND	TXCLKPRE			
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
7	ACKEN	R/W	I2C-bus acknowledge enable bit	0: Disable ACK generation 1: Enable ACK generation
6	CLKSELPRE	R/W	Source clock of I2C-bus transmit clock prescaler selection bit	0: "I2CCLK=f PCLK/16" 1: "I2CCLK=fPCLK/512"
5	INTREN	R/W	I2C-Bus Tx/Rx interrupt enable bit	0: Disable interrupt 1: Enable interrupt
4	INTRPEND	R/W	I2C-bus Tx/Rx interrupt pending flag Writing "1" is impossible. NOTES: A I2C-bus interrupt occurs 1. When a 1-byte transmission or receiver operation is terminated. 2. When a general call or a slave address match occurs. 3. If bus arbitration fails.	0: No interrupt pending (read), this bit is cleared (write). 1: Interrupt is pending (read) No effect. Namely "1" doesn't be written to this bit.

Bit	Function	Type	Description	Condition
[3:0]	TXCLKPRE	R/W	I2C-Bus transmit clock pre-scaler. Transmit clock frequency is determined by this 4-bit pre-scaler value, according to the following formula: $Tx\ clock = I2CCLK/(ICCR[3:0]+1)$ 1. I2CCLK is determined by ICCR[6] 2. Tx clock can vary by SCL transition time. 3. When ICCR[6] = 0, "ICCR[3:0] = 0x0 or 0x1" is not available.	-

P_I2C_ICSR **0x90002004/0x92B02004** **I2C Status Register**

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ICSR[7:6]	ICSR[5]	ICSR[4]	ICSR[3]	ICSR[2]	ICSR[1]	ICSR[0]	
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:6]	ICSR[7:6]	R/W	I2C-bus master/slave Tx/Rx mode select bits. NOTE: Under following 2 kinds of situation2, the I2C mode slave receiver mode automaticaly. In slave mode, receive slave address 0x00. In master mode, detects bus arbitration fail.	00: Slave receive mode. 01: Slave transmit mode. 10: Master receive mode. 11: Master transmit mode.
5	ICSR[5]	R/W	I2C-Bus busy signal status bit.	0: I2C-bus not busy (when read), I2C-bus interface STOP signal generation (when write). 1: I2C-bus busy (when read), I2C-bus interface START signal generation (when write).

Bit	Function	Type	Description	Condition
4	ICSR[4]	R/W	I2C-bus data output enable/disable bit.	0: Disable Rx/Tx 1: Enable Rx/Tx
3	ICSR[3]	R	I2C-bus arbitration procedure status flag bit.	0: Bus arbitration status okay 1: Bus arbitration failed during serial I/O
2	ICSR[2]	R	I2C-bus address-as-slave status flag bit.	0: START/STOP condition was generated 1: Received slave address matches the address value in the IAR.
1	ICSR[1]	R	I2C-bus address zero status flag bit.	0: START/STOP condition was generated 1: Received slave address is "0x00"
0	ICSR[0]	R	I2C-bus last-received bit status flag bit.	0: Last-received bit is "0" (ACK was received) 1: Last-received bit is "1" (ACK was not received)

P_I2C_IAR
0x90002008/0x92B02008
I2C Address Register

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
IAR[7:1]							-
0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	-	-	Reserved	-
[7:1]	IAR[7:1]	R/W	7-bit slave address, latched from the I2C-bus: When serial output enable = "0" in the ICSR, IAR is write enable. It is allowable to read IAR value at any time, regardless of the current serial output enable bit, ICSR[4], condition — Slave address = [7:1] — Not mapped = [0]	

Bit	Function	Type	Description					Condition
0	-	-	Reserved.					-

P_I2C_IDSR 0x9000200C/0x92B0200C I2C Data Register

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0	
IAR[7:1]								-
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:8]	-	-	Reserved					-
[7:1]	IAR[7:1]	R/W	8-bit data shift register for I2C-bus Tx/Rx operation: When serial output enable = "1" in the ICSR, IDSR is write-enabled. It is allowable to read the IDSR value at any time, regardless of the current serial output enable bit, ICSR[4], status.					-
0	-	-	Reserved.					-

P_I2C_IDEBCCLK 0x90002010/0x92B02010 I2C Deblock Register

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0	
IDEBCCLK[7:0]								
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:8]	-	-	Reserved					-
[7:0]	IDEBCCLK	R/W	I2C-bus input de-bounce clock register.					-

39 TIMER

39.1 Introduction

The configurable five channels 16-bit Timer is an AMBA slave device. The Timer consists of an independent five channel general-purpose 16-bit timers. The timer can operate to up or down direction and the control of count direction can be selected by external input or using control register. The count interval of timer is determined by pre-scale register. Timer generates the interrupt signal when the internal counter reaches at the end value. A compare register is used to generate PWM output (TOUT).

39.2 Features

- Supports an independent one to five channel 16-bit timers
- Can operate to up or down direction
- Supports an input for counting the external event
- Contains 16-bit pre-scale register which supports divided by N counter for each channel
- Generates time-out interrupt signals from each channel
- PWM output

39.3 Register Summary

Name	Address	Description
P_TIMER0_CTRL	0x90000000	Timer 0 Control Register
P_TIMER0_PSR	0x90000004	Timer 0 Pre-scale Register
P_TIMER0_LDR	0x90000008	Timer 0 Load Value Register
P_TIMER0_ISR	0x9000000C	Timer 0 Interrupt Register
P_TIMER0_CMP	0x90000010	Timer 0 Compare Register
P_TIMER1_CTRL	0x90000020	Timer 1 Control Register
P_TIMER1_PSR	0x90000024	Timer 1 Pre-scale Register
P_TIMER1_LDR	0x90000028	Timer 1 Load Value Register
P_TIMER1_ISR	0x9000002C	Timer 1 Interrupt Register
P_TIMER1_CMP	0x90000030	Timer 1 Compare Register
P_TIMER2_CTRL	0x90000040	Timer 2 Control Register
P_TIMER2_PSR	0x90000044	Timer 2 Pre-scale Register
P_TIMER2_LDR	0x90000048	Timer 2 Load Value Register
P_TIMER2_ISR	0x9000004C	Timer 2 Interrupt Register

Name	Address	Description
P_TIMER2_CMP	0x90000050	Timer 2 Compare Register
P_TIMER3_CTRL	0x90000060	Timer 3 Control Register
P_TIMER3_PSR	0x90000064	Timer 3 Pre-scale Register
P_TIMER3_LDR	0x90000068	Timer 3 Load Value Register
P_TIMER3_ISR	0x9000006C	Timer 3 Interrupt Register
P_TIMER3_CMP	0x90000070	Timer 3 Compare Register
P_TIMER4_CTRL	0x90000080	Timer 4 Control Register
P_TIMER4_PSR	0x90000084	Timer 4 Pre-scale Register
P_TIMER4_LDR	0x90000088	Timer 4 Load Value Register
P_TIMER4_ISR	0x9000008C	Timer 4 Interrupt Register
P_TIMER4_CMP	0x90000090	Timer 4 Compare Register

39.4 Register Definition

P_TIMERX_CTRL 0x90000000 + (0x20*X) TimerX Control Register								
Bit	15	14	13	12	11	10	9	8
Function	CLK_SRC	-		M		ES		
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	OM	-	UD	PWMON	OE	IE	TE
0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:14]	CLK_SRC	R/W	Timer clock source selection: 00: PCLK 01: 32768Hz 10: 27M/16 = 1.6875MHz 11: not used				
[13:12]	Reserved	-	-				
[11:10]	M	R/W	Operating mode. 00: Free running timer mode. 01: Period timer mode 10: Free running counter mode 11: Period counter mode				

Bit	Function	Type	Description	Condition
[9:8]	ES	R/W	External input active edge selection. 00: Positive edge 01: Negative edge 10: Both edge 11: unused	
7	Reserved	R/W	Reserved	
6	OM	R/W	Timer Output mode 0: Toggle mode 1: Pulse mode. This bit will only effect when timer output is in normal mode.	
5	-	-	-	
4	UD	R/W	Up/down counting selection 0: Down Counting 1: Up Counting	
3	PWMON	R/W	Output mode of timer 0: Normal mode 1: PWM mode	
2	OE	R/W	Output enable register 0: Disable timer output 1: Enable timer output	
1	IE	R/W	Interrupt enable register. 0: Disable interrupt generation 1: Enable interrupt generation	
0	TE	R/W	Timer enable register. 0: Timer Disable 1: Timer Enable	

P_TIMERX_PSR 0x90000004 + (0x20*X) TimerX Pre-Scale Register

Bit	15	14	13	12	11	10	9	8
Function								

Default 0 0 0 0 0 0 0 0

7	6	5	4	3	2	1	0
PSR							

0 0 0 0 0 0 0 0

Bit	Function	Type	Description					Condition
[15:0]	PSR	R/W	Prescaler value. Timer clock ticks = clock source/(PSR+1)					

TIMERX_LDR register is a load register and used to specify the time-out duration. When the internal down counter of timer reaches at the end, the contents of this register is automatically reloaded to counter. Read operation on this register returns the current value of internal counter.

P_TIMERX_LDR 0x90000008 + (0x20*X) TimerX Load Value Register

Bit	15	14	13	12	11	10	9	8
Function	LDR or VLR							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
	LDR or VLR							
	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	LDR	W	A write to this register programs the counter load value with the number of clock ticks					
[15:0]	VLR	R	A read to this register returns the current contents of 16-bit counter					

P_TIMERX_ISR 0x9000000C + (0x20*X) TimerX ISR Register

Bit	15	14	13	12	11	10	9	8
Function	-							
Default	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
	-							
	0	0	0	0	0	0	0	0
	ISR							

Bit	Function	Type	Description					Condition	
0	ISR	R/C	A read to this register returns the current contents of 16-bit counter. This bit is set to 1 if the TimerxISR interrupt is asserted.						

Timer compares value register. This register is used only in PWM output mode. When the counter value reached the value in this register and PWMON bit in TxCTR is set to 1, the timer output will toggle (TOUT). This can be used to modify the duty cycle of the timer output.

P_TIMERX_CMP 0x90000010 + (0x20*X) TimerX CMP Register

Bit	15	14	13	12	11	10	9	8	
Function	CMP								
Default	0	0	0	0	0	0	0	0	
CMP									
0 0 0 0 0 0 0 0 0									

Bit	Function	Type	Description					Condition	
[15:0]	CMP	R/W	Timer compare value register						

39.5 Operation Mode

The timer operates in timer mode or counter mode. Basically, these two modes are almost the same except the counter increase/decrease signal. In timer mode, the timer is increased/decreased by PCLK but in counter mode, it is by the rising/falling of external signal. In counter mode, the counter operates by an external signal (EXTIN) input.

In both free running and period mode, the counter always runs to 0xFFFF(up counting) or to 0x0000(down counting). But after the counter reached the limit, the load value is LDR in period mode and 0xFFFF(down) or 0x0000(up) in free running mode. Here is an example to generate a 10kHz, duty cycle 60% PWM output using this timer in a 66MHz system.

Write 0x19C8 to TxLDR register => Set frequency to 10kHz.

Write 0xA50 to TxCMP register => Set Duty cycle to 60%

Write 0x40D to TxCTR => Enable timer, Enable timer output, Enable PWM mode, set to Period timer mode.

40 RTC

40.1 Introduction

There are a RTC controller and RTC macro in GPL32900A. RTC macro consists of a 48-bit counter. The value of the counter is translated to year, date, and time by the firmware. The RTC controller provides the interface between CPU and RTC macro. The power supply to the RTC macro is separated from the power supply to the other part of the GPL32900A. While the chip powered down, the application must keep supplying power to the RTC macro to maintain RTC operation. Separating the powers guarantees minimum current leakage.

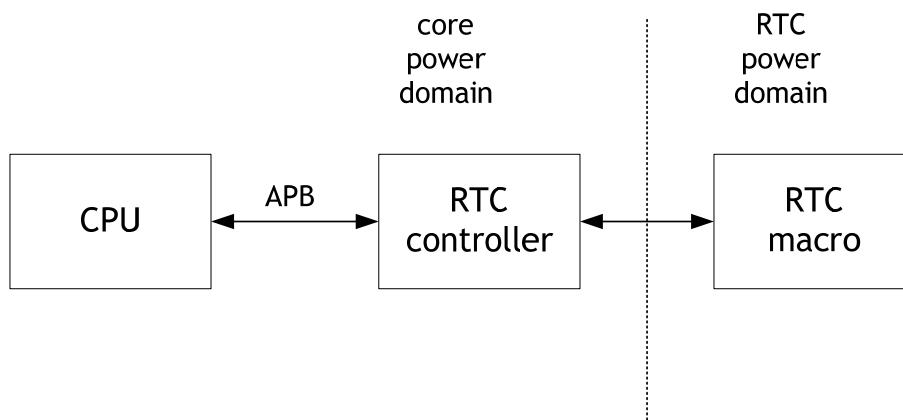
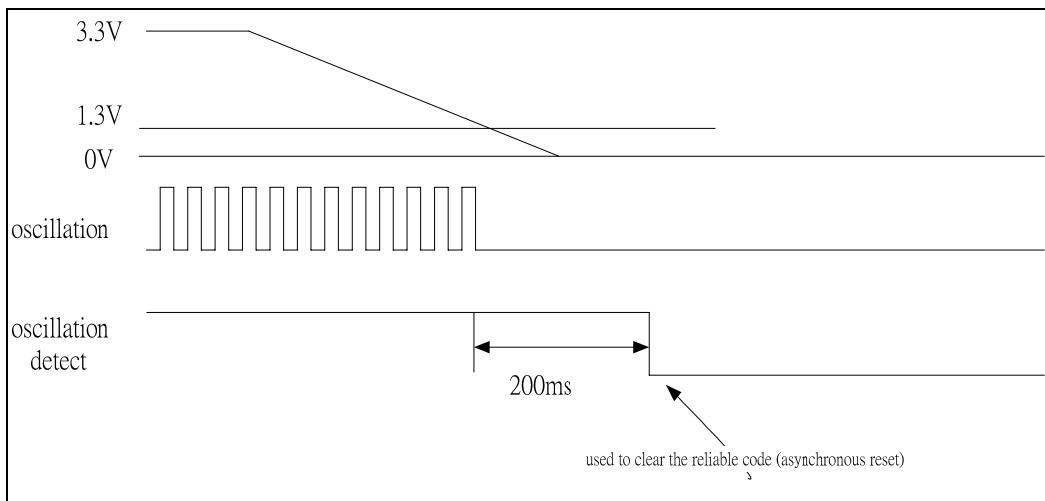


Fig. 9 RTC controller & RTC macro

The RTC macro has a ***reliable*** register, which tells the user if the time count in the RTC module is reliable. This register can be filled with any code from 0x00 to 0xff. If the RTC module losses power, the code is reset to 0x00 the next time the CPU read this register. The users should follow the above idea to ensure the RTC counter integrity. RTC working voltage range is between 3.3 ~ 1.3V.



40.2 Features

- Supports 48bit counter for time
- Down/up count
- Alarm function
- Independent power

40.3 Register Summary

Name	Address	Description
P_RTC_CTRL	0x9000B000	RTC Control
P_RTC_ADDR	0x9000B004	RTC Macro Register Address
P_RTC_WDATA	0x9000B008	RTC Macro Register Data
P_RTC_RWREQ	0x9000B00C	Trigger RTC Macro R/W
P_RTC_RDY	0x9000B010	Check RTC Macro Ready
P_RTC_RDATA	0x9000B014	RTC Macro Serial Data
P_RTC_TSEL	0x9000B028	For Test Probe Use
P_RTC_CLKDIV	0x9000B02C	RTC Serial Interface Clock Configure

40.4 Register Definition

P_RTC_CTRL								0x9000B000	RTC Control register	
Bit	7	6	5	4	3	2	1	0		
Function	-								RTCCEN	
Default	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
0	RTCCEN	R/W	RTC macro clock enable 0: disable 1: enable	-

P_RTC_ADDR								0x9000B004	RTC address register	
Bit	7	6	5	4	3	2	1	0		
Function	RTCADDR									
Default	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[7:0]	RTCADDR	R/W	RTC macro register address.	-

P_RTC_WDATA								0x9000B008	RTC data write register	
Bit	7	6	5	4	3	2	1	0		
Function	RTCWDATA									
Default	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[7:0]	RTCWDATA	R/W	RTC macro register data.	-

P_RTC_RWREQ								0x9000B00C	RTC WREQ register	
Bit	7	6	5	4	3	2	1	0		
Function	- RTERREQ RTEWREQ								RTERREQ	RTEWREQ
Default	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
1	RTERREQ	W	Trigger RTC macro read.	-
0	RTEWREQ	W	Trigger RTC macro write.	-

P_RTC_RDY								0x9000B010	RTC RDY register	
Bit	7	6	5	4	3	2	1	0		
Function									RTCRDY	
Default	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
0	RTCRDY	R	Check RTC macro ready	-

P_RTC_RDATA								0x9000B014	RTC data read register	
Bit	7	6	5	4	3	2	1	0		
Function									RTCRDATA	
Default	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[7:0]	RTCRDATA	R/W	RTC macro serial data	-

P_RTC_CLKDIV								0x9000B02C	RTC Control register	
Bit	7	6	5	4	3	2	1	0		
Function									RTCDIV	
Default	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[7:0]	RTCDIV	R/W	RTC serial clock divider	-

40.5 RTC Macro Internal Register

Address	Bit	Att	Name	Description	Initial Value
0x0	0	R/W	Rtcclken	RTC Clock Enable. To OSC Macro 0:Disable 1: Enable	1'b1
	1	R/W	Rtcrst	RTC Reset. To OSC Macro 0: normal ,1:Reset Analog Circuit	1'b0
	2	R/W	Timerdir	Timer Count Direction 1: up Count ,0: down Count	1'b1
	3	R/W	swreset	Swreset. 1: reset RTC_APB_Core	1'b0

Address	Bit	Att	Name	Description	Initial Value
0x0	4	R/W	Busy	When Write Address 0xc0 or 0xd0 or 0x10~0x15, need wait some time to make sure data be write in. 1: Busy State. When write 1 to this bit will clean this busy Statue. The behave is work in 32k domain. In normal case, please not write this bit.	1'b0
0x1	0	R/W	RRP	To OSC Macro RRP,RRN Signal. RRN = ~RRP. Enable OSC Enhance mode	1'b1
	1	R/W	Strong	The Output pin STRONG Value	
0x2	7:0	R/W	reliable	RTC Reliable Code. The RTC Values is unreliable when the read out value is differ from the Reliable Code. It's also effect wakeup interrupt. wire rtcwakeup = rtcwakeupen&(reliable)&alarmint;.	8'h0
0x3	0	R/W	PRST_SEL0	LVR Voltage Selection	1'b0
	1	R/W	PRST_SEL1	LVR Voltage Selection	1'b1
	2	R/W	Strong_clk_sel0	Output pin STRONG_CLK Value: 00: 2Hz 01: 4Hz 10: 8Hz 11: 16Hz	2'h0
	3	R/W	Strong_clk_sel0		
0x10	7:0	W	Loadcnt[7:0]	Load Value of Timer, only write only. If you want to check, please read 0xa0~0xa5 Timer Value	8'h0
0x11	7:0	W	Loadcnt[15:8]		8'h0
0x12	7:0	W	Loadcnt[23:16]		8'h0
0x13	7:0	W	Loadcnt[31:24]		8'h0
0x14	7:0	W	Loadcnt[39:32]		8'h0
0x15	7:0	W	Loadcnt[47:40]		8'h0
0x21	7:0	R/W	Alarmsdata[15:8]	Only [15:14] available	8'h0
0x22	7:0	R/W	Alarmsdata[23:16]		8'h0
0x23	7:0	R/W	Alarmsdata[31:24]		8'h0
0x24	7:0	R/W	Alarmsdata[39:32]		8'h0
0x25	7:0	R/W	Alarmsdata[47:40]		8'h0

Address	Bit	Att	Name	Description	Initial Value
0xa0	7:0	R	Timer[7:0]	The Timer Value	8'h0
0xa1	7:0	R	Timer[15:8]		8'h0
0xa2	7:0	R	Timer[23:16]		8'h0
0xa3	7:0	R	Timer[31:24]		8'h0
0xa4	7:0	R	Timer[39:32]		8'h0
0xa5	7:0	R	Timer[47:40]		8'h0
0xc0	0	R/W	Secint	The Interrupt of RTC reach to second. Write 0, the interrupt will clear.	
	1	R/W	Alarmint	The Interrupt of RTC Timer reach to the Alarndata Value. Write 0, the interrupt will clear.	
	2	R/W	rtcwakeups	The Interrupt of RTC Wakeup Interrupt. When RTC Timer reach to the Alarndata Value. The interrupt will clean by write Alarmint(0xco[1]) 0.	
0xd0	0	R/W	Secint_en	The Sec interrupt Enable. 1: enable 0: disable	
	1	R/W	Alarmint_en	The Alarm interrupt Enable. 1: enable. 0 : disable	
	2	R/W	Rtcwakeups_en	The Wakeup interrupt Enable. 1: enable. 0: disable	

41 I/O Ports

41.1 Introduction

The I/O ports are primarily designed to communicate with other devices. Each I/O pin is software-programmable. To enhance flexibility and functionality of the microprocessor, almost every I/O pin is assigned with four different types (1st~4th functions) by GID in SCU_B register P_SCUB_PGSx. And some I/O has another four different types (5th~8th functions) are assigned with register P_GPIO_PINMUX. In other words, many special function control signals are shared with I/O ports and controlled by register P_SCUB_PGS0, P_SCUB_PGS1, P_SCUB_PGS2, P_GPIO_PINMUX. For Example:

Pin	1 st function	2 nd function	3 rd function	4 th function	5 th function	6 th function	7 th function	8 th function	GID
IOA3	IOA3	SPI0_CS1	SD1_CMD		DSP_RTCK	PWM1	SENSOR1_D3	NAND_CS0	5
IOA4	IOA4	SPI0_CS0	SD1_D0		DSP_TCK		SENSOR1_D4	NAND_WE	
IOA5	IOA5	SPI0_TX	SD1_D1		DSP_TDO		SENSOR1_D5	NAND_RE	

All I/O ports are classified to four groups General-Purpose-Input-Output (GPIO) ports in GPL32900A: IOA, IOB, IOC, IOD, and can set the input enable or disable, driving strength, pull function enable or disable, and pullup or pulldown selection by register SCU_B 0x90005100~0x9000513C. If I/O set to input disable, GPL32900 can't get any data from this IO. If the I/O set to function GPIO (like IOAx) not other special function (like NAND_WE), programmers can controlled it to input or output mode, enable interrupt source, enable wakeup source...etc by setting register 0x9000A000~0x9000A0AC.



GPL32900A Programming Guide

The following table depicts shared information about I/O port and the special functions.

PortA Special Functions Shared Information

Pin No.	Name	1 st function	2 nd function	3 rd function	4 th function	5 th function	6 th function	7 th function	8 th function	ID
185	IOA0	IOA0	-	-	-	-	-	SENSOR1_D0	-	18
184	IOA1	IOA1	-	-	-	-	-	SENSOR1_D1	-	19
183	IOA2	IOA2	-	SD1_CLK	-	-	-	SENSOR1_D2	-	20
182	IOA3	IOA3	SPI0_CS1	SD1_CMD	-	DSP_RTCK	PWM1	SENSOR1_D3	NAND_CS0	21
181	IOA4	IOA4	SPI0_CS0	SD1_D0	-	DSP_TCK	-	SENSOR1_D4	NAND_WE	22
180	IOA5	IOA5	SPI0_TX	SD1_D1	-	DSP_TDO	-	SENSOR1_D5	NAND_RE	
179	IOA6	IOA6	SPI0_RX	SD1_D2	-	DSP_TDI	-	SENSOR1_D6	NAND_CLE	23
178	IOA7	IOA7	SPI0_CLK	SD1_D3	-	DSP_TMS	-	SENSOR1_D7	NAND_ALE	24
68	IOA10	-	SENSOR0_MCLK_OUT	IOA10	SPI0_CLK	MIPI_MCLK_OUT	-	-	-	0
67	IOA11	-	SENSOR0_PCLK_IN	IOA11	IOA11	-	-	-	-	
66	IOA12	-	SENSOR0_VSYNC	IOA12	SPI0_TX	-	-	-	-	1
65	IOA13	-	SENSOR0_HSYNC	IOA13	SPI0_RX	-	-	-	-	2
166	IOA15	IOA15	-	Note1	-	-	-	-	-	7
24	IOA16	TP_XP	-	-	-	-	-	-	-	-
23	IOA17	TP_XN	-	-	-	-	-	-	-	
25	IOA18	TP_YP	-	-	-	-	-	-	-	
22	IOA19	TP_YN	-	-	-	-	-	-	-	
165	IOA20	NAND_CS1	-	IOA20	-	-	-	-	-	26
188	IOA23	SD0_CLK	MS_CLK	IOA23	-	-	-	-	-	29
189	IOA24	SD0_CMD	MS_BS	IOA24	-	-	-	-	-	30



GPL32900A Programming Guide

Pin No.	Name	1 st function	2 nd function	3 rd function	4 th function	5 th function	6 th function	7 th function	8 th function	ID
187	IOA25	SD0_D0	MS_D0	IOA25	-	-	-	-	-	
186	IOA26	SD0_D1	MS_D3	IOA26	-	-	-	-	-	34
191	IOA27	SD0_D2	MS_D2	IOA27	-	-	-	-	-	
190	IOA28	SD0_D3	MS_D1	IOA28	-	-	-	-	-	
174	IOA29	UART0_TX	-	IOA29	-	ARM_TMS	I2S_MCLK	SENSOR1_HSYNC	SPI1_TX	31
175	IOA30	UART0_RX	-	IOA30	-	ARM_TDI	I2S_SLCK	SENSOR1_VSYNC	SPI1_RX	32

Note1: If setting function 3, the output could be output 1 or output2....., it depend on the LCD controller

pin	Output1	Output2	Output3	Output4
IOC0	LCD_DE	LCD_DE	LCM_RD	TCON_OEV



GPL32900A Programming Guide

PortB Special Functions Shared Information

Pin No.	Name	1 st function	2 nd function	3 rd function	4 th function	5 th function	6 th function	7 th function	8 th function	ID
38	IOB0	IOB0	-	Note2	-	SENSOR0_D0	MIP1_MCLK_OUT	-	-	8
37	IOB1	IOB1	-	Note2	-	SENSOR0_D1	-	LCM_TE	-	
36	IOB2	IOB2	-	Note2	-	SENSOR0_D2	-	-	-	
35	IOB3	IOB3	-	Note2	-	SENSOR0_D3	-	-	-	
34	IOB4	IOB4	-	Note2	-	SENSOR0_D4	-	-	-	
33	IOB5	IOB5	-	Note2	-	SENSOR0_D5	-	-	-	
32	IOB6	IOB6	-	Note2	-	SENSOR0_D6	-	-	-	
31	IOB7	IOB7	-	Note2	-	SENSOR0_D7	-	-	-	
176	IOB16	I2C0_SDA	LCD_R0	-	IOB16	ARM_TDO	I2S_LRCLK	PWM2	SPI1_CLK	13
177	IOB17	I2C0_SCL	LCD_B0	-	IOB17	ARM_TCK	I2S_SDO	-	-	
173	IOB18	UART1_TX	-	IOB18	-	DSUB_HSYNC	I2C0_SDA	SENSOR1_MCLK_OUT	-	33
172	IOB19	UART1_RX	-	IOB19	-	DSUB_VSYNC	I2C0_SCL	SENSOR1_PCLK_IN	-	

Note2: If setting function 3, the output could be output 1 or , it depend on the LCD controller

Pin	Output1	Output2	Output3
IOB0	LCD_G0	LCD_G3	LCM_D8
IOB1	LCD_G1	LCD_G4	LCM_D9
IOB2	LCD_G2	LCD_G5	LCM_D10
IOB3	LCD_G3	LCD_R1	LCM_D11
IOB4	LCD_G4	LCD_R2	LCM_D12
IOB5	LCD_G5	LCD_R3	LCM_D13
IOB6	LCD_G6	LCD_R4	LCM_D14
IOB7	LCD_G7	LCD_R5	LCM_D15



GPL32900A Programming Guide

PortC Special Functions Shared Information

Pin No.	Name	1 st function	2 nd function	3 rd function	4 th function	5 th function	6 th function	7 th function	8 th function	ID
64	IOC0	Note3	-	IOC0	-	-	-	-	-	4
63	IOC1	Note3	-	IOC1	-	-	-	-	-	
62	IOC2	Note3	-	IOC2	-	-	-	-	-	
61	IOC3	Note3	-	IOC3	-	-	-	-	-	
60	IOC4	Note3	-	IOC4	-	-	-	-	-	
59	IOC5	Note3	-	IOC5	-	-	-	-	-	
58	IOC6	Note3	-	IOC6	-	-	-	-	-	
57	IOC7	Note3	-	IOC7	-	-	-	-	-	
30	IOC8	IOC8	-	Note4	-	-	-	-	-	9
29	IOC9	IOC9	-	Note4	-	-	-	-	-	
52	IOC10	IOC10	-	Note4	-	TCON_STHR/L	-	-	-	
51	IOC11	IOC11	-	Note4	-	TCON_STVU/D	-	-	-	
50	IOC12	IOC12	-	LCD_R4	-	TCON_POL	-	-	-	10
49	IOC13	IOC13	-	LCD_R5	-	-	-	-	-	
48	IOC14	IOC14	-	LCD_R6	-	-	-	-	-	
47	IOC15	IOC15	-	LCD_R7	-	-	-	-	-	
77	IOC24	IOC24	-	-	-	MIPI_DATA0N	SENSOR0_HSYNC	SPI1_TX	-	
76	IOC25	IOC25	-	-	-	MIPI_DATA0P	SENSOR0_VSYNC	SPI1_RX	-	-
75	IOC26	IOC26	-	-	-	MIPI_CLKN	SENSOR0_PCLK_IN	SPI1_CLK	-	
74	IOC27	IOC27	-	-	-	MIPI_CLKP	SENSOR0_MCLK_OUT	SPI1_CS0	-	
73	IOC28	IOC28	-	-	-	MIPI_DATA1N	SENSOR0_D1	SPI1_CS1	-	
72	IOC29	IOC29	-	-	-	MIPI_DATA1P	SENSOR0_D0	LCM_TE	-	



GPL32900A Programming Guide

Note 3: If setting function 1, the output could be output 1 or output2, it depend on the Sensor controller

Pin	Output1	Output2
IOC0	SENSOR0_D0	SENSOR0_D2
IOC1	SENSOR0_D1	SENSOR0_D3
IOC2	SENSOR0_D2	SENSOR0_D4
IOC3	SENSOR0_D3	SENSOR0_D5
IOC4	SENSOR0_D4	SENSOR0_D6
IOC5	SENSOR0_D5	SENSOR0_D7
IOC6	SENSOR0_D6	SENSOR0_D8
IOC7	SENSOR0_D7	SENSOR0_D9

Note4: If setting function 3, the output could be output 1 or output2, it depend on the LCD controller

Pin	Output1	Output2
IOC8	LCD_R0	LCD_B0
IOC9	LCD_R1	LCD_R0
IOC10	LCD_R2	LCM_D16
IOC11	LCD_R3	LCM_D17



GPL32900A Programming Guide

PortD Special Functions Shared Information

Pin No.	Name	1 st function	2 nd function	3 rd function	4 th function	5 th function	6 th function	7 th function	8 th function	ID
56	IOD0	IOD0	-	Note5	-	-	-	-	-	6
168	IOD1	IOD1	-	Note5	-	-	-	-	-	
167	IOD2	IOD2	-	Note5	-	-	-	-	-	
46	IOD3	IOD3	-	Note5	-	-	-	-	-	
45	IOD4	IOD4	-	Note5	-	-	-	-	-	
44	IOD5	IOD5	-	Note5	-	-	-	-	-	
43	IOD6	IOD6	-	Note5	-	-	-	-	-	
42	IOD7	IOD7	-	Note5	-	-	-	-	-	
41	IOD8	IOD8	-	Note5	-	-	-	-	-	
40	IOD9	IOD9	-	Note5	-	-	-	-	-	
39	IOD10	IOD10	-	Note5	-	-	-	-	-	25
151	IOD11	NAND_D0	-	IOD11	-	-	-	-	-	
152	IOD12	NAND_D1	-	IOD12	-	-	-	-	-	
153	IOD13	NAND_D2	-	IOD13	-	-	-	-	-	
154	IOD14	NAND_D3	-	IOD14	-	-	-	-	-	
155	IOD15	NAND_D4	-	IOD15	-	-	-	-	-	
156	IOD16	NAND_D5	-	IOD16	-	-	-	-	-	
157	IOD17	NAND_D6	-	IOD17	-	-	-	-	-	
158	IOD18	NAND_D7	-	IOD18	-	-	-	-	-	
159	IOD19	NAND_ALE	-	IOD19	-	-	-	-	-	



GPL32900A Programming Guide

Pin No.	Name	1 st function	2 nd function	3 rd function	4 th function	5 th function	6 th function	7 th function	8 th function	ID
160	IOD20	NAND_CLE	-	IOD20	-	-	-	-	-	-
161	IOD21	NAND_RE	-	IOD21	-	-	-	-	-	
162	IOD22	NAND_WE	IOD22	IOD22	-	-	-	-	-	
163	IOD23	NAND_CS0	IOD23	IOD23	-	-	-	-	-	
164	IOD24	NAND_RDY0	IOD24	IOD24	-	-	-	-	-	
198	IOD25	PWM1_PWM	IOD25	-	-	-	-	-	-	
196	IOD26	PWM1_FB	IOD26	-	-	-	-	-	-	
195	IOD27	PWM1_VC	IOD27	-	-	-	-	-	-	

Note5: If setting function 3, the output could be output 1 or output2 or output3 or output4, it depend on the LCD controller

Pin	Output1	Output2	Output3	Output4
IOD0	LCD_CLK	LCD_CLK	LCM_CS	TCON_DCLK
IOD1	LCD_VSYNC	LCD_VSYNC	LCM_RS	TCON_CKV
IOD2	LCD_HSYNC	LCD_HSYNC	LCM_WR	TCON_OEH
IOD3	LCD_B0	LCD_B1	LCM_D0	-
IOD4	LCD_B1	LCD_B2	LCM_D1	-
IOD5	LCD_B2	LCD_B3	LCM_D2	-
IOD6	LCD_B3	LCD_B4	LCM_D3	-
IOD7	LCD_B4	LCD_B5	LCM_D4	-
IOD8	LCD_B5	LCD_G0	LCM_D5	-
IOD9	LCD_B6	LCD_G1	LCM_D6	-
IOD10	LCD_B7	LCD_G2	LCM_D7	-



GPL32900A Programming Guide

SDRAM bus Special Functions Shared Information

Pin No.	Name	1 st function	2 nd function	ID
118	DRAM_A0	DRAM_A0	NAND_D0	-
117	DRAM_A1	DRAM_A1	NAND_D1	
116	DRAM_A2	DRAM_A2	NAND_D2	
115	DRAM_A3	DRAM_A3	NAND_D3	
109	DRAM_A4	DRAM_A4	NAND_D4	
108	DRAM_A5	DRAM_A5	NAND_D5	
107	DRAM_A6	DRAM_A6	NAND_D6	
105	DRAM_A7	DRAM_A7	NAND_D7	
104	DRAM_A8	DRAM_A8	NAND_ALE	
102	DRAM_A9	DRAM_A9	NAND_CLE	
120	DRAM_A10	DRAM_A10	NAND_RE	
99	DRAM_A11	DRAM_A11	NAND_WE	

The following table depicts how to select different functions by GID:

	[19:18]	[17:16]	[15:14]	[13:12]	[11:10]	[9:8]	[7:6]	[5:4]	[3:2]	[1:0]
PGS0	GID9	GID8	GID7	GID6	GID5	GID4	GID3	GID2	GID1	GID0
PGS1	GID25	GID24	GID23	GID22	GID21	GID20	GID19	GID18	GID17	GID16
PGS2	-	-	-	-	-	-	-	GID34	GID33	GID32

	-	-	-	-	[31:30]	[29:28]	[27:26]	[25:24]	[23:22]	[21:20]
PGS0	-	-	-	-	GID15	GID14	GID13	GID12	GID11	GID10
PGS1	-	-	-	-	GID31	GID30	GID29	GID28	GID27	GID26
PGS2	-	-	-	-	-	-	-	-	-	-

41.2 Register Summary

SCU_B module related register

Name	Address	Description
P_SCUB_PGS0	0x90005080	Pad group selection, for more function descript, please refer to the table in sector introduction.
P_SCUB_PGS1	0x90005084	Pad group selection, for more function descript, please refer to the table in sector introduction.
P_SCUB_PGS2	0x90005088	Pad group selection, for more function descript, please refer to the table in sector introduction.
P_IOA_INPUTEN	0x90005100	IOA input enable
P_IOA_DRIVE	0x90005104	IOA drive strength
P_IOA_PULLEN	0x90005108	IOA pull enable
P_IOA_PULLSEL	0x9000510C	IOA pull select
P_IOB_INPUTEN	0x90005100	IOB input enable
P_IOB_DRIVE	0x90005104	IOB drive strength
P_IOB_PULLEN	0x90005108	IOB pull enable
P_IOB_PULLSEL	0x9000510C	IOB pull select
P_IOC_INPUTEN	0x90005100	IOC input enable
P_IOC_DRIVE	0x90005104	IOC drive strength
P_IOC_PULLEN	0x90005108	IOC pull enable
P_IOC_PULLSEL	0x9000510C	IOC pull select
P_IOD_INPUTEN	0x90005100	IOD input enable
P_IOD_DRIVE	0x90005104	IOD drive strength
P_IOD_PULLEN	0x90005108	IOD pull enable
P_IOD_PULLSEL	0x9000510C	IOD pull select
P_GPIO_PINMUX	0x90005144	Pinmux for some pads

GPIO controller related register

Name	Address	Description
P_IOA_ENABLE	0x9000A000	Select the IOA port to GPIO function or other function
P_IOB_ENABLE	0x9000A004	Select the IOB port to GPIO function or other function
P_IOC_ENABLE	0x9000A008	Select the IOC port to GPIO function or other function
P_IOD_ENABLE	0x9000A00C	Select the IOD port to GPIO function or other function
P_IOA_O_DATA	0x9000A010	Output data in IOA port
P_IOB_O_DATA	0x9000A014	Output data in IOB port
P_IOC_O_DATA	0x9000A018	Output data in IOC port
P_IOD_O_DATA	0x9000A01C	Output data in IOD port
P_IOA_DIR	0x9000A020	Select IOA port to input mode or output mode
P_IOB_DIR	0x9000A024	Select IOB port to input mode or output mode
P_IOC_DIR	0x9000A028	Select IOC port to input mode or output mode
P_IOD_DIR	0x9000A02C	Select IOD port to input mode or output mode
P_IOA_POL	0x9000A030	Select the polarity active low or active high for interrupt
P_IOB_POL	0x9000A034	Select the polarity active low or active high for interrupt
P_IOC_POL	0x9000A038	Select the polarity active low or active high for interrupt
P_IOD_POL	0x9000A03C	Select the polarity active low or active high for interrupt
P_IOA_STICKY	0x9000A040	Select Direct output or clocked rising edge after polarity change
P_IOB_STICKY	0x9000A044	Select Direct output or clocked rising edge after polarity change
P_IOC_STICKY	0x9000A048	Select Direct output or clocked rising edge after polarity change
P_IOD_STICKY	0x9000A04C	Select Direct output or clocked rising edge after polarity change
P_IOA_INTEN	0x9000A050	Enable interrupt by trigger IOA port
P_IOB_INTEN	0x9000A054	Enable interrupt by trigger IOB port
P_IOC_INTEN	0x9000A058	Enable interrupt by trigger IOC port
P_IOD_INTEN	0x9000A05C	Enable interrupt by trigger IOD port
P_IOA_INTPEND	0x9000A060	Interrupt pending signal on IOA port
P_IOB_INTPEND	0x9000A064	Interrupt pending signal on IOB port
P_IOC_INTPEND	0x9000A068	Interrupt pending signal on IOC port
P_IOD_INTPEND	0x9000A06C	Interrupt pending signal on IOD port
P_IOA_DEBOUNCE_CNT	0x9000A070	Debounce counter comparison
P_IOB_DEBOUNCE_CNT	0x9000A074	Debounce counter comparison

Name	Address	Description
P_IOC_DEBOUNCE_CNT	0x9000A078	Debounce counter comparison
P_IOD_DEBOUNCE_CNT	0x9000A07C	Debounce counter comparison
P_IOA_DEBOUNCE_EN	0x9000A080	Enable debounce
P_IOB_DEBOUNCE_EN	0x9000A084	Enable debounce
P_IOC_DEBOUNCE_EN	0x9000A088	Enable debounce
P_IOD_DEBOUNCE_EN	0x9000A08C	Enable debounce
P_IOA_WKUP_EN	0x9000A090	Enable Wake up from IOA port
P_IOB_WKUP_EN	0x9000A094	Enable Wake up from IOB port
P_IOC_WKUP_EN	0x9000A098	Enable Wake up from IOC port
P_IOD_WKUP_EN	0x9000A09C	Enable Wake up from IOD port
P_IOA_STATUS	0x9000A0A0	IOA port Status
P_IOB_STATUS	0x9000A0A4	IOB port Status
P_IOC_STATUS	0x9000A0A8	IOC port Status
P_IOD_STATUS	0x9000A0AC	IOD port Status

41.3 Register Definition

P_SCUB_PGS0																Pad group selection																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																												
Function	GID15				GID14				GID13				GID12				GID11				GID10				GID9				GID8															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																												
Function	GID7				GID6				GID5				GID4				GID3				GID2				GID1				GID0															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																												

Bit	Function	Type	Description	Condition
[31:30]	GID15	R/W	Pad group selection, for more detail please refer to the table depicts in section introduction.	00: select 1 st function 01: select 2 nd function 10: select 3 rd function 11: select 4 th function
[29:28]	GID14	R/W	Ref. GID15	Ref. GID15
[27:26]	GID13	R/W	Ref. GID15	Ref. GID15
[25:24]	GID12	R/W	Ref. GID15	Ref. GID15
[23:22]	GID11	R/W	Ref. GID15	Ref. GID15
[21:20]	GID10	R/W	Ref. GID15	Ref. GID15
[19:18]	GID9	R/W	Ref. GID15	Ref. GID15
[17:16]	GID8	R/W	Ref. GID15	Ref. GID15

Bit	Function	Type	Description	Condition
[15:14]	GID7	R/W	Ref. GID15	Ref. GID15
[13:12]	GID6	R/W	Ref. GID15	Ref. GID15
[11:10]	GID5	R/W	Ref. GID15	Ref. GID15
[9:8]	GID4	R/W	Ref. GID15	Ref. GID15
[7:6]	GID3	R/W	Ref. GID15	Ref. GID15
[5:4]	GID2	R/W	Ref. GID15	Ref. GID15
[3:2]	GID1	R/W	Ref. GID15	Ref. GID15
[1:0]	GID0	R/W	Ref. GID15	Ref. GID15

P_SCUB PGS1 **0x90005084** **Pad group selection**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	GID31		GID30		GID29		GID28		GID27		GID26		GID25		GID24	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	GID23		GID22		GID21		GID20		GID19		GID18		GID17		GID16	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:30]	GID31	R/W	Pad group selection, for more detail please refer to the table depicts in section introduction.	00: select 1 st function 01: select 2 nd function 10: select 3 rd function 11: select 4 th function
[29:28]	GID30	R/W	Ref. GID31	Ref. GID31
[27:26]	GID29	R/W	Ref. GID31	Ref. GID31
[25:24]	GID28	R/W	Ref. GID31	Ref. GID31
[23:22]	GID27	R/W	Ref. GID31	Ref. GID31
[21:20]	GID26	R/W	Ref. GID31	Ref. GID31
[19:18]	GID25	R/W	Ref. GID31	Ref. GID31
[17:16]	GID24	R/W	Ref. GID31	Ref. GID31
[15:14]	GID23	R/W	Ref. GID31	Ref. GID31
[13:12]	GID22	R/W	Ref. GID31	Ref. GID31
[11:10]	GID21	R/W	Ref. GID31	Ref. GID31
[9:8]	GID20	R/W	Ref. GID31	Ref. GID31
[7:6]	GID19	R/W	Ref. GID31	Ref. GID31
[5:4]	GID18	R/W	Ref. GID31	Ref. GID31
[3:2]	GID17	R/W	Ref. GID31	Ref. GID31
[1:0]	GID16	R/W	Ref. GID31	Ref. GID31

P_SCUB_PGS2

0x90005088

Pad group selection

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	GID34	GID33	GID32			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description												Condition
[5:4]	GID34	R/W	Pad group selection, for more detail please refer to the table depicts in section introduction.												00: select 1 st function 01: select 2 nd function 10: select 3 rd function 11: select 4 th function
[3:2]	GID33	R/W	Ref. GID34												Ref. GID34
[1:0]	GID32	R/W	Ref. GID34												Ref. GID34

P_IOA_INPUTEN

0x90005100

IOA port input enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	IOA_INPUTEN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOA_INPUTEN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description												Condition
[31:0]	IOA_INPUTEN	R/W	IOA port input enable												0: disable 1: enable

P_IOA_DRIVE

0x90005104

IOA driving strength

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	IOA_DRIVE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOA_DRIVE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	IOA_DRIVE	R/W	IOA driving strength	0: 4mA 1: 8mA

P_IOA_PULLEN																0x90005108 IOA port pull up or pull down enable															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16															
Function	IOA_PULLEN																														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
Function	IOA_PULLEN																														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															

Bit	Function	Type	Description	Condition
[31:0]	IOA_PULLEN	R/W	IOA port pull up or pull down enable	0: disable 1: enable

P_IOA_PULLSEL																0x9000510C IOA port pull up or pull down selection															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16															
Function	IOA_PULLSEL																														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
Function	IOA_PULLSEL																														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															

Bit	Function	Type	Description	Condition
[31:0]	IOA_PULLSEL	R/W	IOA port pull up or pull down selection	0: pull down 1: pull up

P_IOB_INPUTEN																0x90005110 IOB port input enable															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16															
Function	IOB_INPUTEN																														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
Function	IOB_INPUTEN																														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															

Bit	Function	Type	Description										Condition
[31:0]	IOB_INPUTEN	R/W	IOB port input enable										0: disable 1: enable

P_IOB_DRIVE																
0x90005114																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	IOB_DRIVE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOB_DRIVE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description										Condition
[31:0]	IOB_DRIVE	R/W	IOB driving strength										0: 4mA 1: 8mA

P_IOB_PULLEN																
0x90005118																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	IOB_PULLEN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOB_PULLEN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description										Condition
[31:0]	IOB_PULLEN	R/W	IOB port pull up or pull down enable										0: disable 1: enable

P_IOB_PULLSEL																
0x9000511C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	IOB_PULLSEL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOB_PULLSEL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:0]	IOB_PULLSEL	R/W	IOB port pull up or pull down selection	0: pull down 1: pull up

P_IOC_INPUTEN																0x90005120																IOC port input enable															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOC_INPUTEN																Function	IOC_INPUTEN																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOC_INPUTEN																Function	IOC_INPUTEN																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

Bit	Function	Type	Description	Condition
[31:0]	IOC_INPUTEN	R/W	IOC port input enable	0: disable 1: enable

P_IOC_DRIVE																0x90005124																IOC driving strength															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOC_DRIVE																Function	IOC_DRIVE																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOC_DRIVE																Function	IOC_DRIVE																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

Bit	Function	Type	Description	Condition
[31:0]	IOC_DRIVE	R/W	IOC driving strength	0: 4mA 1: 8mA

P_IOC_PULLEN																0x90005128																IOC port pull up or pull down enable															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOC_PULLEN																Function	IOC_PULLEN																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOC_PULLEN																Function	IOC_PULLEN																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																



Bit	Function	Type	Description	Condition
[31:0]	IOC_PULLEN	R/W	IOC port pull up or pull down enable	0: disable 1: enable

Bit	Function	Type	Description	Condition
[31:0]	IOC_PULLSEL	R/W	IOC port pull up or pull down selection	0: pull down 1: pull up

Bit	Function	Type	Description	Condition
[31:0]	IOD_INPUTEN	R/W	IOD port input enable	0: disable 1: enable

Bit	Function	Type	Description	Condition
[31:0]	IOD_DRIVE	R/W	IOD driving strength	0: 4mA 1: 8mA

P_IOD_PULLEN																0x90005138																IOD port pull up or pull down enable															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOD_PULLEN																Function	IOD_PULLEN																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOD_PULLEN																Function	IOD_PULLEN																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														

Bit	Function	Type	Description	Condition
[31:0]	IOD_PULLEN	R/W	IOD port pull up or pull down enable	0: disable 1: enable

P_IOD_PULLSEL																0x9000513C																IOD port pull up or pull down selection															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOD_PULLSEL																Function	IOD_PULLSEL																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Function	IOD_PULLSEL																Function	IOD_PULLSEL																													
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														

Bit	Function	Type	Description	Condition
[31:0]	IOD_PULLSEL	R/W	IOD port pull up or pull down selection	0: pull down 1: pull up

P_GPIO_PINMUX																0x90005144																Pinmux for some pads																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit	23	22	21	20	19	18	17	16	Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
Function	SPI2	SPICS0	SPICS1	LCMTE2	CDSRAW2	MIPICLK2	MIPICLK1	CSICTRL	-	-	SD1CMD	SD0CMD	SHARENAND	CDSRAW1	PWM2	PWM1	Function	-	-	SD1CMD	SD0CMD	SHARENAND	CDSRAW1	PWM2	PWM1	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	I2S	NANDCS	NANDCTRL	CSICH	SPI1	LCMTE1	NANDRDY	PPUI80
Default	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0

Bit	7	6	5	4	3	2	1	0
Function	I2C	LCDTCON	CSIFIELD	NAND1CS	NAND0CS	CSIDAT	CEVAJTAG	ARMJTAG
Default	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0

Bit	Function	Type	Description	Condition										
31	SPI2	R/W	IOC24~IOC26 is as SPI1 signal <table border="1"> <tr> <td>Pin name</td> <td>function</td> </tr> <tr> <td>IOC24</td> <td>SPI1_TX</td> </tr> <tr> <td>IOC25</td> <td>SPI1_RX</td> </tr> <tr> <td>IOC26</td> <td>SPI1_CLK</td> </tr> </table>	Pin name	function	IOC24	SPI1_TX	IOC25	SPI1_RX	IOC26	SPI1_CLK	0: original function(set in PSG) 1: Map to new function		
Pin name	function													
IOC24	SPI1_TX													
IOC25	SPI1_RX													
IOC26	SPI1_CLK													
30	SPICS0	R/W	IOC27 is SPI1_CS0 <table border="1"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOC27</td> <td>SPI1_CS0</td> </tr> </table>	PAD name	New function	IOC27	SPI1_CS0	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOC27	SPI1_CS0													
29	SPICS1	R/W	IOC28 is SPI1_CSI1 <table border="1"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOC28</td> <td>SPI1_CS1</td> </tr> </table>	PAD name	New function	IOC28	SPI1_CS1	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOC28	SPI1_CS1													
28	LCMTE2	R/W	IOC29 is LCM_TE <table border="1"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOC29</td> <td>LCM_TE</td> </tr> </table>	PAD name	New function	IOC29	LCM_TE	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOC29	LCM_TE													
27	CDSPRAW2	R/W	IOC28, IOC29 for CDSP raw data bit [1:0] <table border="1"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOC28</td> <td>SENSOR0_D0</td> </tr> <tr> <td>IOC29</td> <td>SENSOR0_D1</td> </tr> </table>	PAD name	New function	IOC28	SENSOR0_D0	IOC29	SENSOR0_D1	0: original function(set in PSG) 1: Map to new function				
PAD name	New function													
IOC28	SENSOR0_D0													
IOC29	SENSOR0_D1													
26	MIPICLK2	R/W	IOA10 is MIPI_MCLK_OUT <table border="1"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOA10</td> <td>MIPI_MCLK_OUT</td> </tr> </table>	PAD name	New function	IOA10	MIPI_MCLK_OUT	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOA10	MIPI_MCLK_OUT													
25	MIPICLK1	R/W	IOB0 is MIPI_MCLK_OUT <table border="1"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOB0</td> <td>MIPI_MCLK_OUT</td> </tr> </table>	PAD name	New function	IOB0	MIPI_MCLK_OUT	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOB0	MIPI_MCLK_OUT													
24	CSICTRL	R/W	IOC24~27 is Sensor control signal <table border="1"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOC24</td> <td>SENSOR0_HSYNC</td> </tr> <tr> <td>IOC25</td> <td>SENSOR0_VSYNC</td> </tr> <tr> <td>IOC26</td> <td>SENSOR0_PCLK_IN</td> </tr> <tr> <td>IOC27</td> <td>SENSOR0_MCLK_OUT</td> </tr> </table>	PAD name	New function	IOC24	SENSOR0_HSYNC	IOC25	SENSOR0_VSYNC	IOC26	SENSOR0_PCLK_IN	IOC27	SENSOR0_MCLK_OUT	0: original function(set in PSG) 1: Map to new function
PAD name	New function													
IOC24	SENSOR0_HSYNC													
IOC25	SENSOR0_VSYNC													
IOC26	SENSOR0_PCLK_IN													
IOC27	SENSOR0_MCLK_OUT													

Bit	Function	Type	Description	Condition										
[23:22]	Reserved		Reserved											
21	SD1CMD	R/W	Sdcard interface 1 working but IOA3 (SD1_CMD) can be GPIO <table border="1" data-bbox="504 444 938 534"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOA3</td> <td>SD1_CMD</td> </tr> </table>	PAD name	New function	IOA3	SD1_CMD	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOA3	SD1_CMD													
20	SD0CMD	R/W	Sdcard interface 0 working but IOA24 (SD0_CMD) can be GPIO <table border="1" data-bbox="504 631 938 720"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOA24</td> <td>SD1_CMD</td> </tr> </table>	PAD name	New function	IOA24	SD1_CMD	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOA24	SD1_CMD													
19	SHARENAND	R/W	Nand data pin share with SDRAM. But control pin use the original	0: original function(set in PSG) 1: Map to new function										
18	CDSPRAW1	R/W	IOA0~1 for CDSP raw data bit [1:0] <table border="1" data-bbox="504 862 938 990"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOA0</td> <td>SENSOR0_D0</td> </tr> <tr> <td>IOA1</td> <td>SENSOR0_D1</td> </tr> </table>	PAD name	New function	IOA0	SENSOR0_D0	IOA1	SENSOR0_D1	0: original function(set in PSG) 1: Map to new function				
PAD name	New function													
IOA0	SENSOR0_D0													
IOA1	SENSOR0_D1													
17	PWM2	R/W	IOB16 is PWM2 <table border="1" data-bbox="504 1057 938 1136"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOB16</td> <td>PWM2</td> </tr> </table>	PAD name	New function	IOB16	PWM2	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOB16	PWM2													
16	PWM1	R/W	IOA3 is PWM1 <table border="1" data-bbox="504 1203 938 1282"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOA3</td> <td>PWM1</td> </tr> </table>	PAD name	New function	IOA3	PWM1	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOA3	PWM1													
15	I2S	R/W	IOA29, 30 & IOB16, 17 is I2s function: <table border="1" data-bbox="504 1349 938 1545"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOA29</td> <td>I2S_MCLK</td> </tr> <tr> <td>IOA30</td> <td>I2S_SLCK</td> </tr> <tr> <td>IOB16</td> <td>I2S_LRCLK</td> </tr> <tr> <td>IOB17</td> <td>I2S_SDO</td> </tr> </table>	PAD name	New function	IOA29	I2S_MCLK	IOA30	I2S_SLCK	IOB16	I2S_LRCLK	IOB17	I2S_SDO	0: original function(set in PSG) 1: Map to new function
PAD name	New function													
IOA29	I2S_MCLK													
IOA30	I2S_SLCK													
IOB16	I2S_LRCLK													
IOB17	I2S_SDO													
14	NANDCS	R/W	B_KEYSCAN3 is NAND_CS0 <table border="1" data-bbox="504 1612 938 1691"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOA3</td> <td>NAND_CS0</td> </tr> </table>	PAD name	New function	IOA3	NAND_CS0	0: original function(set in PSG) 1: Map to new function						
PAD name	New function													
IOA3	NAND_CS0													
13	NANDCTRL	R/W	IOA4~7 is nand ctrl: <table border="1" data-bbox="504 1758 938 1951"> <tr> <td>PAD name</td> <td>New function</td> </tr> <tr> <td>IOA4</td> <td>NAND_WE</td> </tr> <tr> <td>IOA5</td> <td>NAND_RE</td> </tr> <tr> <td>IOA6</td> <td>NAND_CLE</td> </tr> <tr> <td>IOA7</td> <td>NAND_ALE</td> </tr> </table>	PAD name	New function	IOA4	NAND_WE	IOA5	NAND_RE	IOA6	NAND_CLE	IOA7	NAND_ALE	0: original function(set in PSG) 1: Map to new function
PAD name	New function													
IOA4	NAND_WE													
IOA5	NAND_RE													
IOA6	NAND_CLE													
IOA7	NAND_ALE													

Bit	Function	Type	Description	Condition																						
12	CSICH	R/W	IOA0~7, IOA29~30 is Sensor signal <table border="1"> <tr><td>PAD name</td><td>New function</td></tr> <tr><td>IOA0</td><td>SENSOR1_D0</td></tr> <tr><td>IOA1</td><td>SENSOR1_D1</td></tr> <tr><td>IOA2</td><td>SENSOR1_D2</td></tr> <tr><td>IOA3</td><td>SENSOR1_D3</td></tr> <tr><td>IOA4</td><td>SENSOR1_D4</td></tr> <tr><td>IOA5</td><td>SENSOR1_D5</td></tr> <tr><td>IOA6</td><td>SENSOR1_D6</td></tr> <tr><td>IOA7</td><td>SENSOR1_D7</td></tr> <tr><td>IOA29</td><td>SENSOR1_HSYNC</td></tr> <tr><td>IOA30</td><td>SENSOR1_VSYNC</td></tr> </table>	PAD name	New function	IOA0	SENSOR1_D0	IOA1	SENSOR1_D1	IOA2	SENSOR1_D2	IOA3	SENSOR1_D3	IOA4	SENSOR1_D4	IOA5	SENSOR1_D5	IOA6	SENSOR1_D6	IOA7	SENSOR1_D7	IOA29	SENSOR1_HSYNC	IOA30	SENSOR1_VSYNC	0: original function(set in PSG) 1: Map to new function
PAD name	New function																									
IOA0	SENSOR1_D0																									
IOA1	SENSOR1_D1																									
IOA2	SENSOR1_D2																									
IOA3	SENSOR1_D3																									
IOA4	SENSOR1_D4																									
IOA5	SENSOR1_D5																									
IOA6	SENSOR1_D6																									
IOA7	SENSOR1_D7																									
IOA29	SENSOR1_HSYNC																									
IOA30	SENSOR1_VSYNC																									
11	SPI1	R/W	IOB16, IOA29~30 is SPI signal <table border="1"> <tr><td>PAD name</td><td>New function</td></tr> <tr><td>IOA29</td><td>SPI1_TX</td></tr> <tr><td>IOA30</td><td>SPI1_RX</td></tr> <tr><td>IOB16</td><td>SPI1_CLK</td></tr> </table>	PAD name	New function	IOA29	SPI1_TX	IOA30	SPI1_RX	IOB16	SPI1_CLK	0: original function(set in PSG) 1: Map to new function														
PAD name	New function																									
IOA29	SPI1_TX																									
IOA30	SPI1_RX																									
IOB16	SPI1_CLK																									
10	LCMTE1	R/W	IOB1 is LCM_TE <table border="1"> <tr><td>PAD name</td><td>New function</td></tr> <tr><td>IOB1</td><td>LCM_TE</td></tr> </table>	PAD name	New function	IOB1	LCM_TE	0: original function(set in PSG) 1: Map to new function																		
PAD name	New function																									
IOB1	LCM_TE																									
9	NANDRDY	R/W	only use one NAND_RDY0 IOD24	0: original function(set in PSG) 1: Map to new function																						
8	PPUI80	R/W	IOB0~7 & IOD3~10 is PPU I80 input	0: original function(set in PSG) 1: Map to new function																						
7	I2C	R/W	IOB18~19 is I2C interface <table border="1"> <tr><td>PAD name</td><td>New function</td></tr> <tr><td>IOB18</td><td>I2C0_SDA</td></tr> <tr><td>IOB19</td><td>I2C0_SCL</td></tr> </table>	PAD name	New function	IOB18	I2C0_SDA	IOB19	I2C0_SCL	0: original function(set in PSG) 1: Map to new function																
PAD name	New function																									
IOB18	I2C0_SDA																									
IOB19	I2C0_SCL																									
6	LCDTCON	R/W	IOC10~12 is tcon signal <table border="1"> <tr><td>PAD name</td><td>New function</td></tr> <tr><td>IOC10</td><td>TCON_STHR/L</td></tr> <tr><td>IOC11</td><td>TCON_STVU/D</td></tr> <tr><td>IOC12</td><td>TCON_POL</td></tr> </table>	PAD name	New function	IOC10	TCON_STHR/L	IOC11	TCON_STVU/D	IOC12	TCON_POL	0: original function(set in PSG) 1: Map to new function														
PAD name	New function																									
IOC10	TCON_STHR/L																									
IOC11	TCON_STVU/D																									
IOC12	TCON_POL																									
5	CSIFIELD	R/W	Not support in 216 pin	0: original function(set in PSG) 1: Map to new function																						
4	NAND1CS	R/W	Set IOA20 from NAND_CS1 to GPIO	0: original function(set in PSG) 1: Map to new function																						
3	NAND0CS	R/W	Set IOC23 from NAND_CS0 to GPIO	0: original function(set in PSG) 1: Map to new function																						

Bit	Function	Type	Description	Condition																		
2	CSIDAT	R/W	IOB0~7 is cmos data <table border="1"> <tr><td>PAD name</td><td>New function</td></tr> <tr><td>IOB0</td><td>SENSOR0_D0</td></tr> <tr><td>IOB1</td><td>SENSOR0_D1</td></tr> <tr><td>IOB2</td><td>SENSOR0_D2</td></tr> <tr><td>IOB3</td><td>SENSOR0_D3</td></tr> <tr><td>IOB4</td><td>SENSOR0_D4</td></tr> <tr><td>IOB5</td><td>SENSOR0_D5</td></tr> <tr><td>IOB6</td><td>SENSOR0_D6</td></tr> <tr><td>IOB7</td><td>SENSOR0_D7</td></tr> </table>	PAD name	New function	IOB0	SENSOR0_D0	IOB1	SENSOR0_D1	IOB2	SENSOR0_D2	IOB3	SENSOR0_D3	IOB4	SENSOR0_D4	IOB5	SENSOR0_D5	IOB6	SENSOR0_D6	IOB7	SENSOR0_D7	0: original function(set in PSG) 1: Map to new function
PAD name	New function																					
IOB0	SENSOR0_D0																					
IOB1	SENSOR0_D1																					
IOB2	SENSOR0_D2																					
IOB3	SENSOR0_D3																					
IOB4	SENSOR0_D4																					
IOB5	SENSOR0_D5																					
IOB6	SENSOR0_D6																					
IOB7	SENSOR0_D7																					
1	CEVAJTAG	R/W	IOA3~7 is CEVA JTAG <table border="1"> <tr><td>PAD name</td><td>New function</td></tr> <tr><td>IOA3</td><td>DSP_RTCK</td></tr> <tr><td>IOA4</td><td>DSP_TCK</td></tr> <tr><td>IOA5</td><td>DSP_TDO</td></tr> <tr><td>IOA6</td><td>DSP_TDI</td></tr> <tr><td>IOA7</td><td>DSP_TMS</td></tr> </table>	PAD name	New function	IOA3	DSP_RTCK	IOA4	DSP_TCK	IOA5	DSP_TDO	IOA6	DSP_TDI	IOA7	DSP_TMS	0: original function(set in PSG) 1: Map to new function						
PAD name	New function																					
IOA3	DSP_RTCK																					
IOA4	DSP_TCK																					
IOA5	DSP_TDO																					
IOA6	DSP_TDI																					
IOA7	DSP_TMS																					
0	ARMJTAG	R/W	IOA29~30, IOB16~17 is ARMJTAG <table border="1"> <tr><td>PAD name</td><td>New function</td></tr> <tr><td>IOA29</td><td>ARM_TMS</td></tr> <tr><td>IOA30</td><td>ARM_TDI</td></tr> <tr><td>IOB16</td><td>ARM_TDO</td></tr> <tr><td>IOB17</td><td>ARM_TCK</td></tr> </table>	PAD name	New function	IOA29	ARM_TMS	IOA30	ARM_TDI	IOB16	ARM_TDO	IOB17	ARM_TCK	0: original function(set in PSG) 1: Map to new function								
PAD name	New function																					
IOA29	ARM_TMS																					
IOA30	ARM_TDI																					
IOB16	ARM_TDO																					
IOB17	ARM_TCK																					

P_IOA_ENABLE **0x9000A000** **Select the IOA port to GPIO function or other function**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															

Default 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																

Default 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Function	Type	Description	Condition
[30:0]	IOA_EN	R/W	Select the IOA port to GPIO function or other function	0: normal mode 1: GPIO mode

P_IOB_ENABLE

0x9000A004

Select the IOB port to GPIO function or other function

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description													Condition
[30:0]	IOB_EN	R/W	Select the IOB port to GPIO function or other function													0: normal mode 1: GPIO mode

P_IOC_ENABLE

0x9000A008

Select the IOC port to GPIO function or other function

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description													Condition
[30:0]	IOC_EN	R/W	Select the IOC port to GPIO function or other function													0: normal mode 1: GPIO mode

P_IOD_ENABLE

0x9000A00C

Select the IOD port to GPIO function or other function

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1



Bit	Function	Type	Description	Condition
[30:0]	IOD_EN	R/W	Select the IOD port to GPIO function or other function	0: normal mode 1: GPIO mode

Bit	Function	Type	Description	Condition
[30:0]	IOA_O_DATA	R/W	Output data in IOA port	0: output low 1: output high

Bit	Function	Type	Description	Condition
[30:0]	IOB_O_DATA	R/W	Output data in IOB port	0: output low 1: output high

Bit	Function	Type	Description	Condition
[30:0]	IOC_O_DATA	R/W	Output data in IOC port	0: output low 1: output high

P_IOD_O_DATA **0x9000A01C** **Output data in IOD port**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOD_O_DATA	R/W	Output data in IOD port	0: output low 1: output high

P_IOA_DIR **0x9000A020** **Select IOA port to input mode or output mode**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOA_DIR	R/W	Select IOA port to input mode or output mode	0: Output mode 1: Input mode

P_IOB_DIR **0x9000A024** **Select IOB port to input mode or output mode**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOB_DIR	R/W	Select IOB port to input mode or output mode	0: Output mode 1: Input mode

P_IOC_DIR **0x9000A028** **Select IOC port to input mode or output mode**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOC_DIR	R/W	Select IOA port to input mode or output mode	0: Output mode 1: Input mode

P_IOD_DIR **0x9000A02C** **Select IOD port to input mode or output mode**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOD_DIR	R/W	Select IOD port to input mode or output mode	0: Output mode 1: Input mode

P_IOA_POL **0x9000A030** **Select the polarity active low or active high for interrupt**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[30:0]	IOA_POL	R/W	Setting the polarity to trigger interrupt, rising edge or falling edge	0: falling edge 1: rising edge

P_IOB_POL
0x9000A034
Select the polarity active low or active high for interrupt

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[30:0]	IOB_POL	R/W	Setting the polarity to trigger interrupt, rising edge or falling edge	0: falling edge 1: rising edge

P_IOC_POL
0x9000A038
Select the polarity active low or active high for interrupt

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[30:0]	IOC_POL	R/W	Setting the polarity to trigger interrupt, rising edge or falling edge	0: falling edge 1: rising edge

P_IOD_POL
0x9000A03C
Select the polarity active low or active high for interrupt

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[30:0]	IOD_POL	R/W	Setting the polarity to trigger interrupt, rising edge or falling edge	0: falling edge 1: rising edge

P_IOA_STICKY
0x9000A040
Select Direct output or clocked rising edge after polarity change

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															

Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															

Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bit	Function	Type	Description	Condition
[30:0]	IOA_STICKY	R/W	If sticky set to 1 P_IOA_STATUS read the INT Flag status If sticky set to 0 P_IOA_STATUS read the pad input	

P_IOB_STICKY
0x9000A044
Select Direct output or clocked rising edge after polarity change

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															

Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															

Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bit	Function	Type	Description	Condition
[30:0]	IOB_STICKY	R/W	If sticky set to 1 P_IOB_STATUS read the INT Flag status If sticky set to 0 P_IOB_STATUS read the pad input	

P_IOC_STICKY

0x9000A048

**Select Direct output or clocked rising
edge after polarity change**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	IOC_STICKY														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOC_STICKY															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description														Condition
[30:0]	IOC_STICKY	R/W	If sticky set to 1 P_IOC_STATUS read the INT Flag status If sticky set to 0 P_IOC_STATUS read the pad input														

P_IOD_STICKY

0x9000A04C

**Select Direct output or clocked rising
edge after polarity change**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	IOD_STICKY														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description														Condition
[30:0]	IOD_STICKY	R/W	If sticky set to 1 P_IOD_STATUS read the INT Flag status If sticky set to 0 P_IOD_STATUS read the pad input														

P_IOA_INTEN

0x9000A050

Interrupt Enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	IOA_INTEN														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOA_INTEN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOA_INTEN	R/W	Interrupt enable	0: Disable 1: Enable

P_IOB_INTEN																
0x9000A054																
Interrupt Enable																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOB_INTEN	R/W	Interrupt enable	0: Disable 1: Enable

P_IOC_INTEN																
0x9000A058																
Interrupt Enable																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOC_INTEN	R/W	Interrupt enable	0: Disable 1: Enable

P_IOD_INTEN																
0x9000A05C																
Interrupt Enable																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOD_INTEN	R/W	Interrupt enable	0: Disable 1: Enable

P_IOA_INTPEND **0x9000A060** **Interrupt Pending Signal**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOA_INTPEND	R/W/C	Interrupt pending (=INT flag & INT en) This interrupt pending flag can be clear by: 1.disable GPIO 2.set direct to output 3.write 1 then write 0 to the corresponding bit	0: no interrupt 1: interrupt pending

P_IOB_INTPEND **0x9000A064** **Interrupt Pending Signal**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOB_INTPEND	R/W/C	Interrupt pending (=INT flag & INT en) This interrupt pending flag can be clear by: 1.disable GPIO 2.set direct to output 3.write 1 then write 0 to the corresponding bit	0: no interrupt 1: interrupt pending

P_IOC_INTPEND
0x9000A068
Interrupt Pending Signal

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	IOC_INTPEND															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOC_INTPEND															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description													Condition
[30:0]	IOC_INTPEND	R/W/C	Interrupt pending (=INT flag & INT en) This interrupt pending flag can be clear by: 1.disable GPIO 2.set direct to output 3.write 1 then write 0 to the corresponding bit													0: no interrupt 1: interrupt pending

P_IOD_INTPEND
0x9000A06C
Interrupt Pending Signal

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	IOD_INTPEND															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOD_INTPEND															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description													Condition
[30:0]	IOD_INTPEND	R/W/C	Interrupt pending (=INT flag & INT en) This interrupt pending flag can be clear by: 1.disable GPIO 2.set direct to output 3.write 1 then write 0 to the corresponding bit													0: no interrupt 1: interrupt pending

P_IOA_DEBOUNCE_CNT
0x9000A070
Debounce counter setting

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOA_CNT															

Default 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Function	Type	Description	Condition
[21:0]	IOA_CNT	R/W	Debounce counter setting For example: ARM main clock is 400Mhz, Subsystem run on 100Mhz. If this register set to 100, then the debounce time is 1us	Number of clock cycle

P_IOB_DEBOUNCE_CNT 0x9000A074 Debounce counter setting																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-										IOB_CNT					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOB_CNT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[21:0]	IOB_CNT	R/W	Debounce counter setting For example: ARM main clock is 400Mhz, Subsystem run on 100Mhz. If this register set to 100, then the debounce time is 1us	Number of clock cycle

P_IOC_DEBOUNCE_CNT 0x9000A078 Debounce counter setting																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-										IOC_CNT					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOC_CNT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[21:0]	IOC_CNT	R/W	Debounce counter setting For example: ARM main clock is 400Mhz, Subsystem run on 100Mhz. If this register set to 100, then the debounce time is 1us.	Number of clock cycle

P_IOD_DEBOUNCE_CNT															0x9000A07C	Debounce counter setting																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		IOD_CNT														
Function	-																IOD_CNT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		IOD_CNT														
Function	IOD_CNT																IOD_CNT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

Bit	Function	Type	Description															Condition
[21:0]	IOD_CNT	R/W	Debounce counter setting For example: ARM main clock is 400Mhz, Subsystem run on 100Mhz. If this register set to 100, then the debounce time is 1us.															Number of clock cycle

P_IOA_DEBOUNCE_EN															0x9000A080	Debounce Enable																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		IOA_DEB_EN														
Function	-	IOA_DEB_EN																IOA_DEB_EN														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		IOA_DEB_EN														
Function	IOA_DEB_EN																IOA_DEB_EN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Bit	Function	Type	Description															Condition
[30:0]	IOA_DEB_EN	R/W	Debounce enable 0: Disable 1: Enable debounce															0: Disable 1: Enable debounce

P_IOB_DEBOUNCE_EN															0x9000A084	Debounce Enable																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		IOB_DEB_EN														
Function	-	IOB_DEB_EN															IOB_DEB_EN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		IOB_DEB_EN														
Function	IOB_DEB_EN																IOB_DEB_EN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			



Bit	Function	Type	Description	Condition
[30:0]	IOB_DEB_EN	R/W	Debounce enable	0: Disable 1: Enable debounce

Bit	Function	Type	Description	Condition
[30:0]	IOC_DEB_EN	R/W	Debounce enable	0: Disable 1: Enable debounce

Bit	Function	Type	Description	Condition
[30:0]	IOD_DEB_EN	R/W	Debounce enable	0: Disable 1: Enable debounce



Bit	Function	Type	Description	Condition
[30:0]	IOA_WKUEN	R/W	Wake up enable	0: Disable 1: Enable wake up

Bit	Function	Type	Description	Condition
[30:0]	IOB_WKUEN	R/W	Wake up enable	0: Disable 1: Enable wake up

Bit	Function	Type	Description	Condition
[30:0]	IOC_WKUEN	R/W	Wake up enable	0: Disable 1: Enable wake up

Bit	Function	Type	Description	Condition
[30:0]	IOD_WKUEN	R/W	Wake up enable	0: Disable 1: Enable wake up

P_IOA_STATUS		0x9000A0A0																GPIO Status	
Bit	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	IOA_STATUS	
Default	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IOA_STATUS	
Bit	Function	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IOA_STATUS	
Default	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IOA_STATUS	

Bit	Function	Type	Description	Condition
[30:0]	IOA_STATUS	R	If P_IOA_STICKY set to 1 read the INT Flag status If P_IOA_STICKY set to 0 read the pad input	

P_IOB_STATUS		0x9000A0A4																GPIO Status	
Bit	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	IOB_STATUS	
Default	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IOB_STATUS	
Bit	Function	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IOB_STATUS	
Default	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IOB_STATUS	

Bit	Function	Type	Description	Condition
[30:0]	IOB_STATUS	R	If P_IOB_STICKY set to 1 read the INT Flag status If P_IOB_STICKY set to 0 read the pad input	

P_IOC_STATUS		0x9000A0A8																GPIO Status	
Bit	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	IOC_STATUS	
Default	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IOC_STATUS	
Bit	Function	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IOC_STATUS	
Default	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IOC_STATUS	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOC_STATUS															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOC_STATUS	R	If P_IOC_STICKY set to 1 read the INT Flag status If P_IOC_STICKY set to 0 read the pad input	

P_IOD_STATUS																
0x9000A0AC																
GPIO Status																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Function	-	IOD_STATUS														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOD_STATUS															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[30:0]	IOD_STATUS	R	If P_IOD_STICKY set to 1 read the INT Flag status If P_IOD_STICKY set to 0 read the pad input	

42 AES

42.1 Features

- Hardware Encryption
- Hardware Decryption
- 128bits key

43 Random Number Generator

43.1 Introduction

GPL32900A has a built-in random number generator (RNG for short) for getting one 32-bit random number. The generated number can be used as KEY of DES, TDES and AES function built-in DRM module. When using random number generator, user writes 1 to “READ_EN” (bit 0 of Address: 0x9000_9000) at first. Then reading the “OUT_VALID” (bit 0 of Address 0x9000_9018). If “OUT_VALID”=1, the generated number is ready and 32-bit random number can be read back from “OUTPUT_DATA” port (Address 0x9000_9014). If “OUT_VALID=0”, the generated number is not ready yet and user should continue pulling until “OUT_VALID”=1. The internal seed of Random Number Generator will change over time randomly. User can't access the intenal seed for security except manual reset for intenal seed by writing1 to “seed reset” (Address 0x9000_9008).

43.2 Register Summary

Name	Address	Description
READ	0x90009000	RNG Enable Register
LONG_RUN_VALUE	0x90009004	Reserved Register
MANUAL_SEED_RESET	0x90009008	RNG Seed Reset Register
OUTPUT_DATA	0x90009014	RNG Output Data Register
OUTPUT_VALID	0x90009018	RNG Output Status Register
LONG_RUN_COUNT	0xC008001C	Reserved Register

43.3 Register Definition

0x90009000								RNG Enable Register	
Bit	15	14	13	12	11	10	9	8	
Function	-	-	-	-	-	-	-	-	
Default	0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0	
-	-	-	-	-	-	-	-	READ_EN
0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:1]	-	-	Reserved	
1	READ_EN	W	Write 1 to trigger RNG to generate a new random number output	

Manual Seed Reset
0x90009008
RNG Seed Reset Register

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	Seed Reset
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:1]	-	-	Reserved					
0	Seed Reset	R	Manual Reset For Internal Seed					

OUTPUT_DATA
0x90009014
RNG Output Data Register

Bit	31	30	29	28	27	26	25	24
Function	RNG OUTPUT DATA							
Default	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
RNG OUTPUT DATA							
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
RNG OUTPUT DATA							
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
RNG OUTPUT DATA							
0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	OUTPUT_DATA	R	32-bit output data for RNG					

OUTPUT_VALID		0x9009001C								RNG Output Status Register							
Bit		15	14	13	12	11	10	9	8								
Function	-	-	-	-	-	-	-	-	-								
Default	0	0	0	0	0	0	0	0	0								

7	6	5	4	3	2	1	0	OUT_VALID
-	-	-	-	-	-	-	-	OUT_VALID
0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:1]	-	-	Reserved	
0	OUT_VALID		Status for valid output data of RNG	1 : Data Valid

44 Power Control (PWRC)

44.1 Introduction

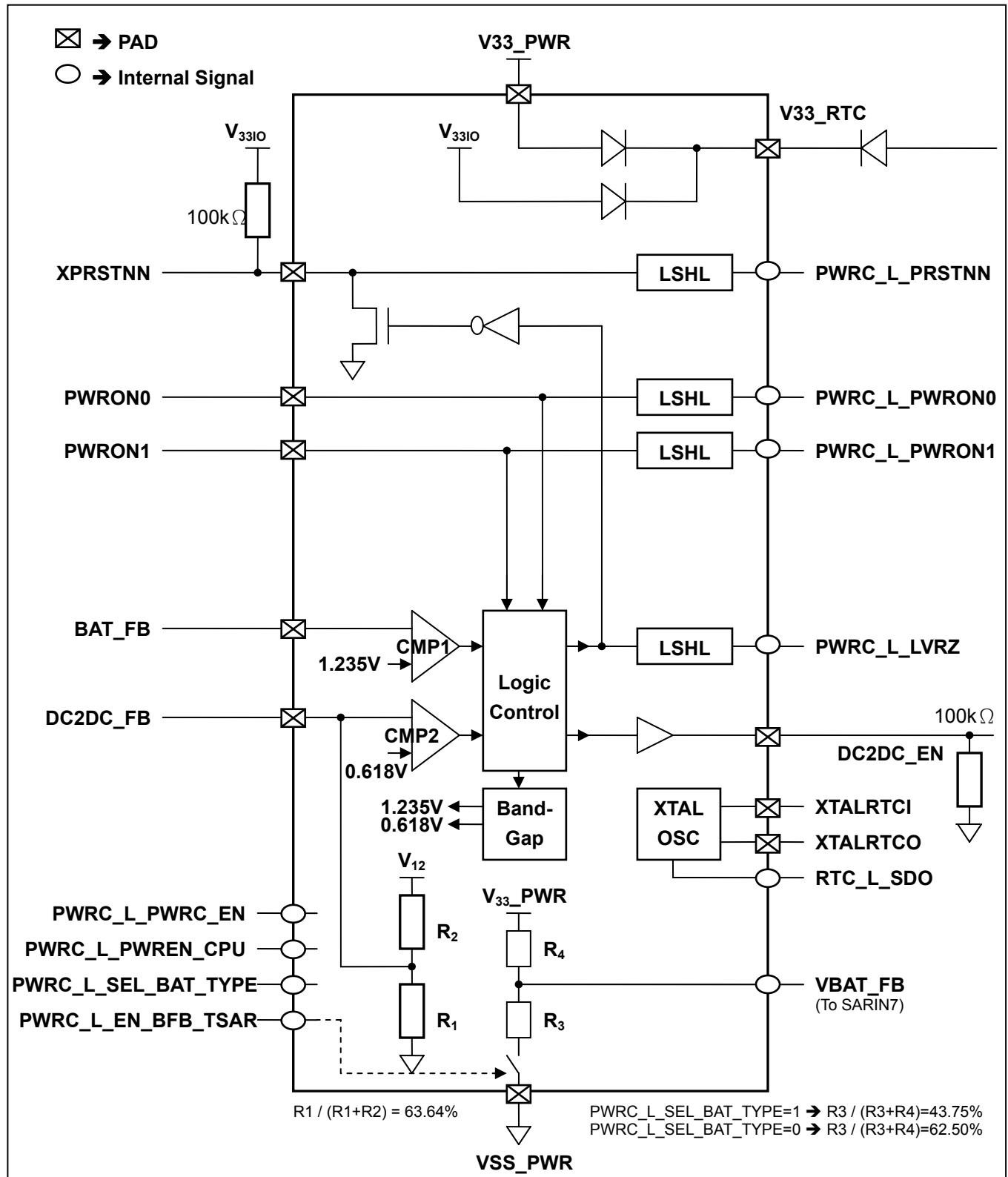
The main function of PWRC macro is to turn on external power (when battery voltage is good) and generate low voltage reset for main chip. The RTC power can be supplied from V33_PWR by internal path. When RTC uses V33_PWR power, external OVDDRTC supply can be removed. PWRC also feedbacks battery voltage (with 43.75% or 62.5% of V33_PWR) to SARIN7 to measure battery voltage. Users can choose to use an external component to replace the PWRC function or disable low voltage reset/ battery good detect by higher BAT_FB & DC2DC_FB voltage.

44.2 Features

- Power good detection
- Power-on signal and internal reset generation
- Low voltage detection
- Can supply RTC power through internal path
- Can power-on from RTC interrupt
- Can directly feed V33_PWR voltage to SARIN7
- When BAT_FB > VBG1 and is approx. 1.235 V, DC2DC_EN is high. When BAT_FB < VBG1 and is approx. 1.06V, DC2DC_EN is low.
- When DC2DC_FB > VBG2 and is approx. (1.235/2)V, LVRZ=1. When DC2DC_FB < VBG2 and is approx. 0.5V, the low voltage reset XPRSTNN will be triggered and LVRZ=0.
- VBAT_FB is approx. 43.75% of V33_PWR when SEL_BAT_TYPE=1
- VBAT_FB is approx. 62.5% of V33_PWR when SEL_BAT_TYPE=0

44.3 Block Diagram

The following diagram is a functional block diagram of the power control module.



44.4 Power Control Pin Configuration

Item	PIN Name	Dir	Description
1	V33_PWR	I	3.3v Power source
2	BAT_FB	I	Battery power feedback
3	DC2DC_FB	I	DC2DC power feedback
4	DC2DC_EN	O	DC2DC power output
5	PWRON1	I	Power ON control pin1
6	PWRON0	I	Power ON control pin0
7	VSS_PWR	I	3.3v Power ground
8	XTALRTCI	O	RTCI
9	XTALRTCO	O	RTCO
10	OVDDRTC	O	RTC power source

Notes: DC2DC_FB is an internal node.

PWR_DC2DC_EN:

Function description:

The external LDO, which is the power source (I/O, hard macro, DDRPHY and core) of main chip, was controlled by this PWR_DC2DC_EN pin. There will be no power source provided to IC from LDO if PWR_DC2DC_EN pin = 0.

Formula:

$$\text{PWR_DC2DC_EN} = (\text{V}_{\text{PWR_ONKEY0}} \parallel \text{V}_{\text{PWR_ONKEY1}}) \&\& \text{V}_{\text{PWR_BT_FB}}$$

Application:

The default value of PWREN_CPU is low when the power is on (by pressing the key of PWR_ONKEY1 or PWR_ONKEY0). Then PWREN_CPU will be set to low by boot loader and the PWRON key can be released. The procedure of switching off the IC is as below: First, CPU will detect the interrupt send from POWRON key. Then the PWREN_CPU will be forced to turn off and the PWRON key will be released. Finally, the whole chip will be turn off.

PWR_BT_FB:

Function description:

PWR_BT_FB usually is the divided-voltage of the main system power source (such as battery).

PWR_DC2DC_EN will be forced to LOW or not is decided by comparing PWR_BT_FB and 1.235V.

Compare function:

$$\text{PWR_BT_FB} \geq 1.26V (1.235 + 0.025), \text{ PWR_BT_FB_COMPARE_RESULT} = 1$$

$$\text{PWR_BT_FB} \leq 1.21V (1.235 - 0.025), \text{ PWR_BT_FB_COMPARE_RESULT} = 0$$

Application:

When the power of main system is running off, PWR_BT_FB will be changed by low voltage and the PWR_BT_FB_COMPARE_RESULT will be 0. Then PWR_DC2DC_EN will be forced to be LOW and

the chip can NOT be turned on anymore even the RTC_ALARM or PWRON key is on. At the same time, power on reset will be triggered and PWREN_CPU will be LOW.

PWR_BT_FB:

Function description:

PWR_BT_FB is the internal divided-voltage of VDD3V3_PWR power and connect with SARIN7 signal from SAR_ADC. The formula is as below:

Select 0 : $PWR_BT_FB = VDD3V3_PWR * 0.625$

Select 1 : $PWR_BT_FB = VDD3V3_PWR * 0.4375$

Application:

PWR_BT_FB is used to detect low power of the main system. Usually VDD3V3_PWR is the divided-voltage of main system power source. When power is exhausted but LVR is not triggered yet, PWR_BT_FB_COMPARE_RESULT will be LOW and the detecting signal of SARIN7 from SAR_ADC is below the standard value. Then a warning message will be sent to users to power down through UI.

44.5 Power Control Mode

A. Standby Mode

Test Condition: V33_PWR=3.7V, V33=3.3V, $V_{PWRON0}=0V$, $V_{PWRON1}=0V$, (Power on key → switch OFF)

$V_{BAT_FB}=1.5V$, $V_{DC2DC_FB}=0.8V$, (Battery and power feedback both exist)

PWRC_L_PWRC_EN = 0

PWRC_L_PWREN_CPU=0

PWRC_L_SEL_BAT_TYPE=0

PWRC_L_EN_BFB_TSAR=0

Measure Results: PWRC_L_PWRON0 = 0 & (No power on 0 interrupt)

PWRC_L_PWRON1 = 0 & (No power on 1 interrupt)

PWRC_L_PRSTNN = 0 & (power on reset interrupt enable)

PWRC_L_LVRZ = 0 & (low voltage reset interrupt enable)

-0.1V ≤ DC2DC_EN ≤ 0.1V ==> DC2DC_EN doesn't work, Pass!

DC2DC_EN = ($V_{PWRON0} \parallel V_{PWRON1}$) && V_{BAT_FB}

PWRC_L_LVRZ = DC2DC_EN && V_{DC2DC_FB}

PWRC_L_PRSTNN = PWRC_L_LVRZ

B. Normal Power ON

Test Condition: V33_PWR=3.7V, V33=3.3V, $V_{PWRON0}=3.7V$, $V_{PWRON1}=3.7V$, (Power on key → switch ON)

$V_{BAT_FB}=1.5V$, $V_{DC2DC_FB}=0.8V$, (Battery and power feedback both exist)

PWRC_L_PWRC_EN = 0

PWRC_L_PWREN_CPU=0

PWRC_L_SEL_BAT_TYPE=0

PWRC_L_EN_BFB_TSAR=0

Measure Results: PWRC_L_PWRON0 = 1 & (power on 0 interrupt enable)

PWRC_L_PWRON1 = 1 & (power on 0 interrupt enable)

PWRC_L_PRSTNN = 1 & (power on reset interrupt disable)

PWRC_L_LVRZ = 1 & (low voltage reset interrupt disable)

3.5V ≤ DC2DC_EN ≤ 3.8V ==> DC2DC_EN works, Pass!

C. Battery Power ON

Test Condition: V33_PWR=3.7V, V33=3.3V, V_{PWRON0}=3.7V, V_{PWRON1}=3.7V, (Power on key → switch ON)

V_{BAT_FB}=1.5V, V_{DC2DC_FB}=0.5V, (ONLY Battery exists)

PWRC_L_PWRC_EN = 0

PWRC_L_PWREN_CPU=0

PWRC_L_SEL_BAT_TYPE=0

PWRC_L_EN_BFB_TSAR=0

Measure Results: PWRC_L_PWRON0 = 1 & (power on 0 interrupt enable)

PWRC_L_PWRON1 = 1 & (power on 1 interrupt enable)

PWRC_L_PRSTNN = 0 & (power on reset interrupt enable)

PWRC_L_LVRZ = 0 & (low voltage reset enable)

3.5V ≤ DC2DC_EN ≤ 3.8V ==> DC2DC_EN works, Pass!

D. Low Power Lock

Test Condition: V33_PWR=3.7V, V33=3.3V, V_{PWRON0}=3.7V, V_{PWRON1}=3.7V, (Power on key → switch ON)

V_{BAT_FB}=1.15V, V_{DC2DC_FB}=0.5V, (Low power)

PWRC_L_PWRC_EN = 0

PWRC_L_PWREN_CPU=0

PWRC_L_SEL_BAT_TYPE=0

PWRC_L_EN_BFB_TSAR=0

Measure Results: PWRC_L_PWRON0 = 1 & (power on 0 interrupt enable)

PWRC_L_PWRON1 = 1 & (power on 1 interrupt enable)

PWRC_L_PRSTNN = 0 & (power on reset interrupt enable)

PWRC_L_LVRZ = 0 & (low voltage reset interrupt enable)

-0.1V ≤ DC2DC_EN ≤ 0.1V ==> DC2DC_EN doesn't work, Pass!