



everyday genius

[MT2712] IVI_Functional Specification

Version: 2.20

Release Date: 2021-10-09

Use of this document and any information contained therein is subject to the terms and conditions set forth in [Exhibit 1](#). This document is subject to change without notice.

Abbreviations

Abbreviation	Description
ADAS	Advanced Driver Assistance Systems
ADC	Analog-Digital Converter
AER	Advanced Error Reporting
AGC	Automatic Gain Control
AHB	Advanced High-performance Bus
AI	Analog Input
AIO	Analog Bi-direction
AO	Analog Output
AP	Application Processor
APB	Advanced Peripherals Bus
AP_DMA	Application Processor DMA
APMCU	Application Processing Microcontroller Unit
Apmixedsys	Application Processor Mixed Mode Control System
APXGPT	Application Processor X General-Purpose Timer
ASRC	Asynchronous Sample Rate Converter
ATB	Advanced Trace Bus
AUXADC	Auxiliary Analog-Digital Converter
AVB	Audio Video Bridging
AVM	Around-View Monitoring
AXI	Advanced eXtensible Interface
BCK	Bit Clock
BD	Buffer Descriptor
BDP	Buffer Descriptor Present
BG	Background
BNR	Block Noise Reduction
BPS	Bypass
BT	Bluetooth
BTA	Bus Turnaround
CABC	Content Adaptive Backlight Controller
CBP	Constrained Baseline Profile
CG	Clock Gating
CHiP	Constrained High Profile
CLK	Clock
CPU	Central Processing Unit
CPHA	Clock Phase
CPOL	Clock Polarity

Abbreviation	Description
CQ_DMA	Command Queue DMA
CRC	Cyclic Redundancy Check
CS	Complete Spilt
CSI	Camera Serial Interface
CSSYS	CoreSight System
CTI	Cross Trigger Interface
CTM	Cross Trigger Macrocell
CTS	Clear to Send
CVBS	Composite Video Baseband Signal
DAP	Debug Access Port
DAPC	Device Access Protection Control
DB	Deblocking Filter
DCC	Debug Communication Channels
DCM	Dynamic Clock Management
DCS	Display Command Set
DE	Data Enable
DFD	Design for Debug
DFS	Dynamic Frequency Scaling
DI	Digital Input
DI	De-interlacing
DIO	Digital Bi-direction
DISP	Display
DISP_AAL	Display Adaptive Ambient Light
DISP_COLOR	Display Color
DISPFMT	Display Format
DISP_GAMMA	Display Gamma
DISP_MONITOR	Display Monitor
DISP_MUTEX	Display Mutex
DISP_OD	Display OD
DISP_OVL	Display Overlay
DISP_RDMA	Display Read Direct Memory Access
Dispsys	Display System
DISP_UFOe	Display UFOe
DISP_WDMA	Display Write Direct Memory Access
DMA	Direct Memory Access
DnT	Debug and Trace
DO	Digital Output
DPHY	Digital Physical
DPI	Digital Parallel Interface

Abbreviation	Description
DR	Data Ready
DRAM	Dynamic Random-Access Memory
DRAMC	DRAM Controller
DRE	Dark Region Enhancement
DSI	Display Serial Interface
DTV	Digital TV
DW	Dual Word
EAV	End of Active Video
EC	Entropy encoder
ECC	Error Correction Code
ECC	Error Correcting Code
EINTC	External Interrupt Controller
EMCU	External MCU
EMI	External Memory Interface
EOF	End of Frame
EOL	End of List
EP	Endpoint
ETB	Embedded Trace Buffer
ETHER_QOS	Ethernet QOS
FCR	FIFO Control Register
FHCTL	Frequency Hopping Controller
FIFO	First-In First-Out
FIR	Finite impulse response
FLR	Function Level Reset
FM	Frequency Modulation
FPS	frames per second
FS	Full-speed
FSM	Finite State Machine
G	Ground
GIC	Generic Interrupt Controller
GPD	General Packet Descriptor
GPD	General Purpose Descriptor
GPI	General-Purpose Input
GPIO	General-Purpose Input/Output
GPT	General-Purpose Timer
GPU	Graphic Processing Unit
HD	High Definition
HDMI	High-Definition Multimedia Interface
HiP	High Profile

Abbreviation	Description
HMI	Human-Machine Interface
HS	Horizontal Sync
HS	High-speed
HSTX	High-Speed Transmitter
H/V	Horizontal/Vertical
HW	Hardware
HWO	Hardware Own
IER	Interrupt Enable Register
IIR	Interrupt Identification Register
IMG_RSZ	Image Resizer
IMGSYS_TOP	Image Subsystem
IMGSYS	Image System
INFRACFG	Infrastructure System Configuration
IOC	Interrupt On Completion
IRQ	Interrupt Request
IRRX	Infrared Receiver
ISP	Internet Service Provider
ISR	Interrupt Service Request
ISR	Interrupt Status Register
I2S	Inter-IC Sound
I2C/SCCB	Inter-IC/Serial Camera Control Bus
IVI	In-Vehicle-Infotainment
LCD	Liquid Crystal Display
LCM	Liquid Crystal Monitor
LCR	Line Control Register
LCs	Liquid Crystals
LDO	Low Drop-out
LFPS	Low Frequency Periodic Signaling
LFSR	Linear Feedback Shift Register
LPF	Low-Pass Filter
LPM	Lower Power Management
LPTX	Low-Power Transmitter
LS	Low-speed
LSB	Least Significant Bit
LSR	Line Status Register
LUT	Look-up Table
LVDS	Low-Voltage Differential Signaling
MA	Motion-Adaptive
MAU	Memory Assertion Unit

Abbreviation	Description
MC	Motion Compensation
MCDI	Multi-core-deep-idle
MCR	Modem Control Register
MCU	Microcontroller Unit
MCUSYS	Microcontroller Unit System
MDDi	Media Direct De-interlacing
MDP 2.0	Multimedia Data Path v2.0
MDP_RDMA	Multimedia Data Path Read DMA
MDP_RSZ	Multimedia Data Path Resizer
MDP_TDSHP	Multimedia Data Path 2D Sharpness
MDP_WDMA	Multimedia Data Path Write DMA
MDP_WROT	Multimedia Data Path Rotation
ME	Motion Estimation
MFG	MFlexGraphics
MHL	Mobile High-Definition Link
MIC	Microphone
MIPI_TX_Config	MIPI TX Configuration module
MM	Multimedia
MMSYS	Multimedia System
MMSYS_TOP	Multimedia Subsystem
MM_IOMMU	Multimedia Memory Management Unit
MNR	Mosquito Noise Reduction
MP	Main Profile
MPU	Memory Protection Unit
MPS	Maximum Packet Size
MSB	Most Significant Bit
MSR	Modem Status Register
MTCMOS	Multi-threshold CMOS
MUX	Multiplexer
NFI	NAND Flash Interface
NR	Noise Reduction
OS	Operating system
OVL_RDMA	Overlay RDMA
P	Power
PBC	Peaking by color
PCM	Pulse-Code Modulation
PCIe	PCI Express
Pericfg	Peripheral Configuration
PHY	Physical

Abbreviation	Description
PHYA	Physical Analog
PHYD	Physical Digital
PLL	Phase-Locked Loop
PLLGP	Phase-Locked Loop Group
POR	Power-On-Reset
PQ	Picture Quality
ProHiP	Progressive High Profile
PTM	Program Flow Trace Macrocell
PTP	Precision Time Protocol
PWM	Pulse-width Modulation
QMU	Queue Management Unit
QOS	Quality of Service
RAM	Random Access Memory
RBR	Rx Buffer Register
RC	Root Complex
RF	Radio Frequency
RH	Relative Humidity
ROI	Region of Interest
ROM	Read-only Memory
RSZ	Resizer
RTC	Real Time Clock
RTS	request to send
RX	Receiver
SAR	Successive-approximation-register
SAV	Start of Active Video
SCL	Serial Clock Line
SCPSYS	System Control Processor System
SCR	Scratch Register
SD	Standard Definition
SDA	Serial Data Line
SMI	Smart Multimedia Interface
SMI	Smart Memory Interface
SNR	Spatial Noise Reduction
SoC	System on a Chip
SOF	Start of Frame
SP	Settle Parameters
SPM	System Power Management
SPM	System Power Manger
SPI	Serial Peripheral Interface

Abbreviation	Description
SSUSB_PCIE_PHYD	USB SuperSpeed and PCI Express PHY Digital
SRAM	Static Random-Access Memory
SRC	Sample Rate Converter
SS	Start Split
SS	SuperSpeed
SSC	Spread Spectrum Clocking
SSUSB	SuperSpeed Universal Serial Bus
SW	Software
SWD	Serial Wire Debug
SYS_CIRQ	System Interrupt Controller
sys_timer	System Timer
TDM	Time-Division Multiplexing
TE	Tearing Effect
THR	TX Holding Register
TLB	Translation Look Aside Buffer
TNR	Temporal Based Frame Noise Reduction
TOPCKGEN	Top Clock Generator
TOPRGU	Top Reset Generation Unit
TPIU	Trace Port Interface Unit
TQ	Transform and Quantization
TRB	Transfer Request Block
TVD	TV Decoder
TX	Transmitter
2D	Two-dimensional
UART	Universal Asynchronous Receiver/Transmitter
UI	Unit Interval
ULPM	Ultra-low Power Mode
ULPS	Ultra low power state
USB	Universal Serial Bus
VDEC	Video Decoder
VDO	Video De-interlacing
VENC	Video Encoder
VS	Vertical Sync
WDMA	Write Direct Memory Access
WR_CHANNEL	CAM Write Channel
xHC	Host Controller
Xhcd	Host Controller Driver
Xhci	eXtensible Host Controller Interface
XO	Crystal Oscillator

Abbreviation	Description
ZLP	Zero Length Packet

Table of Contents

Abbreviations	2
Table of Contents.....	10
List of Tables.....	30
List of Figures.....	35
1 System Overview.....	42
1.1 General Description	42
1.2 Target Applications	43
1.3 Key Features.....	43
1.3.1 CPU Clusters.....	43
1.3.2 Memory	43
1.3.3 GPU	44
1.3.4 Video/Audio Interfaces.....	44
1.3.5 Multimedia Processing.....	45
1.3.6 Peripherals and Control	46
1.4 Key Parameters	46
1.4.1 Process and Package.....	46
1.4.2 Quality.....	47
1.5 Supporting.....	47
1.5.1 OS and BSP Support	47
1.5.2 Companion Components	47
2 Product Description.....	49
2.1 Pin Description	49
2.1.1 Ball Map View	49
2.1.2 Pin Coordinates.....	50
2.1.3 Detailed Pin Description	57

2.2	Electrical Characteristics	71
2.2.1	Absolute Maximum Ratings	71
2.2.2	Recommended Operating Conditions.....	72
2.2.3	Storage Conditions.....	74
2.2.4	AC Electrical Characteristics and Timing Diagram.....	74
2.3	System Configuration.....	80
2.3.1	Constant Tied Pins	80
2.4	Power-On Sequence.....	81
2.5	Analog IP Information	83
2.5.1	Introduction	83
2.5.2	Analog Block Features.....	83
2.5.3	AUXADC Block	83
2.5.4	XTAL	85
2.5.5	Phase-Locked Loop (PLL).....	85
2.5.6	Temperature Sensor	91
2.5.7	AADC	91
2.5.8	CVBS	93
2.6	Package Information	94
2.6.1	Package Outlines.....	94
2.6.2	Thermal Operating Specifications.....	96
2.6.3	Lead-free Packaging	96
2.7	Ordering Information	96
2.7.1	Top Marking Definition	97
3	Clock and Power Control	98
3.1	Crystal Oscillator	98
3.1.1	Introduction	98
3.1.2	Features	98
3.2	Application Processor Mixed Mode Control System.....	98

3.2.1	Introduction	98
3.2.2	Features	98
3.2.3	Block Diagram	99
3.2.4	Register Definition	99
3.2.5	Programming Guide	99
3.3	Top Clock Generator	108
3.3.1	Introduction	108
3.3.2	Features	108
3.3.3	Block Diagram	109
3.3.4	Register Definition	115
3.3.5	Programming Guide	115
3.4	Frequency Hopping Controller	131
3.4.1	Introduction	131
3.4.2	Features	131
3.4.3	Block Diagram	132
3.4.4	Register Definition	132
3.4.5	Programming Guide	132
3.5	Top Reset Generation Unit	134
3.5.1	Introduction	134
3.5.2	Features	134
3.5.3	Block Diagram	134
3.5.4	Register Definition	134
3.5.5	Programming Guide	135
3.6	MTCMOS Domains and DFS	137
3.6.1	Introduction	137
3.6.2	Features	137
3.6.3	Block Diagram	137
3.7	Real Time Clock	139

3.7.1	Introduction	139
3.7.2	Features	140
3.7.3	RTC Block Diagram	140
3.7.4	Register Definition	140
3.7.5	Programming Guide	140
4	MCU and Bus Fabric.....	142
4.1	MCU System	142
4.1.1	Introduction	142
4.1.2	Features	142
4.1.3	MCUSYS Block Diagram.....	142
4.1.4	Register Definition	143
4.1.5	MCUSYS Interrupts	143
4.1.6	MCUSYS Power Architecture	145
4.2	Debug System.....	147
4.2.1	Introduction	147
4.2.2	Reference	147
4.2.3	Features	147
4.2.4	Block Diagram	147
4.2.5	JTAG AC Timing	148
4.2.6	Application Processor Debug Subsystem.....	149
4.2.7	Register Definition	149
4.2.8	Programming Guide	149
4.3	System Interrupt Controller	150
4.3.1	Introduction	150
4.3.2	Features	150
4.3.3	Block Diagram	150
4.3.4	Register Definition	151
4.3.5	Programming Guide	152

4.4	External Interrupt Controller.....	152
4.4.1	Introduction	152
4.4.2	Features	153
4.4.3	Block Diagram	153
4.4.4	Register Definition	154
4.4.5	Programming Guide	155
4.5	Infrastructure System Configuration.....	155
4.5.1	Introduction	155
4.5.2	Features	156
4.5.3	Register Definition	157
4.6	On-chip Memory Controller	157
4.6.1	Introduction	157
4.6.2	Block Diagram	158
4.6.3	Boot ROM Power-Down Mode	158
4.6.4	Register Definition	159
4.7	Application Processor DMA (AP_DMA).....	159
4.7.1	Introduction	159
4.7.2	Features	159
4.7.3	Block Diagram	159
4.7.4	Register Definition	160
4.7.5	Programming Guide	160
4.8	Command Queue DMA (CQ_DMA)	161
4.8.1	Introduction	161
4.8.2	Features	161
4.8.3	Block Diagram	162
4.8.4	Register Definition	162
4.8.5	Programming Guide	163
4.9	CM4SYS	163

4.9.1	Introduction	163
4.9.2	Features	163
4.9.3	Block Diagram	164
4.9.4	Programming Guide	164
5	Peripherals	167
5.1	Memory Map	167
5.1.1	Introduction	167
5.1.2	Top Memory Map	167
5.1.3	Boot Section Memory Map	168
5.1.4	Infrastructure System Memory Map	169
5.1.5	Peripheral System Memory Map	170
5.1.6	Multimedia System Memory Map	172
5.1.7	Memory Map for 4GB DRAM Support	175
5.2	GPIO Controller	176
5.2.1	Introduction	176
5.2.2	Features	177
5.2.3	Block Diagram	177
5.2.4	GPIO AC Timing	178
5.2.5	Register Definition	186
5.2.6	Multiple Functions	186
5.2.7	Programming Guide	233
5.3	Peripheral Configuration Controller	234
5.3.1	Introduction	234
5.3.2	Features	234
5.3.3	Block Diagram	235
5.3.4	Register Definition	235
5.4	Keypad Scanner	235
5.4.1	Introduction	235

5.4.2	Features	235
5.4.3	Block Diagram	236
5.4.4	Keypad AC Timing	237
5.4.5	Register Definition	237
5.4.6	Programming Guide	238
5.5	UART	238
5.5.1	Introduction	238
5.5.2	Features	238
5.5.3	Block Diagram	239
5.5.4	UART AC Timing	239
5.5.5	Communication Protocol	241
5.5.6	Theory of Operations	241
5.5.7	Register Definition	247
5.5.8	Programming Guide	247
5.6	USB Host Controller	250
5.6.1	Introduction	250
5.6.2	Features	251
5.6.3	Block Diagram	251
5.6.4	Register Definition	252
5.6.5	Programming Guide	252
5.7	USB Device Controller	257
5.7.1	Introduction	257
5.7.2	Features	257
5.7.3	Block Diagram	258
5.7.4	Register Definition	258
5.7.5	Programming Guide	258
5.7.6	EPO Programming Guide	261
5.7.7	QMU Programming Guide	268

5.7.8	Reset	287
5.8	PCI Express	288
5.8.1	Introduction	288
5.8.2	Features	288
5.8.3	Block Diagram	288
5.8.4	Clock Scheme	290
5.8.5	Register Definition	291
5.8.6	Programming Guide	291
5.9	SSUSB_PCIE_PHY.....	296
5.9.1	Introduction	296
5.9.2	Features	296
5.9.3	Block Diagram	297
5.9.4	USB 2.0 High Speed Controller AC Timing	298
5.9.5	USB 3.0 High Speed Controller AC Timing	308
5.9.6	PCIe PHY AC Timing.....	308
5.9.7	Clock Relationship.....	309
5.9.8	Register Definition	309
5.9.9	Programming Guide	309
5.10	SPI Interface	309
5.10.1	Introduction	309
5.10.2	Features	310
5.10.3	Transmission Formats.....	312
5.10.4	Block Diagram	312
5.10.5	SPI AC Timing	313
5.10.6	Register Definition	315
5.10.7	Programming Guide	315
5.11	MMC and SD Controller	316
5.11.1	Introduction	316

5.11.2	Features	316
5.11.3	Block Diagram	318
5.11.4	MSDC AC Timing	319
5.11.5	Register Definition	329
5.11.6	Programming Guide	329
5.12	NAND Flash Interface	340
5.12.1	Introduction	340
5.12.2	Features	340
5.12.3	Block Diagram	340
5.12.4	NFI AC Timing	341
5.12.5	Register Definition	347
5.12.6	Programming Guide	347
5.13	Serial Flash Controller	347
5.13.1	Introduction	347
5.13.2	Features	347
5.13.3	Block Diagram	348
5.13.4	Flashif AC Timing	349
5.13.5	Register Definition	349
5.13.6	Programming Guide	350
5.14	AUXADC	355
5.14.1	Introduction	355
5.14.2	Features	355
5.14.3	Block Diagram	356
5.14.4	AUXADC AC Timing	357
5.14.5	Theory of Operations	359
5.14.6	Register Definition	359
5.14.7	Programming Guide	359
5.15	I2C/SCCB Controller	360

5.15.1	Introduction	360
5.15.2	Features	361
5.15.3	Block Diagram	361
5.15.4	I2C AC Timing	362
5.15.5	Manual Transfer Mode	364
5.15.6	Transfer Format Support	364
5.15.7	Register Definition	366
5.15.8	Programming Guide	366
5.16	Pulse-Width Modulation	367
5.16.1	Introduction	367
5.16.2	Specification	368
5.16.3	Features	368
5.16.4	Block Diagram	368
5.16.5	PWM AC Timing	369
5.16.6	Clock Relationship	369
5.16.7	Register Definition	370
5.16.8	Programming Guide	370
5.17	System Timer	375
5.17.1	Introduction	375
5.17.2	Features	375
5.17.3	Block Diagram	375
5.17.4	Register Definition	377
5.17.5	Programming Guide	377
5.18	General-Purpose Timer	377
5.18.1	Introduction	377
5.18.2	Features	378
5.18.3	Block Diagram	378
5.18.4	Register Definition	379

5.18.5 Programming Guide	379
5.19 General-Purpose Timer (GPT2)	383
5.19.1 Introduction	383
5.19.2 Features	383
5.19.3 Block Diagram	384
5.19.4 Register Definition	385
5.19.5 Programming Guide	385
5.20 Thermal Controller	389
5.20.1 Introduction	389
5.20.2 Features	389
5.20.3 Block Diagram	389
5.20.4 Register Definition	391
5.20.5 Programming Guide	391
5.21 Infrared Receiver	394
5.21.1 Introduction	394
5.21.2 Features	394
5.21.3 Block Diagram	395
5.21.4 IRRX AC Timing	395
5.21.5 Register Definition	396
5.21.6 Programming Guide	396
5.22 Ethernet QOS	397
5.22.1 Introduction	397
5.22.2 Features	397
5.22.3 Block Diagram	398
5.22.4 ETHER_QOS AC Timing	398
5.22.5 Clocks	403
5.22.6 Register Definition	405
5.22.7 Programming guide	405

6	Multimedia System	411
6.1	Audio System.....	411
6.1.1	Introduction	411
6.1.2	Features	411
6.1.3	Block Diagram	423
6.1.4	Audio AC Timing.....	423
6.1.5	Audio Clock Structure	426
6.1.6	Interface Timing	428
6.1.7	Register Definition	429
6.1.8	Programming Guide.....	429
6.2	MFlexGraphics.....	436
6.2.1	Introduction	436
6.2.2	Features	436
6.2.3	3D Features.....	436
6.2.4	Block Diagram	437
6.2.5	Register Definition	437
6.2.6	Programming Guide.....	437
6.3	Image Subsystem	437
6.3.1	Introduction	437
6.3.2	Features	438
6.3.3	Block Diagram	438
6.3.4	Register Definition	438
6.4	Camera Serial Interface	438
6.4.1	Introduction	438
6.4.2	Features	439
6.4.3	Block Diagram	439
6.4.4	Register Definition	440
6.4.5	Programming Guide.....	440

6.5	CAM Write Channel.....	440
6.5.1	Introduction	440
6.5.2	Features	440
6.5.3	Block Diagram	441
6.5.4	Register Definition	441
6.5.5	Programming Guide	441
6.6	H.264 Video Encoder.....	441
6.6.1	Introduction	441
6.6.2	Features	442
6.6.3	Block Diagram	442
6.6.4	Register Definition	443
6.7	Video Decoder.....	443
6.7.1	Introduction	443
6.7.2	Features	443
6.7.3	Adaptive MPEG De-blocking Filter	444
6.7.4	Error Handling	444
6.7.5	Block Diagram	445
6.7.6	Register Definition	446
6.8	JPEG Decoder	446
6.8.1	Introduction	446
6.8.2	Features	447
6.8.3	Block Diagram	448
6.8.4	Register Definition	449
6.8.5	Programming Guide	449
6.9	Image Resizer	451
6.9.1	Introduction	451
6.9.2	Features	451
6.9.3	Block Diagram	452

6.9.4	Register Definition	452
6.9.5	Programming Guide	453
6.10	Display Subsystem.....	453
6.10.1	Introduction	453
6.10.2	Features	453
6.10.3	Block Diagram	454
6.10.4	Register Definition	456
6.10.5	Programming Guide.....	456
6.11	Multimedia Data Path Read DMA	461
6.11.1	Introduction	461
6.11.2	Features	462
6.11.3	Block Diagram	462
6.11.4	Register Definition	463
6.11.5	Programming Guide	463
6.12	Multimedia Data Path Resizer (MDP_RSZ).....	466
6.12.1	Introduction	466
6.12.2	Features	467
6.12.3	Block Diagram	470
6.12.4	Register Definition	470
6.12.5	Programming Guide	470
6.13	Multimedia Data Path 2D Sharpness	473
6.13.1	Introduction	473
6.13.2	Features	473
6.13.3	Block Diagram	474
6.13.4	Register Definition	475
6.13.5	Programming Guide	475
6.14	Multimedia Data Path Write DMA	477
6.14.1	Introduction	477

6.14.2	Features	477
6.14.3	Block Diagram	477
6.14.4	Register Definition	478
6.14.5	Programming Guide	478
6.15	Multimedia Data Path Rotation DMA	479
6.15.1	Introduction	479
6.15.2	Features	479
6.15.3	Block Diagram	479
6.15.4	Register Definition	480
6.15.5	Programming Guide	480
6.16	Display Read Direct Memory Access	486
6.16.1	Introduction	486
6.16.2	Features	486
6.16.3	Block Diagram	487
6.16.4	Register Definition	487
6.16.5	Programming Guide	487
6.17	Display Write Direct Memory Access	489
6.17.1	Introduction	489
6.17.2	Features	489
6.17.3	Block Diagram	490
6.17.4	Register Definition	490
6.17.5	Programming Guide	490
6.18	Display Overlay.....	491
6.18.1	Introduction	491
6.18.2	Features	492
6.18.3	Block Diagram	492
6.18.4	Register Definition	492
6.18.5	Programming Guide	493

6.19	Display Color Engine (DISP_COLOR)	494
6.19.1	Introduction	494
6.19.2	Features	494
6.19.3	Block Diagram	494
6.19.4	Register Definition	495
6.19.5	Programming Guide	495
6.20	Display Adaptive Ambient Light Controller	498
6.20.1	Introduction	498
6.20.2	Features	498
6.20.3	Block Diagram	498
6.20.4	Color Correction	499
6.20.5	Gamma Correction	499
6.20.6	DRE	500
6.20.7	Register Definition	500
6.20.8	Programming Guide	500
6.21	Display GAMMA Processing Engine	501
6.21.1	Introduction	501
6.21.2	Features	501
6.21.3	Block Diagram	501
6.21.4	Color Correction	502
6.21.5	Gamma Correction	502
6.21.6	Register Definition	503
6.21.7	Programming Guide	503
6.22	UFOe Processor	503
6.22.1	Introduction	503
6.22.2	Features	503
6.22.3	Block Diagram	503
6.22.4	Register Definition	504

6.22.5 Programming Guide	504
6.23 Display OD Engine	504
6.23.1 Introduction	504
6.23.2 Features	504
6.23.3 Block Diagram	505
6.23.4 Register Definition	505
6.23.5 Programming Guide	505
6.24 Digital Parallel Interface	505
6.24.1 Introduction	505
6.24.2 Features	505
6.24.3 DPI Block Diagram.....	506
6.24.4 Register Definition	506
6.24.5 Programming Guide	506
6.25 Display Serial Interface.....	507
6.25.1 Introduction	507
6.25.2 Features	507
6.25.3 Block Diagram	508
6.25.4 DSI AC Timing	509
6.25.5 Register Definition	513
6.25.6 Programming Guide	513
6.25.7 Command Queue	516
6.25.8 Clock Lane Non-continuous Low-power Control	524
6.26 MIPI TX Configuration Module.....	528
6.26.1 Introduction	528
6.26.2 Features	528
6.26.3 Block Diagram	529
6.26.4 Register Definition	529
6.27 DISP LVDS Encoder	529

6.27.1	Introduction	529
6.27.2	Features	530
6.27.3	Block Diagram	530
6.27.4	LVDS AC Timing	531
6.27.5	LVDS Clock.....	532
6.27.6	LVDS Encoder Standard	534
6.27.7	DC/AC Parameters	536
6.27.8	Chanel Swap.....	537
6.27.9	Register Definition	537
6.27.10	Programming Guide	537
6.28	DISP_PWM Generator.....	537
6.28.1	Introduction	537
6.28.2	Features	537
6.28.3	Block Diagram	538
6.28.4	Register Definition	538
6.28.5	Programming Guide	538
6.29	Display Mutex.....	538
6.29.1	Introduction	538
6.29.2	Features	538
6.29.3	Block Diagram	538
6.29.4	Register Definition	539
6.29.5	Programming Guide	539
6.30	Smart Multimedia Interface	541
6.30.1	Introduction	541
6.30.2	Features	541
6.30.3	Block Diagram	541
6.30.4	Register Definition	542
6.31	Multimedia Memory Management Unit (MM_IOMMU).....	542

6.31.1	Introduction	542
6.31.2	Features	544
6.31.3	Block Diagram	546
6.31.4	Register Definition	546
6.32	Video De-interlacing.....	546
6.32.1	Introduction	546
6.32.2	Features	547
6.32.3	Block Diagram	547
6.32.4	Register Definition	548
6.32.5	Programming Guide	548
6.33	Display Format	555
6.33.1	Introduction	555
6.33.2	Features	555
6.33.3	Block Diagram	555
6.33.4	Register Definition	556
6.33.5	Programming Guide	556
6.34	Noise Reduction (NR)	556
6.34.1	Introduction	556
6.34.2	Features	557
6.34.3	NR Block Diagram.....	557
6.34.4	Register Definition	558
6.34.5	Programming Guide	558
6.35	TV Decoder	560
6.35.1	Introduction	560
6.35.2	Features	560
6.35.3	Block Diagram	561
6.35.4	Programming Guide	561
6.35.5	Register Definition	562

6.36 DISP_MONITOR	562
6.36.1 Introduction	562
6.36.2 Features	563
6.36.3 Block Diagram	563
6.36.4 Register Definition	564
6.36.5 Programming Guide	564
6.36.6 IRQ Guide	565
Document Revision History.....	566
Exhibit 1 Terms and Conditions.....	576

List of Tables

Table 2-1 Pin Coordinate Using LPDDR4	50
Table 2-2 Acronym for Pin Type	57
Table 2-3 Detailed Pin List.....	57
Table 2-4 Absolute Maximum Ratings	71
Table 2-5 Recommended Operating Conditions for Power Supply.....	72
Table 2-6 LPDDR4 AC Timing Parameter for External Memory Interface	76
Table 2-7 General 1.8V IO DC Timing Parameters.....	77
Table 2-8 General 3.3V IO DC Timing Parameters.....	78
Table 2-9 VCC33IO_MSDC1/2 DC Timing Parameters.....	78
Table 2-10 VCC18IO for MSDCOE/MSDC3 DC Timing Parameters	79
Table 2-11 MSDC Note.....	80
Table 2-12 MSDC Overshoot/Undershoot	80
Table 2-13 Constant Tied Pins.....	80
Table 2-14 Signals Details.....	81
Table 2-15 Definitions of AUXADC Channels.....	84
Table 2-16 AUXADC Specifications.....	84
Table 2-17 XTAL Specifications.....	85
Table 2-18 ARMCA35PLL Specifications.....	86
Table 2-19 ARMCA72PLL Specifications.....	87
Table 2-20 MAINPLL Specifications.....	87
Table 2-21 MMPLL Specifications	87
Table 2-22 UNIVPLL Specifications.....	87
Table 2-23 MSDCPLL Specifications	88
Table 2-24 MSDCPLL2 Specifications	88
Table 2-25 VENCPLL Specifications	88
Table 2-26 VCODECPPLL Specifications.....	88
Table 2-27 TVDPLL Specifications	89
Table 2-28 MMPLL Specifications	89
Table 2-29 APLL1 Specifications.....	89
Table 2-30 APLL2 Specifications.....	89
Table 2-31 LVDSPLL Specifications	89
Table 2-32 ETHERPLL Specifications.....	90
Table 2-33 CVBSPLL Specifications	90
Table 2-34 CVBSREFPLL Specifications	90
Table 2-35 Temperature Sensor Specifications	91
Table 2-36 AACD Specification	91
Table 2-37 Thermal Operating Specifications	96
Table 2-38 Scalable Platform Solution	96
Table 3-1 XO Specifications	98
Table 3-2 ARMCA35PLL Control Sequence	100
Table 3-3 ARMCA72PLL Control Sequence	101
Table 3-4 VENCPLL Control Sequence.....	101
Table 3-5 LVDSPLL Control Sequence	102
Table 3-6 LVDSPLL2 Control Sequence	102
Table 3-7 MAINPLL Control Sequence	103
Table 3-8 UNIVPLL Control Sequence	103
Table 3-9 VCODECPPLL Control Sequence	104
Table 3-10 MMPLL Control Sequence	104
Table 3-11 APLL1 Control Sequence	105

Table 3-12 APLL2 Control Sequence	105
Table 3-13 MSDCPLL Control Sequence	106
Table 3-14 MSDCPLL2 Control Sequence	106
Table 3-15 ETHERPLL Control Sequence	107
Table 3-16 CVBSPLL Control Sequence	107
Table 3-17 CVBSREFPLL Control Sequence	108
Table 3-18 PLL Related Control	113
Table 3-19 Clock Gating Settings.....	114
Table 3-20 Clock Delay Comment	116
Table 3-21 Clock Mux.....	117
Table 3-22 Hopping PLLs	131
Table 3-23 Power Domains in MCUSYS.....	138
Table 3-24 VCORE MTCMOS Domains	138
Table 3-25 Power Mode Scenarios	138
Table 3-26 RTC Clock Specification	139
Table 4-1 Interrupt Request List for MCUSYS.....	143
Table 4-2 Power State	146
Table 4-3 JTAG Timing Parameter	149
Table 4-4 Clock Frequency of Major AP System Bus Fabric	149
Table 4-5 De-bounce Time in MT2712	153
Table 4-6 External Interrupt Request Signal Connection	154
Table 4-7 On-chip Memory Controller Memory Map	157
Table 4-8 CM4 HW Interrupt.....	165
Table 5-1 Memory Map of MT2712	167
Table 5-2 Boot Section Memory Map of MT2712	168
Table 5-3 Infrastructure System Memory Map of MT2712	169
Table 5-4 Peripheral System Map of MT2712	170
Table 5-5 Audio System Memory Map.....	172
Table 5-6 MFG System Memory Map	172
Table 5-7 MMSYS System Memory Map.....	172
Table 5-8 Image System Memory Map	174
Table 5-9 Video Decoding System Memory Map	175
Table 5-10 Video Encoding System Memory Map	175
Table 5-11 JPEGDEC System Memory Map.....	175
Table 5-12 1.8V GPIO Pads DC Parameters	178
Table 5-13 1.8V I2C GPIO Pads DC Parameters	179
Table 5-14 180D33V GPIO Pads DC Parameters	179
Table 5-15 MSDC 1.8V IO DC Timing	180
Table 5-16 MSDC 180D33V IO DC Timing	181
Table 5-17 RGMII I/O Timing.....	182
Table 5-18 NOR Flash I/O DC Timing.....	182
Table 5-19 Schmitt Input I/O DC Timing	183
Table 5-20 AC Parameters 1.8 V GPIO Mode	184
Table 5-21 AC Parameters 3.3V GPIO Mode	185
Table 5-22 GPIO Control	187
Table 5-23 MT2712 Pins Reset Status & Driving	224
Table 5-24 MT2712 Interrupt Source.....	229
Table 5-25 Keypad IO AC Timing	237
Table 5-26 Keypad Setting Flow	238
Table 5-27 UART Transmit Timing Parameters	240
Table 5-28 UART Receive Timing Parameters	240
Table 5-29 UART Interrupt Control Bits and Interrupt Factors.....	242

Table 5-30 UART Interrupt Types	242
Table 5-31 UART Register Map	243
Table 5-32 Bit Extend Number Reference	246
Table 5-33 Suggestion for UART Baud Rate Setting	247
Table 5-34 UART Baud Rate Setting Example	247
Table 5-35 UART HW Initialization	248
Table 5-36 USB Host Initialization Flow	252
Table 5-37 Endpoint Context	254
Table 5-38 Extra Defined Fields for Endpoint Context	255
Table 5-39 USB Device Initialization Flow	258
Table 5-40 Suggested EP0 Programming Sequence	259
Table 5-41 Suggested TX EPn Programming Sequence (e.g. EP1)	260
Table 5-42 RX EPn Programming Sequence (e.g. EP2)	260
Table 5-43 PCIe Interrupt	291
Table 5-44 PCIe MAC Memory Map	295
Table 5-45 SPI Master Electrical Specifications	314
Table 5-46 SPI Slave Electrical Specifications	314
Table 5-47 SPI Slave Electrical Specifications	314
Table 5-48 eMMC5.0 Timing Parameter (backward-compatible device interface)	319
Table 5-49 eMMC5.0 Timing Parameter (high-speed mode)	320
Table 5-50 eMMC5.0 Timing Parameter (high-speed dual rate mode)	321
Table 5-51 eMMC5.0 Timing Parameter (high-speed 200 mode)	322
Table 5-52 eMMC5.0 Output Parameter (high-speed 400 mode host output)	323
Table 5-53 eMMC5.0 Output Parameter (high-speed 400 mode host input)	324
Table 5-54 SD Timing Parameter (default-speed mode)	324
Table 5-55 SD Timing Parameter (high-speed mode/SDR12/SDR25 mode)	325
Table 5-56 SD Timing Parameter (SDR50 mode)	326
Table 5-57 SD Timing Parameter (DDR50 mode)	327
Table 5-58 SD Timing Parameter (SDR104 mode)	328
Table 5-59 S18R and S18A Combinations	338
Table 5-60 NFI ONFI1.0 Command/Address/Data Write Access Timing	341
Table 5-61 NFI ONFI1.0 Read Access Timing	343
Table 5-62 NFI Toggle1.0 Write Access Timing	344
Table 5-63 NFI Toggle1.0 Read Access Timing	346
Table 5-64 Serial NOR Flash Interface AC Timing Diagram Key (Tentative)	349
Table 5-65 Definitions of AUXADC Channels	357
Table 5-66 AUXADC Specifications	358
Table 5-67 Temperature Sensor Specifications	358
Table 5-68 Immediate Mode Programming Sequence	360
Table 5-69 Background Detection Programming Sequence	360
Table 5-70 I2C Electrical Characteristic Specifications (Current Chip)	362
Table 5-71 I2C Electronic Characteristic Specifications (Standard Protocol)	363
Table 5-72 Characteristics of the PWM Timing	369
Table 5-73 Old Mode Setting Procedures	370
Table 5-74 FIFO Mode Setting Procedures	370
Table 5-75 Memory Mode Setting Procedures	372
Table 5-76 Random Mode Setting Procedures	373
Table 5-77 sys_timer Setting Flow	377
Table 5-78 Operation Modes of GPT	378
Table 5-79 APXGPT ONE-SHOT Mode Setting Flow	379
Table 5-80 APXGPT REPEAT Mode Setting Flow	380
Table 5-81 APXGPT KEEP-GO Mode Setting Flow	381

Table 5-82 APXGPT FREERUN Mode Setting Flow	382
Table 5-83 GPT6 64-bit Counter Read Flow	383
Table 5-84 Operation Mode of GPT	384
Table 5-85 APXGPT ONE-SHOT Mode Setting Flow	385
Table 5-86 APXGPT REPEAT Mode Setting Flow	386
Table 5-87 APXGPT KEEP-GO Mode Setting Flow	387
Table 5-88 APXGPT FREERUN Mode Setting Flow	388
Table 5-89 GPT6 64-bit Counter Read Flow	389
Table 5-90 IRRX Decode Programming Flow	396
Table 5-91 MII Interface Timing Requirement	398
Table 5-92 RMII Interface Timing Requirement	399
Table 5-93 RGMII Interface Timing Requirement	400
Table 5-94 MDIO Interface Timing Requirement	403
Table 5-95 Clock Requirements	403
Table 6-1 TDM Sample Rates	413
Table 6-2 I2S AC Timing Characteristics	423
Table 6-3 TDM AC Timing Characteristics	424
Table 6-4 PCM AC Timing Characteristics	425
Table 6-5 I2S-IQ AC Timing Characteristics	426
Table 6-6 I2S Interface Timing Delay	428
Table 6-7 PCM Interface Timing Delay	428
Table 6-8 TDM Interface Timing Delay	429
Table 6-9 ASRC Programming Guide	433
Table 6-10 IRQ Information	434
Table 6-11 Main Features	442
Table 6-12 Resizer Functional Specifications	466
Table 6-13 Resizer Hardware Specifications	466
Table 6-14 MDP_WROT_OFST_ADDR Settings for Rotation/Flip	482
Table 6-15 VIDEO UV SEL for YUV420 Format Cooperated with CRSP	482
Table 6-16 DISP_OVL Features	492
Table 6-17 UFO Modes	504
Table 6-18 UFO Width/Height Reference in Each Mode	504
Table 6-19 DSI LP Driver Characteristics	509
Table 6-20 DSI High-Speed Driver Characteristics DPHY	510
Table 6-21 Full Speed Electrical and Timing Information	512
Table 6-22 CSI Low-Power Receiver Characteristics DPHY	513
Table 6-23 Sequence to Enable Module Clock	513
Table 6-24 Sequence to Enable High-Speed Clock	513
Table 6-25 Sequence to Exit Ultra-low Power Mode on Clock Lane	514
Table 6-26 Sequence of Sleep-in Control (Entering Ultra-low Power Mode)	514
Table 6-27 Sequence of Sleep-out Control (Exiting Ultra-low Power Mode)	514
Table 6-28 DPHY Timing Parameter Register Settings	515
Table 6-29 Configuration Field Description of Main Instruction	517
Table 6-30 Type-0 Instruction Format	518
Table 6-31 Type-0 TX Example	518
Table 6-32 Type-1 Instruction Format	519
Table 6-33 Type-1 TX Example	519
Table 6-34 Type-2 Instruction Format	519
Table 6-35 Type-2 TX Example	519
Table 6-36 Type-3 Instruction Format	520
Table 6-37 Type-3 TX Example	520
Table 6-38 Command Mode Status in Debugging Register	521

Table 6-39 Example of TE Signaling Detection	526
Table 6-40 Example of External TE Pin Detection	526
Table 6-41 Example of Short Packet Transmission in Video Mode	527
Table 6-42 Example of Long Packet Transmission in Video Mode	527
Table 6-43 Dual DSI Mode Control Sequence	528
Table 6-44 LVDS Specifications.....	532
Table 6-45 DC/AC Parameters.....	536
Table 6-46 Related Register Fields for Frame Buffer Address Swap Control.....	558
Table 6-47 Corresponding Register Fields Configuration Values for Address Swap Mode	559

List of Figures

Figure 1-1 MT2712 Block Diagram	48
Figure 2-1 Ball Map View for LPDDR4	49
Figure 2-2 Basic Timing Parameter for LPDDR4 Commands	75
Figure 2-3 Basic Timing Parameter for LPDDR4 Write	75
Figure 2-4 Basic Timing Parameter for LPDDR4 Read	75
Figure 2-5 Power-On/Off Sequence with XTAL	81
Figure 2-6 AUXADC Block Diagram	84
Figure 2-7 PLLGP Block Diagram	86
Figure 2-8 AADC Input Specification and Connection with Chip MT2712	92
Figure 2-9 CVBS Application Reference Circuit	93
Figure 2-10 Outlines and Dimensions of HFC FBGA 23mm*23mm, 852-ball, 0.65mm Pitch Package	96
Figure 2-11 Top Mark of MT2712	97
Figure 3-1 Apmixedsys Block Diagram	99
Figure 3-2 Clock Architecture Block Diagram	109
Figure 3-3 Example of Clock Multiplexer	109
Figure 3-4 PLLGP Block Diagram	110
Figure 3-5 PLL Divider1 Block Diagram	111
Figure 3-6 PLL Divider2 Block Diagram	112
Figure 3-7 PLL Divider3 Block Diagram	113
Figure 3-8 ABIST FMETER Structure	115
Figure 3-9 CKGEN Structure	115
Figure 3-10 FHCTL Block Diagram	132
Figure 3-11 Hopping Enable Control	133
Figure 3-12 Block Diagram of TOPRGU	134
Figure 3-13 Block Diagram of MT2712 Power Domain	137
Figure 3-14 RTC Block Diagram	140
Figure 3-15 RTC Register Map Sketch	141
Figure 4-1 MCUSYS Block Diagram	143
Figure 4-2 Power Domain	146
Figure 4-3 Debug System Block Diagram in MT2712	148
Figure 4-4 JTAG Timing Diagram	148
Figure 4-5 SYS_CIRQ System Level Block Diagram	150
Figure 4-6 System Interrupt Controller Architecture Block Diagram	151
Figure 4-7 Software Programming Flow Chart	152
Figure 4-8 Block Diagram of External Interrupt Controller in MT2712	153
Figure 4-9 DCM in Action	156
Figure 4-10 Infra Bus Fabric	157
Figure 4-11 On-chip Memory Controller Block Diagram	158
Figure 4-12 AP_DMA System Level Block Diagram	160
Figure 4-13 System Level Block Diagram of CQ_DMA	162
Figure 4-14 Block Diagram of CM4SYS	164
Figure 4-15 Memory Mapping of CM4SYS	165
Figure 5-1 DRAM Physical Address Mapping in Legacy Mode	176
Figure 5-2 DRAM Physical Address Mapping in 4GB Mode	176
Figure 5-3 MT2712 GPIO Controller	177
Figure 5-4 IO Types	178
Figure 5-5 Pericfg Controller Block Diagram	235
Figure 5-6 Keypad Block Diagram	236
Figure 5-7 7x7 Keypad Scan Waveform	237

Figure 5-8 One Key Pressed with De-bounce Mechanism Denoted	237
Figure 5-9 UART Block Diagram	239
Figure 5-10 UART Transmit Timing Diagram	240
Figure 5-11 UART Receive Timing Diagram	240
Figure 5-12 UART Communication Protocol	241
Figure 5-13 UART Data Transmission with Polling the THRE Bit Status	249
Figure 5-14 UART Data Reception with Polling the DR Bit Status	250
Figure 5-15 USB Host Architecture	251
Figure 5-16 Relation between Extra Defined Parameters and Micro Frames	255
Figure 5-17 Relation between bCSCount Parameters and Micro Frames	255
Figure 5-18 (a) Standard xHCI_add_endpoint () Flow (b) Patched xHCI_add_endpoint () Flow	256
Figure 5-19 (a) Standard xHCI_drop_endpoint () Flow (b) Patched xHCI_drop_endpoint () Flow	257
Figure 5-20 USB Device Architecture	258
Figure 5-21 EPO State	261
Figure 5-22 EPO IDLE Flow Chart	262
Figure 5-23 EPO IDLE	263
Figure 5-24 EPO TX Mode Flow Chart	264
Figure 5-25 EPO TX Mode Flow Chart	265
Figure 5-26 EPO RX Mode Flow Chart	266
Figure 5-27 EPO IDLE for RX Mode	267
Figure 5-28 GPD/BD/Bus Transfer	268
Figure 5-29 GPD/BD with Extension	268
Figure 5-30 TX GPD/BD Format	270
Figure 5-31 RX GPD/BD Format	271
Figure 5-32 TXQ Programming Flow	274
Figure 5-33 TXQ Initialization Flow	275
Figure 5-34 TXQ STALL Flow	276
Figure 5-35 TXQ Short Packet (I) Flow	277
Figure 5-36 TXQ Short Packet (II) Flow	278
Figure 5-37 TXQ Error Handling Flow	279
Figure 5-38 TXQ GPD Execution Flow	280
Figure 5-39 RX Programming Flow	281
Figure 5-40 RX Initialization Flow	282
Figure 5-41 RX STALL Flow	283
Figure 5-42 RX Drop Packet Flow	284
Figure 5-43 RX Error Handling Flow	285
Figure 5-44 RX GPD Execution Flow	286
Figure 5-45 MT2712 PCIe Block Diagram	289
Figure 5-46 PCIe MAC Block Diagram	290
Figure 5-47 PCIe MAC Clock Domain Partition Illustration	291
Figure 5-48 SSUSB_PCIE_PHY Block Diagram	297
Figure 5-49 USB 2.0 Top Block Diagram	298
Figure 5-50 Data Signal Rise and Fall Time	299
Figure 5-51 Full-Speed Load	299
Figure 5-52 Low-Speed Port Loads	300
Figure 5-53 Measurement Planes	300
Figure 5-54 Template 1	302
Figure 5-55 Template 2	303
Figure 5-56 Template 3	304
Figure 5-57 Template 4	305
Figure 5-58 Template 5	306
Figure 5-59 Template 6	307

Figure 5-60 Transmitter/Receiver Test Fixture.....	308
Figure 5-61 SSUSB_PCIE_PHY Clocking Block Diagram.....	309
Figure 5-62 Pin Connection between SPI Master and SPI Slave	309
Figure 5-63 Operation Flow with or without PAUSE Mode	310
Figure 5-64 CS_N De-assert Mode.....	310
Figure 5-65 Timing Sequence of SPI_SLAVE_MISO Line When CPOL=0 and CPHA=0.....	311
Figure 5-66 SPI Transmission Formats	312
Figure 5-67 SPI Master Block Diagram.....	312
Figure 5-68 SPI Slave Block Diagram	313
Figure 5-69 SPI Master/Slave Timing Diagram.....	313
Figure 5-70 SCK Generation	315
Figure 5-71 MSDC Block Diagram	318
Figure 5-72 eMMC5.0 Timing Diagram (backward-compatible device interface)	319
Figure 5-73 eMMC5.0 Timing Diagram (high-speed mode)	320
Figure 5-74 eMMC5.0 Timing Diagram (high-speed dual rate mode)	321
Figure 5-75 eMMC5.0 Timing Diagram (high-speed 200 mode).....	322
Figure 5-76 eMMC5.0 Timing Diagram (high-speed 400 mode host output).....	323
Figure 5-77 eMMC5.0 Timing Diagram (high-speed 400 mode host input)	323
Figure 5-78 SD Timing Diagram (default-speed mode).....	324
Figure 5-79 SD Timing Diagram (high-speed mode/SDR12/SDR25 mode)	325
Figure 5-80 SD Timing Diagram (SDR50 mode).....	326
Figure 5-81 SD Timing Diagram (DDR50 mode)	327
Figure 5-82 SD Timing Diagram (SDR104 mode).....	328
Figure 5-83 Single Block and Multi Block Read in PIO Mode	329
Figure 5-84 Single Block and Multi Block Write in PIO Mode	330
Figure 5-85 Single Block and Multi Block Read in DMA Mode	331
Figure 5-86 Single Block and Multi Block Write In DMA Mode	331
Figure 5-87 eMMC Boot Mode 0 Timing Diagram	332
Figure 5-88 eMMC Boot Mode 1 Timing Diagram	333
Figure 5-89 Basic DMA.....	335
Figure 5-90 Descriptor and Enhanced DMA	336
Figure 5-91 Voltage Switch Command	337
Figure 5-92 ACM41 Timing Followed by Voltage Switch Sequence	337
Figure 5-93 UHS-I Host Initialization Flow Chart	338
Figure 5-94 Signal Voltage Switch Sequence	339
Figure 5-95 NFI Block Diagram.....	341
Figure 5-96 NFI ONFI1.0 Command Input Cycle	342
Figure 5-97 NFI ONFI1.0 Address Input Cycle	342
Figure 5-98 NFI ONFI1.0 Consecutive Data Write Cycles	343
Figure 5-99 NFI ONFI1.0 Serial Read Cycle.....	344
Figure 5-100 NFI Toggle1.0 Command Input Cycle	345
Figure 5-101 NFI Toggle1.0 Address Input Cycle	345
Figure 5-102 NFI Toggle1.0 Data Write Cycle.....	346
Figure 5-103 NFI Toggle1.0 Data Read Cycle.....	347
Figure 5-104 Flashif Block Diagram.....	348
Figure 5-105 Serial NOR Flash Interface AC Timing	349
Figure 5-106 RDID Operation Sequence	350
Figure 5-107 SE and BE Operation Sequence	350
Figure 5-108 CE Operation Sequence	351
Figure 5-109 Program Operation Sequence	352
Figure 5-110 Read Operation Sequence	353
Figure 5-111 Quad Output Read Mode Sequence (Address is sent in single-bit mode).....	354

Figure 5-112 Quad I/O Read Mode Sequence (Address is sent in 4-bit mode)	354
Figure 5-113 AXUADC Block Diagram	356
Figure 5-114 AUXADC Block Diagram	357
Figure 5-115 Theory of AUXADC Operation.....	359
Figure 5-116 I2C Block Diagram.....	361
Figure 5-117 Definition of Timing for the F/S Mode Devices on the I2C-Bus	362
Figure 5-118 Common Transfer Programmable Parameters	367
Figure 5-119 Output Waveform Timing Programmable Parameters	367
Figure 5-120 PWM Generation Procedure	367
Figure 5-121 PWM Block Diagram	368
Figure 5-122 PWM Timing	369
Figure 5-123 PWM Module Clock Structure	369
Figure 5-124 sys_timer Block Diagram	375
Figure 5-125 Behavior of sys_timer Counter Timeout Value	376
Figure 5-126 APXGPT Block Diagram	378
Figure 5-127 APXGPT Block Diagram	384
Figure 5-128 System Level Thermal Controller Block Diagram	390
Figure 5-129 System Temperature Measurement	390
Figure 5-130 Programming Flow.....	391
Figure 5-131 Immediate Measurement Programming Flow	393
Figure 5-132 Pulse-width Coding.....	394
Figure 5-133 Bi-phase Coding	394
Figure 5-134 Infrared Receiver Block Diagram	395
Figure 5-135 NEC Protocol	395
Figure 5-136 RC5 Protocol	396
Figure 5-137 RC6 Protocol	396
Figure 5-138 RCMM Protocol	396
Figure 5-139 Ethernet with AVB MAC Controller Block Diagram.....	398
Figure 5-140 MII Interface Timing Diagram	399
Figure 5-141 RMII Interface Timing Diagram.....	400
Figure 5-142 RGMII Interface Timing Diagram	401
Figure 5-143 Block Diagram of Delay Macro.....	402
Figure 5-144 MDIO Interface Timing Diagram	403
Figure 5-145 RMII Clock Structure	404
Figure 5-146 RGMII Clock Structure	405
Figure 6-1 I2S Protocol.....	412
Figure 6-2 TDM Protocol.....	414
Figure 6-3 Waveform of I2S Format in PCM Interface	415
Figure 6-4 Waveform of EIAJ Format in PCM Interface	416
Figure 6-5 Waveform of Mode A Format in PCM Interface	416
Figure 6-6 Waveform of Mode B Format in PCM Interface	417
Figure 6-7 Waveform of CASE0 Format in I2S-IQ Interface.....	418
Figure 6-8 Waveform of CASE1 Format in I2S-IQ Interface.....	418
Figure 6-9 Waveform of CASE2 Format in I2S-IQ Interface.....	419
Figure 6-10 Waveform of CASE3 Format in I2S-IQ Interface.....	419
Figure 6-11 Waveform of CASE4 Format in I2S-IQ Interface.....	419
Figure 6-12 Waveform of CASE5 Format in I2S-IQ Interface	420
Figure 6-13 Waveform of CASE6 Format in I2S-IQ Interface	420
Figure 6-14 Waveform of CASE7 Format in I2S-IQ Interface	420
Figure 6-15 Audio Safety Feature Diagram	421
Figure 6-16 Audio Safety Feature Diagram	421
Figure 6-17 PCM Format of Audio Warning Sound	422

Figure 6-18 CRC of Audio Warning Sound	422
Figure 6-19 Audio System Block Diagram	423
Figure 6-20 I2S Mode Timing Diagram	424
Figure 6-21 TDM Mode Timing Diagram.....	425
Figure 6-22 PCM Mode Timing Diagram.....	425
Figure 6-23 I2S-IQ Slave Mode Timing Diagram	426
Figure 6-24 Audio System Clock Structure.....	427
Figure 6-25 I2S Interface Timing	428
Figure 6-26 PCM Interface Timing	428
Figure 6-27 TDM Interface Timing	429
Figure 6-28 MFlexGraphics Block Diagram	437
Figure 6-29 IMGSYS_TOP Function Blocks	438
Figure 6-30 CSI Block Diagram	439
Figure 6-31 CSI CLK Domain.....	439
Figure 6-32 Settle Delay Calculations	440
Figure 6-33 WR_CHANNEL Block Diagram	441
Figure 6-34 Video Encoder Procedure	442
Figure 6-35 Video Encoder Block Diagram.....	443
Figure 6-36 De-blocking Image	444
Figure 6-37 Error Handling Example 1	445
Figure 6-38 Error Handling Example 2	445
Figure 6-39 VDEC Block Diagram	446
Figure 6-40 Basic Structure of JPEG Files	447
Figure 6-41 Supported Sampling Format of Input JPEG File	448
Figure 6-42 jpgdec Block Diagram	449
Figure 6-43 Filter Types of Image Resizer in Horizontal Scaling	451
Figure 6-44 Filter Types of Image Resizer in Vertical Scaling.....	452
Figure 6-45 IMG_RSZ Block Diagram	452
Figure 6-46 MDP Functional Blocks of Multimedia Partition.....	455
Figure 6-47 DISP Functional Blocks of Multimedia Partition	455
Figure 6-48 IMGSYS_TOP Functional Blocks	456
Figure 6-49 Stream Programming Sequence	458
Figure 6-50 Detailed Mutex Waveform.....	458
Figure 6-51 DISP_MONITOR Function Blocks	459
Figure 6-52 IRQ Status	460
Figure 6-53 DISP_MONITOR Clock Diagram.....	461
Figure 6-54 Support Clipping from Source Frame Buffer.....	462
Figure 6-55 MDP_RDMA Block Diagram.....	463
Figure 6-56 SW Control Flow	465
Figure 6-57 HW FSM and Mutex Control Flow	465
Figure 6-58 Separate 1D FIR Operations.....	467
Figure 6-59 4-Tap Cubic Block Diagram	468
Figure 6-60 Cubic Accumulation (scaling ratio = 1x)	469
Figure 6-61 Cubic Accumulation (scaling ratio = 1/2x)	469
Figure 6-62 MDP_RSZ Block Diagram	470
Figure 6-63 Suggested Algorithm vs. Supported Size	471
Figure 6-64 Programming Guide.....	473
Figure 6-65 Visual Effect of 2-dimensional Sharpness	474
Figure 6-66 Visual Effect of Peaking by Color	474
Figure 6-67 Display 2D Sharpness Block Diagram	474
Figure 6-68 Sharpness Core Block Diagram	475
Figure 6-69 Usage Scenarios	475

Figure 6-70 Gain Curve Control of Sharpness	476
Figure 6-71 Demo Curve Control of Luma Adjustment	476
Figure 6-72 DISP_WDMA Block Diagram	477
Figure 6-73 Typical MDP_WDMA Programming Sequence	479
Figure 6-74 MDP_WROT Block Diagram	480
Figure 6-75 Firmware Settings of OFST_ADDR in 0° Rotation (scan-line)	481
Figure 6-76 Firmware Settings of OFST_ADDR in 90° Rotation (scan-line)	481
Figure 6-77 Firmware Settings of OFST_ADDR in 180° Rotation (scan-line)	481
Figure 6-78 Firmware Settings of OFST_ADDR in 270° Rotation (scan-line)	482
Figure 6-79 Probability of Crossing 256-byte Boundary by Random Access	485
Figure 6-80 Scan-line Request without Considering 256-byte Boundary	485
Figure 6-81 Scan-line Request Considering 256-byte Boundary	486
Figure 6-82 4-16 Burst in 270° Rotation with/without Flip	486
Figure 6-83 Block Diagram of RDMA Engine	487
Figure 6-84 General Programming Sequence	488
Figure 6-85 Basic Memory Mode Configuration	488
Figure 6-86 Programmable Color Matrix	489
Figure 6-87 INPUT_FORMAT/SWAP	489
Figure 6-88 DISP_WDMA Block Diagram	490
Figure 6-89 Typical DISP_WDMA Programming Sequence	491
Figure 6-90 DISP_OVL Block Diagram	492
Figure 6-91 Typical DISP_OVL Programming Procedure	493
Figure 6-92 Color Processor	494
Figure 6-93 Hue Phase Distribution	495
Figure 6-94 Partial Hue Adjustment Example	495
Figure 6-95 Contrast and Brightness Adjustment Example	496
Figure 6-96 Chroma Boost Adjustment Example	496
Figure 6-97 Example of Partial Saturation for One Hue Phase	497
Figure 6-98 Illustration of Global Saturation Adjustment	497
Figure 6-99 AAL Engine Block Diagram	498
Figure 6-100 Color Correction Block Diagram	499
Figure 6-101 AAL Processor Block Diagram	499
Figure 6-102 Visual Effect of Color Correction	499
Figure 6-103 Visual Effect of Gamma Correction	500
Figure 6-104 Visual Effect of DRE	500
Figure 6-105 GAMMA Engine Block Diagram	501
Figure 6-106 Color Correction Block Diagram	502
Figure 6-107 Visual Effect of Color Correction	502
Figure 6-108 Visual Effect of Gamma Correction	502
Figure 6-109 Usage Scenarios	503
Figure 6-110 UFOe Engine Block Diagram	504
Figure 6-111 DISP_OD Engine	505
Figure 6-112 DPI Engine Block Diagram	506
Figure 6-113 Programming Flow Diagram	507
Figure 6-114 Block Diagram of DSI Modules in MMSYS	508
Figure 6-115 D-PHY Signaling Levels	509
Figure 6-116 Possible ΔVCMTX and ΔVOD Distortions of the Single-ended HS Signals	510
Figure 6-117 HS Receiver Implementation Example	511
Figure 6-118 Input Glitch Rejection of Low-Power Receivers	511
Figure 6-119 Data to Clock Timing Definitions	512
Figure 6-120 Registers for Data Lane Timing Parameters	515
Figure 6-121 Registers for Clock Lane Timing Parameters	516

Figure 6-122 Register for BTA Timing Parameters.....	516
Figure 6-123 DS1 Command Queue Instruction Format.....	517
Figure 6-124 Non-burst Transmission: Sync-pulse Mode	522
Figure 6-125 Sync-pulse Mode Line Period	522
Figure 6-126 Sync-pulse Mode Word-count Parameters	522
Figure 6-127 Non-burst Transmission: Sync-event Mode	523
Figure 6-128 Sync-event Mode Line Period	523
Figure 6-129 Sync-event Mode Word-count Parameters	523
Figure 6-130 Burst Mode Transmission	524
Figure 6-131 Blanking Packet in Video Mode Transmission.....	525
Figure 6-132 Low-power Counter Settings for Video Mode Low-power Control	525
Figure 6-133 MIPI_TX_CONFIG Module's Block Diagram	529
Figure 6-134 LVDS in MMSYS Path Block Diagram	530
Figure 6-135 LVDS Encoder Block Diagram	531
Figure 6-136 Block Diagram of LVDS Encoder Datapath	531
Figure 6-137 LVDS Waveform	532
Figure 6-138 Overview of LVDS Clock Path	532
Figure 6-139 Block Diagram of LVDSPLL	533
Figure 6-140 Clock Path of VOPLL.....	534
Figure 6-141 Single Link 6-bit VESA Standard	535
Figure 6-142 Single Link 8-bit DISM Standard.....	535
Figure 6-143 Single Link 8-bit VESA Standard	535
Figure 6-144 Dual Link 8-bit VESA Standard	536
Figure 6-145 Dual Link 8-bit DISM Standard	536
Figure 6-146 DISP_PWM Block Diagram.....	538
Figure 6-147 DISP_MUTEX Block Diagram	539
Figure 6-148 Refresh Mode Programming Sequence	540
Figure 6-149 Single Mode Programming Sequence.....	540
Figure 6-150 SMI Local Arbiter Block Diagram	542
Figure 6-151 SMI_COMMON and Neighboring Blocks	542
Figure 6-152 Simple System Architecture	544
Figure 6-153 MM_IOMMU Block Diagram	546
Figure 6-154 VDO Module Block Diagram	547
Figure 6-155 Internal Block Diagram of VDO	548
Figure 6-156 DISPFMT Block Diagram.....	555
Figure 6-157 NR Block Diagram	557
Figure 6-158 TVD Block Diagram	561
Figure 6-159 DISP_MONITOR Overview	563
Figure 6-160 DISP_MONITOR Function Blocks	564
Figure 6-161 IRQ Status	565

1 System Overview

MT2712 is a highly integrated and scalable application automotive processor with rich multimedia features. It integrates a dual-core ARM® Cortex-A72 MPCore™ operating up to 1.5 GHz, a quad-core ARM® Cortex-A35 MPCore™, a powerful ARM® Mali™ T880 GPU, and a well-proven multi-standard video codec. MT2712 incorporates a quad-channel 16-bit LPDDR4 DRAM interface for ultimate performance requirement. In addition, an extensive set of interfaces are integrated to interface to camera modules, external audio components, MMC/SD cards, external modules with connectivity of Bluetooth/WLAN/GNSS/LTE, and display panels.

The 1.5 GHz dual-core ARM® Cortex-A72 MPCore™ application processor offers computing power to support the latest open-source operating systems along with the In-Vehicle-Infotainment (IVI) and instrument cluster applications. The quad-core ARM® Cortex-A35 MPCore™ can provide sufficient computing power to support dedicated usages such as software-based audio DSP processing, software-based graphic rendering and software-defined radio for digital radio broadcasting, etc. In addition, the unique Memory Protection Unit (MPU) and Device Access Protection Control (DAPC) unit integrated in MT2712 provide a simple hardware mechanism which minimizes the virtualization overhead of the hypervisors when the multiple operating systems are coexisting for the usage of different application domains.

The multi-standard video codec in MT2712, supporting HEVC, H.264 and MPEG-4, etc., is also integrated to provide advanced multimedia processing and support multi-streaming audio and video usages. Rich automotive camera features, such as around view monitoring, backview monitoring, automotive driving recording, and driver monitoring, etc., can be supported via the maximum 6x video stream inputs with CVBS/MIPI-CSI2, or Ethernet AVB inputs. The maximum 3x display paths with Low-Voltage Differential Signaling (LVDS) or MIPI-DSI outputs can support up to three display panels. MT2712 also provides the display monitoring, camera monitoring and audio monitoring features which can be utilized for the realization of safety requirement.

1.1 General Description

MT2712 integrates a dual-core ARM® Cortex-A72 MPCore™ operating up to 1.5 GHz, a quad-core ARM® Cortex-A35 MPCore™, a powerful ARM® Mali™ T880 GPU, a well-proven multi-standard video codec, and an extensive set of interfaces.

World-leading technology

Based on MediaTek's world-leading System on a Chip (SoC) architecture with advanced 28 nm automotive process, MT2712 integrates digital and analog IPs into one chip which brings the lower BOM and design effort to cost-effectively develop applications with high reliability.

1.2 Target Applications

- Application processor for In-Vehicle-Infotainment (IVI) and instrument cluster systems
 - Powerful computing and graphics for embedded navigation, rendering of interactive graphic Human-Machine Interface (HMI) and speedometers.
 - Three independent displays may serve as main/sub central consoles and/or main/sub instrument clusters
 - Multi-channel audio input/output interfaces and powerful CPU computing may support software-based audio digital signal processing
- Informative driver assistance processor for camera-based Advanced Driver Assistance Systems (ADAS)
 - Around-View Monitoring (AVM) with Camera inputs via Ethernet Audio Video Bridging (AVB) or MIPI-CSI2 (with external multiplexing bridge chip)
 - Forward/Internal Camera Recorder
 - Driver Monitoring
- Industrial/business application processor

1.3 Key Features

1.3.1 CPU Clusters

- ARM® big.LITTLE heterogeneous computing of Cortex A72 and A35 cores
 - BCPU: Cortex-A72-r0p3-00rel0
 - LCPU: Cortex-A35-r0p1-00eac0
 - 2x High performance Cortex A72 operating up to 1.5 GHz
 - 4x Power efficient Cortex A35 operating up to 1.2 GHz
 - Scalable configurations from 7.5K to 22.6K DMIPS performance
 - 32 KB L1I-cache and 32 KB L1D-cache
 - Unified L2 cache: 1 MB for A72 cluster, 512 KB for A35 cluster
 - NEON multimedia processing engine and SIMDv2/VFPv4 ISA support
 - ARM® TrustZone® Security
 - Thermal throttle with Dynamic Frequency Scaling (DFS) power control
 - Per CPU core power control: off/on and clock rate scaling

1.3.2 Memory

- DRAM: LPDDR4
 - Support 2/4-channel mode by 16-bit up to 2800 Mb/s, with peak bandwidth at 22.4 GB/s
 - Maximum capacity: 1 GB per channel; 4 GB in total
 - Configurations supported:

- 32b (2-ch) – 512 MB, 1024 MB, 1536 MB, 2048 MB
- 64b (4-ch) – 1024 MB, 2048 MB, 3072 MB, 4096 MB
- Note: Each DDR channel needs to use equal capacity.
- Speed up to 2800 MHz
- Self-refresh/partial self-refresh mode support
- Flash storage
 - NOR: SPI interface (1/2/4b) up to 100 MHz, 256 MB capacity
 - NAND: 8b parallel (data bus shared with eMMC) for SLC with 12b ECC, or MLC with 60b ECC
 - 1x eMMC 5.0: 8b interface, up to 2 TB capacity
 - 3x SDIO3.0/SD3.0: 4b interface, up to 100 MB/s
- Fuse-On-chip embedded memories
 - SRAM with ECC for error detection/correction

1.3.3 GPU

- ARM® Mali-T880 MP4
 - Operating up to 500 MHz, with embedded 512 KB cache
 - Graphic processing performance up to
 - 250 M triangles/s (3D),
 - 2 G pixels/s (2D),
 - 102 Gflops/s (floating point)
 - Support:
 - OpenGL/ES 1.1/2.0/3.0/3.1/3.2
 - Vulkan 1.0
 - OpenVG 1.1,
 - OpenCL 1.1/1.2

1.3.4 Video/Audio Interfaces

- Video input/output interfaces:
 - 3x Display outputs: via MIPI-DSI or LVDS
 - Display1: up to 1920×1080@p60, 2560×1600@p60 (left/right or odd/even) or 2880×1080@p60 (odd/even split)
 - Display2: up to 1920×1080@p60
 - Display3: up to 1920×1080@p60
 - Color depth: RGB888
 - Touch panel and back-light control via I2C
 - Embedded LCD gamma correction
 - Portrait/landscape mode

- Supported size: 2880×1080, 2560×1440, 1920×1080, 1920×720, 1600×900, 1600×480, 1440×540, 1280×800, 1280×720, 1024×768, 1024×600, 800×600, 800×480
- Display processing accelerator for
 - 4-layer overlay engine for each display
 - Support scaling, flipping, rotation, color format conversion
 - Support Picture Quality (PQ), including color enhancement, adaptive contrast/sharpness
- 2x Digital camera inputs: Each via MIPI-CSI2, up to 1920×1080 60 fps, or 4x 1280×1080 30fps for around view camera inputs
- 1x Analog camera input: CVBS-in up to 480p, with decoder and 3D comb filter for noise reduction
- Support display monitoring function
 - 24-bit pixel data input
 - Cyclic Redundancy Check (CRC) calculation of a maximum of 12 independent regions
 - All the regions are with programmable location and boundary.
- Audio input/output interfaces
 - Digital audio via I2S with TDM support
 - Master/Slave 2-ch 3x I2S inputs
 - Master 2-ch 2x and 4-ch 1x I2S outputs
 - Master 2x TDM outputs
 - Master/Slave 1x TDM input (shares pin with I2S input)
 - Slave 3x I2S-IQ interface
 - Sampling rate: 8K to 384K, 8/16/24 bit
 - Master/Slave 1x PCM
 - Microphone ADC input: 1x 2-channel (L/R)
 - SNDR 74 db@full-swing, or 19 db@-60 db
 - Support audio safety function (I2S/TDM CRC Check)

1.3.5 Multimedia Processing

- Video codec
 - H.264 encoder up to 1920×1080 30 fps
 - Multi-format video decoder up to 1920×1080 60 fps
 - Flexible to support decoding of multiple streams, e.g. 4x 1280×1024 30 fps for AVM in H.264
 - Supporting:
 - MPEG-1/2/4,
 - H.264 (CBP, MP, HiP, MVC),
 - H.265(HEVC),
 - WMV7/8/9,
 - RealVideo 8/9/10,
 - VP8/VP9
- Image decoder

- MJPEG for up to 4x 1280×1024 30 fps for AVM
- Audio codec
 - Hardware supports for Asynchronous Sampling Rate Conversion
 - 5x 2-ch Asynchronous Sample Rate Converter (ASRC) for sampling rates from 8K to 384K
 - Up to 4 conversion ratio setting from 1/16 to 8x
 - 16 b/32 b inputs, 16 b/32 b outputs
 - THD+N: about 150 db

1.3.6 Peripherals and Control

- High speed data interfaces
 - 4x USB
 - 2x USB3.1 Gen1 host mode for up to 8 end-point devices
 - 2x USB2.0 OTG, supporting Battery Charge (BC 1.2), up to 8 end-point devices
 - TypeC and PD (power delivery) support via external power components
 - 2x PClexpress (pin-sharing with USB3)
 - 1x Gen2 1-lane, RC/EP mode (root-complex or end-point)
 - 1x Gen2 1-lane, RC mode (root-complex only)
 - 1x Gigabit Ethernet
 - MII/RMII/RGMII and MDC/MDIO to external PHY
 - Ethernet AVB (IEEE 802.1AS) support
- Peripheral interfaces
 - 6x SPI: 1.8 V/3.3 V, up to 52 Mb/s
 - 6x UART: 1.8 V/3.3 V, up to 3 Mb/s
 - 6x I2C: 1.8 V/3.3 V, up to 3.4 MHz
 - Keypad input: 7×7 matrix (49 keys)
 - 10x GPIO
 - 8x PWM: 1.8 V/3.3 V, 16-bit, frequency 33 MHz~0.5 Hz
 - AUX ADC: 11-bit resolution, 0~1.5 V, 1.6 MHz clock, 100 ksps
- System control
 - Reset/ICE/JTAG/Wake_up
 - 4x external interrupt
 - Real Time Clock (RTC)

1.4 Key Parameters

1.4.1 Process and Package

- Fab node: CMOS 28 nm HPC

- Package: HFC FBGA 23×23 mm, 0.65 mm ball-pitch
 - 852 Pin
- PCB: 4/4 mil trace/spacing, 10/18 via size, 6-layer PTH

1.4.2 Quality

- Reliability: AEC-Q100 grade 3 (-40°C~+85°C)
- Zero defect target: <10 dppm for 1st year production
- Environment: ISO-14001

1.5 Supporting

1.5.1 OS and BSP Support

- AGL/Yocto Linux
- Android 8.0 and later version

1.5.2 Companion Components

- Connectivity
 - Connectivity combo chip: MT6630 Bluetooth + WLAN + GNSS
 - Support a single 2.4 GHz antenna for BT & WLAN, or a single tri-band antenna for WLAN (2.4 GHz, 5 GHz), Bluetooth, and GNSS, with intelligent coexistence scheme
 - WLAN: 802.11 a/b/g/n/ac
 - Bluetooth: v2.1+EDR, 3.0, 4.1
 - GNSS: GPS, Glonass, Beidou, Galileo, QZSS, tri-band concurrent reception, with SBAS support for WAAS/MSAS/EGNOS/GAGAN
 - Interface: SDIO3.0. Support SDR104 (Bus clock rate 208 MHz).
 - Stand-alone connectivity: BT, GNSS
- Power supply solution: discrete component solution

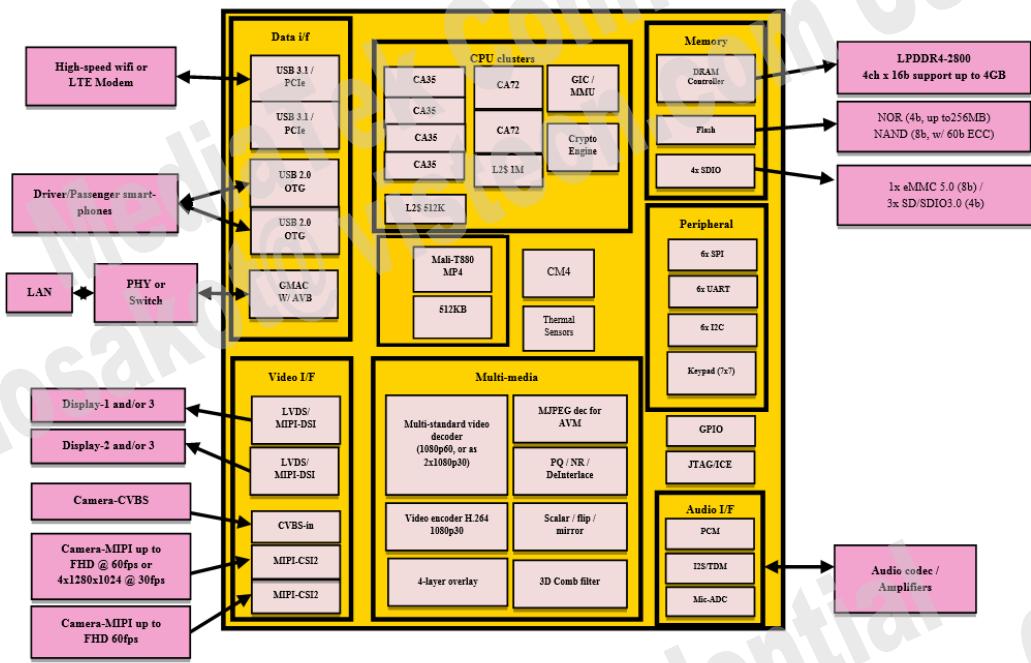


Figure 1-1 MT2712 Block Diagram

2 Product Description

2.1 Pin Description

2.1.1 Ball Map View

852	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
A	ARDQ11	ARDQ11	ARDQ11	ARDQ9	ARDQ9	ARDQ5	ARDQ5	ARDQ1	ARDQ1	ARDQ9	ARDQ9	ARDQ12	ARDQ12	ARDQ1	ARDQ9	ARDQ9	ARDQ12	ARDQ11	VCC31	VCC181	DS10_T	DPO	DS10_T	A											
B	ARDQ11	ARDQ11	ARDQ11	ARDQ8	ARDQ8	ARDQ15	ARDQ15	ARDQ13	ARDQ13	ARDQ6	ARDQ6	ARDQ2	ARDQ2	ARDQ1	ARDQ8	ARDQ8	ARDQ13	ARDQ13	ARDQ5	ARDQ5	ARDQ11	ARDQ11	ARDQ5	ARDQ5	ARDQ11	B									
C	ARDQ11	ARDQ11	ARDQ11	ARDQ11	ARDQ11	ARDQ10	ARDQ10	ARDQ14	ARDQ14	ARDQ4	ARDQ4	ARDQ7	ARDQ7	ARDQ3	ARDQ3	ARDQ2	ARDQ2	ARDQ7	ARDQ4	ARDQ4	ARDQ11	ARDQ11	ARDQ9	ARDQ9	ARDQ11	C									
D	ARDQ11	ARDQ10	ARDQ11	D																															
E	ARDQ9	ARDQ8	E																																
F	ARDQ11	F																																	
G	ARDQ13	ARDQ13	ARDQ12	G																															
H	ARDQ9	ARDQ9	ARDQ4	H																															
J	ARDQ5	ARDQ6	J																																
K	ARDQ7	ARDQ8	K																																
L	ARDQ9	L																																	
M	ARDQ1	ARDQ2	ARDQ3	M																															
N	ARD1	N																																	
P	ARDQ3	ARDQ3	ARDQ2	P																															
R	ARDQ1	R																																	
T	ARDQ6	ARDQ7	ARDQ7	ARDQ8	T																														
U	ARDQ5	ARDQ4	U																																
V	ARDQ9	ARDQ9	ARDQ9	ARDQ12	V																														
W	ARDQ13	ARDQ14	W																																
Y	ARDQ15	Y																																	
AA	ARDQ1	AA																																	
AB	ARDQ10	ARDQ10	ARDQ11	AB																															
AC	DVSS	AC																																	
AD	AVD103	AD																																	
AB	AVD103	AB																																	
AP	WC331	DATA1	AP																																
AG	TD801	AG																																	
AH	I2S01	I2S02	AH																																
AJ	I2S02	AJ																																	
AK	I2S11	AK																																	
AL	I2S11	AL																																	
AM	I2S12	AM																																	
AN	MCLK	AN																																	
AP	VCC331	AP																																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	

Figure 2-1 Ball Map View for LPDDR4

2.1.2 Pin Coordinates

Table 2-1 Pin Coordinate Using LPDDR4

Ball Loc.	Ball Name	Ball Loc.	Ball Name	Ball Loc.	Ball Name
A1	DUMMY	M20	VCCK	AC11	DVSS
A2	DUMMY	M21	VCCK	AC12	DVSS
A3	CRDQ11	M22	VCCK	AC13	DVSS
A4	CRDQ9	M23	DVSS	AC14	DVSS
A5	CRDQS1	M24	DVSS	AC15	DVSS
A8	CRDQS0#	M25	AVDD10_SSUSB_P3	AC16	DVSS
A9	CRDQ5	M29	DVSS	AC17	DVSS
A11	CRDQ1	M30	AVSS18_MIPITX_CH2	AC19	DVSS
A13	CRA1	M31	AVSS18_MIPITX_CH2	AC20	DVSS
A15	DRDQ0	M32	DSI3_TDPO	AC21	DVSS
A17	DRDQ6	N2	ARA1	AC22	DVSS
A19	DRDQ12	N3	NC	AC23	DVSS
A22	DRDQ11	N11	DVSS	AC24	DVSS
A25	VCC33IO_GPIO	N12	DDRV_BRDDR	AC26	NREB
A27	VCC18IO	N13	DVSS	AC27	NCEB1
A29	DSI0_TDPO	N14	DVSS	AC28	NF_DQS
A32	DSI0_TDN3	N15	DVSS	AC29	DVSS
A33	DUMMY	N16	DVSS	AC30	NWEB
A34	DUMMY	N17	DVSS	AC31	NCEB0
B1	DUMMY	N18	DVSS	AC32	PCM_SYNC
B2	BRA5	N19	DVSS	AC33	PCM_TX
B3	REXTDN	N20	DVSS	AC34	PCM_RX
B4	CRDQ8	N21	DVSS	AD1	AVDD33_USB_P1
B5	CRDQS1#	N22	DVSS	AD2	AVDD33_USB_P0
B6	CRDQ15	N23	DVSS	AD3	AVDD18_USB_P0
B7	CRDQ13	N24	DVSS	AD4	USB_DM_P0
B8	CRDQS0	N25	DVSS	AD5	USB_DP_P0
B9	CRDQ6	N26	DSI2_TCP	AD6	DVSS
B10	CRDQMO	N27	DSI2_TCN	AD7	USB_DM_P1
B11	CRDQ2	N28	AVSS18_MIPITX_CH2	AD8	USB_DP_P1
B12	NC	N29	DSI2_TDPO	AD9	USB_VBUS_P1
B13	NC	N30	DSI2_TDN0	AD11	DVSS
B14	DRDQ2	N31	AVSS18_MIPITX_CH2	AD13	VCCK_BCPU
B15	DRDQS1	N32	DSI3_TDN0	AD14	VCCK_BCPU
B16	DRDQMO	N33	DSI3_TDPO	AD15	VCCK_BCPU
B17	DRDQ5	N34	DSI3_TDN1	AD16	VCCK_BCPU
B18	DRDQS0	P1	NC	AD17	VCCK_BCPU
B19	DRDQ13	P2	ARDQ3	AD18	VCCK

Ball Loc.	Ball Name	Ball Loc.	Ball Name	Ball Loc.	Ball Name
B20	DRDQ15	P3	ARDQ2	AD19	VCCK_LCPU
B22	DRA4	P4	NC	AD20	VCCK_LCPU
B23	DVSS	P5	NC	AD21	VCCK_LCPU
B25	EINT1	P6	NC	AD22	DVSS
B26	KPCOL1	P7	DVSS	AD25	VCC18IO
B27	KPCOL2	P8	NC	AD32	MSDCOE_DAT7
B29	DSI0_TDNO	P9	NC	AD33	DVSS
B30	DSI0_TDNI	P10	BRA0	AE3	AVDD18_USB_P1
B31	DSI0_TDP2	P11	DVSS	AE4	TDM00_DATA
B32	DSI0_TDP3	P12	DDRV_BRDDR	AE5	TDM00_LRCK
B33	VRT0	P13	VCCK	AE6	TDM00_BCK
B34	DUMMY	P14	VCCK	AE7	TDM00_MCLK
C1	BRA4	P15	VCCK	AE8	DVSS
C2	BRRESET#	P16	VCCK	AE9	DVSS
C3	DVSS	P17	VCCK	AE10	DVSS
C4	CRDQ10	P18	VCCK	AE11	KPROW6
C5	CRDQM1	P19	VCCK	AE12	KPROW5
C6	CRDQ14	P20	VCCK	AE14	URXD1
C7	CRDQ12	P21	VCCK	AE15	URXDO
C8	CRDQ4	P22	VCCK	AE16	RESET_
C9	CRDQ7	P23	DVSS	AE20	AVDD33_AUADC
C10	CRDQ0	P24	DVSS	AE22	AVSS33_AUADC
C11	CRDQ3	P32	AVSS18_MIPITX_CH2	AE24	URTS5
C12	CRA0	P33	DSI3_TCN	AE25	UCTS5
C13	DRDQ3	P34	DSI3_TCP	AE27	DVSS
C14	DRDQ1	R2	ARDQ1	AE28	NCLE
C15	DRDQS1#	R3	ARDQ0	AE29	NALE
C16	DRDQ7	R4	DVSS	AE30	MSDCOE_RSTB
C17	DRDQ4	R5	DVSS	AE31	MSDCOE_CLK
C18	DRDQS0#	R6	DVSS	AE32	MSDCOE_DAT5
C19	DRDQ14	R7	NC	AE33	MSDCOE_DAT6
C20	DRDQM1	R8	NC	AE34	MSDCOE_DAT2
C21	DRDQ9	R11	DVSS	AF1	VCC33IO_TDMO
C22	DRA5	R12	DDRV_BRDDR	AF2	TDM01_DATA
C23	CLKREQN_P0	R13	DVSS	AF3	TDM01_BCK
C24	WAKEEN_P0	R14	DVSS	AF8	DVSS
C25	KPROW1	R15	DVSS	AF10	GPIO148
C26	KPROW2	R16	DVSS	AF12	KPROW4
C27	AVSS18_MIPITX_CH1	R17	DVSS	AF14	UTXD1
C28	DSI0_TCP	R18	DVSS	AF16	I2C_SCL0
C29	DSI0_TCN	R19	DVSS	AF19	AU_VIN1_P

Ball Loc.	Ball Name	Ball Loc.	Ball Name	Ball Loc.	Ball Name
C30	DSI0_TDP1	R20	DVSS	AF20	AU_VINO_P
C31	DSI0_TDNN2	R21	DVSS	AF22	AU_TP
C32	VRT1	R22	DVSS	AF24	URXD5
C33	AVDD18_MIPITX_CH1	R23	DVSS	AF32	MSDCOE_DAT4
C34	AVDD18_USB_P2	R24	DVSS	AF33	MSDCOE_DAT1
D1	BRDQ11	R25	VRT3	AG2	TDMO1_LRCK
D2	BRDQ10	R26	AVDD18_MIPITX_CH2	AG3	TDMO1_MCLK
D3	BRA3	R27	DSI2_TDN2	AG4	I2SO0_BCK
D5	NC	R28	DSI2_TDP2	AG5	I2SO0_MCLK
D6	CRA3	R29	AVSS18_MIPITX_CH2	AG7	AUD_EXT_CK1
D8	DVSS	R30	DSI2_TDN1	AG8	SPI1_SO
D9	NC	R31	DSI2_TDP1	AG10	GPIO149
D11	NC	R32	DSI3_TDN2	AG11	KPCOL6
D12	DVSS	R33	DSI3_TDP2	AG12	KPCOL5
D13	DRA1	T1	ARDQMO	AG14	KPCOL4
D15	NC	T2	ARDQ7	AG15	UTXD0
D16	DVSS	T3	ARDQ6	AG16	I2C_SCL1
D17	NC	T4	NC	AG18	AU_MICBIAS0
D19	NC	T5	ARA0	AG19	AU_VIN1_N
D20	DRDQ8	T6	ARCKE1	AG20	AU_VINO_N
D21	DRDQ10	T7	DVSS	AG22	AU_TN
D23	CLKREQN_P1	T8	ARCKEO	AG24	UTXD5
D24	PERSTB_PO	T9	ARCS1	AG26	GBE_RXD3
D25	PWM5	T10	DVSS	AG27	MSDC3_CLK
D27	PWM6	T11	DVSS	AG28	DVSS
D28	DSI1_TDPO	T12	DDRV_BRDDR	AG29	MSDCOE_DSL
D30	DSI1_TDP2	T13	VCK	AG30	MSDCOE_CMD
D33	USB_DM_P2	T14	VCK	AG31	DVSS
D34	USB_DP_P2	T15	VCK	AG32	DVSS
E1	BRDQ9	T16	VCK	AG33	MSDCOE_DAT0
E2	BRDQ8	T17	VCK	AG34	MSDCOE_DAT3
E3	NC	T18	VCK	AH1	VCC33IO_I2SO02
E5	NC	T19	VCK	AH2	I2SO2_MCLK
E6	CRA2	T20	VCK	AH3	I2SO2_BCK
E8	NC	T21	VCK	AH7	AUD_EXT_CK2
E9	CRCLK#	T22	VCK	AH8	SPI1_SI
E11	DVSS	T23	VCK	AH10	GPIO150
E12	DVSS	T24	AVSS18_MIPITX_CH2	AH12	KPROW3
E13	NC	T25	VRT2	AH14	URXD2
E15	DVSS	T28	AVSS18_MIPITX_CH2	AH16	I2C_SCL2
E17	DRCLK#	T29	AVSS18_MIPITX_CH2	AH20	AVSS33_AUADC

Ball Loc.	Ball Name	Ball Loc.	Ball Name	Ball Loc.	Ball Name
E19	NC	T30	AVSS18_MIPITX_CH2	AH22	AVDD22_AUADC
E20	DVSS	T31	AVSS18_MIPITX_CH2	AH23	DVSS
E21	DRRESET#	T32	AVSS18_MIPIRX	AH24	URTS4
E23	WAKEEN_P1	T33	DSI3_TDN3	AH26	GBE_RXD2
E24	EINT0	T34	DSI3_TDP3	AH27	GBE_RXC
E25	PWM4	U2	ARDQ5	AH29	MSDC3_INS
E27	PWM7	U3	ARDQ4	AH30	MSDC3_CMD
E28	DSI1_TDN0	U11	DVSS	AH31	MSDC3_DSL
E30	DSI1_TDN2	U12	DDRV_ARDDR	AH32	MSDC3_DAT2
E32	DVSS	U13	DVSS	AH33	MSDC3_DAT3
E33	AVDD33_USB_P2	U14	DVSS	AJ2	I2SO2_LRCK
F1	BRDQM1	U15	DVSS	AJ3	I2SO2_DATA0
F2	BRDQ15	U16	DVSS	AJ4	I2SO0_LRCK
F3	BRDQ14	U17	DVSS	AJ5	I2SO0_DATA0
F4	DVSS	U18	DVSS	AJ6	I2SO0_DATA1
F5	BRCKE0	U19	DVSS	AJ7	VCC18IO_EFUSE
F6	CRA5	U20	DVSS	AJ8	SPI5_CK
F7	DVSS	U21	DVSS	AJ10	GPIO151
F8	CRA4	U22	DVSS	AJ11	JTCK_ICE
F9	CRCLK	U23	DVSS	AJ12	KPCOL3
F11	NC	U24	AVSS18_MIPITX_CH2	AJ14	UTXD2
F13	DVSS	U26	DSI2_TDP3	AJ16	I2C_SCL3
F15	NC	U27	DSI2_TDN3	AJ18	DRV_VBUS_P2
F17	DRCLK	U28	AVSS18_MIPIRX	AJ19	DVSS
F19	DVSS	U30	CSI0_RCP	AJ20	AUXIN1
F20	NC	U31	CSI0_RCN	AJ22	MSDC2_INS
F24	KPROW0	U32	CSI1_RDNO	AJ23	DVSS
F25	PWM3	U33	CSI1_RDP0	AJ24	UCTS4
F27	AVSS18_MIPITX_CH1	V1	ARDQSO#	AJ26	GBE_RXD1
F28	DSI1_TDP1	V2	ARDQSO	AJ27	GBE_TXC
F30	DSI1_TDP3	V3	ARDQ12	AJ32	NOR_IO3
F31	DSI1_TDN3	V4	DVSS	AJ33	MSDC3_DAT1
F32	DVSS	V5	NC	AJ34	MSDC3_DAT0
F33	SSUSB_RXP_P2	V6	NC	AK1	VCC33IO_I2SIO1
F34	SSUSB_RXN_P2	V7	ARCLK#	AK2	I2SO1_DATA0
G2	BRDQ13	V8	ARCLK	AK3	I2SI1_MCLK
G3	BRDQ12	V9	ARC50	AK4	I2SO1_BCK
G9	CRCS0	V10	NC	AK5	I2SO1_MCLK
G11	NC	V11	DVSS	AK6	SPI1_CK
G12	NC	V12	DDRV_ARDDR	AK8	SPI5_MO
G13	NC	V13	VCCK	AK10	GPIO152

Ball Loc.	Ball Name	Ball Loc.	Ball Name	Ball Loc.	Ball Name
G15	DRA0	V14	VCCK	AK11	SPI2_MI
G16	DVSS	V15	VCCK	AK12	SPI2_CK
G17	DRCS0	V16	VCCK	AK14	SPI2_CSN
G19	DRA2	V17	VCCK	AK15	I2C_SCL4
G21	NC	V18	VCCK	AK18	DRV_VBUS_P3
G23	PERSTB_P1	V19	VCCK	AK19	AUXIN2
G24	DRV_VBUS_PO	V20	VCCK	AK20	AUXINO
G25	PWM2	V21	VCCK	AK22	MSDC2_CMD
G28	DSI1_TDNN1	V22	VCCK	AK23	DVSS
G32	DVSS	V23	VCCK	AK24	UTXD4
G33	DVSS	V24	AVSS18_MIPITX_CH2	AK26	GBE_RXD0
G34	DVSS	V32	AVSS18_MIPIRX	AK27	GBE_TXER
H1	BRDQS0#	V33	CSI1_RDN1	AK29	IDDIG_P1
H2	BRDQS0	V34	CSI1_RDP1	AK30	MSDC1_DAT0
H3	BRDQ4	W2	ARDQ13	AK32	MSDC1_PSW
H4	BRA2	W3	ARDQ14	AK33	NOR_CS
H5	DVSS	W4	DVSS	AK34	NOR_IO1
H6	NC	W5	DVSS	AL2	I2S11_LRCK
H7	BRA1	W7	DVSS	AL3	I2S11_BCK
H8	BRCKE1	W9	DVSS	AL6	SPI1_CSN
H9	NC	W11	DVSS	AL8	SPI5_CSN
H11	CRCKE1	W12	DDRV_ARDDR	AL10	SPI5_MI
H13	CRCKE0	W13	VCCK	AL12	SPI3_CK
H15	NC	W15	VCCK	AL14	SPI2_MO
H16	DVSS	W16	DVSS	AL16	I2C_SCL5
H17	NC	W17	DVSS	AL18	DVSS
H19	DRCKE1	W19	DVSS	AL19	DVSS
H21	DVSS	W20	DVSS	AL20	DVSS
H23	EINT2	W21	DVSS	AL22	MSDC2_DAT1
H25	PWM1	W22	DVSS	AL23	DVSS
H27	DSI1_TCN	W24	AVSS18_MIPIRX	AL24	URXD4
H28	DSI1_TCP	W25	AVDD18_MIPIRX	AL26	VCC33IO_UART4B
H30	PCIe_CLKN_P2	W26	CSI0_RDN1	AL27	GBE_TXEN
H31	PCIe_CLKP_P2	W27	CSI0_RDP1	AL29	DRV_VBUS_P1
H32	DVSS	W28	AVSS18_MIPIRX	AL30	MSDC1_DAT1
H33	SSUSB_TXN_P2	W29	CSI0_RDNO	AL32	MSDC1_DAT3
H34	SSUSB_TXP_P2	W30	CSI0_RDP0	AL33	NOR_IO2
J2	BRDQ5	W31	AVSS18_MIPIRX	AL34	VCC33IO_NOR
J3	BRDQ6	W32	CSI1_RCN	AM1	I2S12_LRCK
J10	NC	W33	CSI1_RCP	AM2	I2S12_BCK
J11	CRCS1	Y2	ARDQ15	AM3	I2S12_DATA

Ball Loc.	Ball Name	Ball Loc.	Ball Name	Ball Loc.	Ball Name
J13	DVSS	Y3	ARDQM1	AM4	I2SI1_DATA
J15	NC	Y4	ARA5	AM5	I2SO1_LRCK
J17	DVSS	Y5	NC	AM6	SPI4_CK
J19	NC	Y6	DVSS	AM7	SPI4_CSN
J21	DRCKE0	Y7	NC	AM8	SPI0_CSN
J23	EINT3	Y8	ARA4	AM9	JTDI_ICE
J25	PWM0	Y9	ARA3	AM10	JTMS_ICE
J27	AVSS18_MIPITX_CH1	Y10	NC	AM11	JTDO_ICE
J32	SSUSB_VRT_P2	Y11	DVSS	AM12	SPI3_CSN
J33	DVSS	Y12	DDRV_ARDDR	AM13	UTXD3
J34	AVDD10_SSUSB_P2	Y13	VCCK_BCPU	AM14	I2C_SDA1
K2	BRDQ7	Y14	VCCK_BCPU	AM15	I2C_SDA5
K3	BRDQM0	Y15	VCCK_BCPU	AM16	I2C_SDA4
K4	NC	Y16	VCCK_BCPU	AM17	AVSS18_CVBS
K5	BRCLK#	Y17	VCCK_BCPU	AM19	AVDD18_PLLGP
K6	BRCLK	Y18	VCCK	AM20	XTALO
K7	BRCS0	Y19	VCCK_LCPU	AM21	AVSS18_AP
K8	NC	Y20	VCCK_LCPU	AM22	MSDC2_PSW
K9	NC	Y21	VCCK_LCPU	AM23	MSDC2_DAT2
K11	AVDD18_RDDR	Y22	VCCK	AM24	MSDC2_DAT3
K12	NC	Y23	VCCK	AM25	MSDC2_CLK
K13	NC	Y24	AVSS18_MIPIRX	AM26	GBE_MDIO
K15	NC	Y25	AVSS18_MIPIRX	AM27	GBE_RXDV
K16	DVSS	Y26	AVSS18_MIPIRX	AM28	GBE_TXD1
K17	NC	Y27	AVSS18_MIPIRX	AM29	GBE_TXD3
K19	DRA3	Y28	AVSS18_MIPIRX	AM30	RTC_EINT
K20	DVSS	Y29	AVSS18_MIPIRX	AM31	MSDC1_DAT2
K21	DRCS1	Y30	AVSS18_MIPIRX	AM32	MSDC1_CLK
K23	IDDIG_P0	Y32	AVSS18_MIPIRX	AM33	NOR_CK
K24	KPCOL0	Y33	CSI1_RDN2	AM34	NOR_IO0
K27	SSUSB_RXN_P3	Y34	CSI1_RDP2	AN1	DUMMY
K28	SSUSB_RXP_P3	AA1	ARDQS1	AN2	I2S12_MCLK
K30	SSUSB_TXP_P3	AA2	ARDQS1#	AN3	I2S10_MCLK
K31	SSUSB_TXN_P3	AA3	ARDQ8	AN4	I2S10_BCK
K32	DVSS	AA11	DVSS	AN5	I2S10_DATA
K33	AVDD33_USB_P3	AA12	DDRV_ARDDR	AN6	SPI4_MI
L1	BRDQS1	AA13	DVSS	AN7	SPI0_MO
L2	BRDQS1#	AA14	VCCK_SRAM_BCPU	AN8	SPI0_CK
L3	BRDQ0	AA15	VCCK_SRAM_BCPU	AN9	SPI0_MI
L4	DVSS	AA16	VCCK_SRAM_BCPU	AN10	JTRSTB_ICE
L5	DVSS	AA17	DVSS	AN12	SPI3_MO

Ball Loc.	Ball Name	Ball Loc.	Ball Name	Ball Loc.	Ball Name
L6	NC	AA19	VCCK_SRAM_LCPU	AN13	URXD3
L7	BRCS1	AA20	VCCK_SRAM_LCPU	AN14	URTS3
L10	RTP	AA21	VCCK_SRAM_LCPU	AN15	I2C_SDA0
L11	RTN	AA22	DVSS	AN16	I2C_SDA3
L12	DDRV_CRDDR	AA23	DVSS	AN17	AVDD18_CVBS
L13	DDRV_CRDDR	AA27	CSI0_RDP2	AN18	CVBS_COM
L14	DDRV_CRDDR	AA28	CSI0_RDN2	AN20	XTAL1
L15	DDRV_CRDDR	AA29	CSI0_RDN3	AN21	DVSS
L16	DDRV_CRDDR	AA30	CSI0_RDP3	AN22	AVSS_REFN
L17	DDRV_DRDDR	AA31	AVSS18_MIPIRX	AN23	MSDC2_DAT0
L18	DDRV_DRDDR	AA32	CSI1_RDN3	AN25	GBE_INTR
L19	DDRV_DRDDR	AA33	CSI1_RDP3	AN26	GBE_COL
L20	DDRV_DRDDR	AB1	ARDQ9	AN27	GBE_RXER
L21	DDRV_DRDDR	AB2	ARDQ10	AN28	GBE_TXD0
L22	VCCK	AB3	ARDQ11	AN29	GBE_TXD2
L23	DVSS	AB4	ARA2	AN30	VCC33IO_GBE
L25	SSUSB_VRT_P3	AB5	DVSS	AN31	TESTMODE
L26	AVDD18_USB_P3	AB6	NC	AN32	MSDC1_CMD
L27	AVDD18_SSUSB_P3	AB7	NC	AN33	MSDC1_INS
L28	AVDD18_SSUSB_P2	AB8	NC	AN34	DUMMY
L29	PCIe_CLKP_P3	AB9	DVSS	AP1	DUMMY
L30	PCIe_CLKN_P3	AB10	NC	AP2	DUMMY
L32	DVSS	AB12	VCCK	AP3	VCC33IO_I2SI02
L33	USB_DP_P3	AB13	DVSS	AP4	I2SI0_LRCK
L34	USB_DM_P3	AB14	VCCK_BCPU	AP5	VCC33IO_SPIMCU
M1	BRDQ1	AB15	VCCK_BCPU	AP6	SPI4_MO
M2	BRDQ2	AB16	VCCK_BCPU	AP8	VCC18IO
M3	BRDQ3	AB17	VCCK_BCPU	AP9	VCC33IO_SPICTP
M4	DVSS	AB18	VCCK	AP12	SPI3_MI
M5	NC	AB19	VCCK_LCPU	AP14	UCTS3
M6	NC	AB20	VCCK_LCPU	AP16	I2C_SDA2
M7	NC	AB21	VCCK_LCPU	AP18	CVBS0P
M8	DVSS	AB22	VCCK	AP21	AVDD18_AP
M12	DDRV_BRDDR	AB23	VCCK	AP22	REFP
M13	VCCK	AB24	DVSS	AP25	VCC33IO_MSDC2
M14	VCCK	AB25	CM2MCLK	AP27	GBE_MDC
M15	VCCK	AB26	CMMCLK	AP30	RTC_XI
M16	VCCK	AB31	PCM_CLK	AP31	VCC18IO_RTC
M17	VCCK	AB33	AVSS18_MIPIRX	AP32	VCC33IO_MSDC1
M18	VCCK	AC2	DVSS	AP33	DUMMY
M19	VCCK	AC9	USB_VBUS_P0	AP34	DUMMY

2.1.3 Detailed Pin Description

Table 2-2 Acronym for Pin Type

Abbreviation	Description
AI	Analog input
AO	Analog output
AIO	Analog bi-direction
DI	Digital input
DO	Digital output
DIO	Digital bi-direction
P	Power
G	Ground

Table 2-3 Detailed Pin List

Pin Name	Type	Power Domain
RESET_	DIO	VCC18IO
TESTMODE	DIO	VCC18IO_RTC
RTC_EINT	DIO	VCC18IO_RTC
RTC_XI	DIO	VCC18IO_RTC
JTCK_ICE	DIO	VCC18IO
JTDI_ICE	DIO	VCC18IO
JTDO_ICE	DIO	VCC18IO
JTMS_ICE	DIO	VCC18IO
JTRSTB_ICE	DIO	VCC18IO
GPIO148	DIO	VCC18IO
GPIO149	DIO	VCC18IO
GPIO150	DIO	VCC18IO
GPIO151	DIO	VCC18IO
GPIO152	DIO	VCC18IO
I2S10_BCK	DIO	VCC33IO_I2SI02
I2S10_DATA	DIO	VCC33IO_I2SI02
I2S10_LRCK	DIO	VCC33IO_I2SI02
I2S10_MCLK	DIO	VCC33IO_I2SI02
I2S11_BCK	DIO	VCC33IO_I2SI01
I2S11_DATA	DIO	VCC33IO_I2SI01
I2S11_LRCK	DIO	VCC33IO_I2SI01
I2S11_MCLK	DIO	VCC33IO_I2SI01
I2S12_BCK	DIO	VCC33IO_I2SI02
I2S12_DATA	DIO	VCC33IO_I2SI02
I2S12_LRCK	DIO	VCC33IO_I2SI02
I2S12_MCLK	DIO	VCC33IO_I2SI02

Pin Name	Type	Power Domain
I2S00_BCK	DIO	VCC33IO_I2S002
I2S00_DATA0	DIO	VCC33IO_I2S002
I2S00_DATA1	DIO	VCC33IO_I2S002
I2S00_LRCK	DIO	VCC33IO_I2S002
I2S00_MCLK	DIO	VCC33IO_I2S002
I2S01_BCK	DIO	VCC33IO_I2SIO1
I2S01_DATA0	DIO	VCC33IO_I2SIO1
I2S01_LRCK	DIO	VCC33IO_I2SIO1
I2S01_MCLK	DIO	VCC33IO_I2SIO1
I2S02_BCK	DIO	VCC33IO_I2S002
I2S02_DATA0	DIO	VCC33IO_I2S002
I2S02_LRCK	DIO	VCC33IO_I2S002
I2S02_MCLK	DIO	VCC33IO_I2S002
AUD_EXT_CK1	DIO	VCC33IO_I2S002
AUD_EXT_CK2	DIO	VCC33IO_I2S002
TDM00_BCK	DIO	VCC33IO_TDM0
TDM00_DATA	DIO	VCC33IO_TDM0
TDM00_LRCK	DIO	VCC33IO_TDM0
TDM00_MCLK	DIO	VCC33IO_TDM0
TDM01_BCK	DIO	VCC33IO_TDM0
TDM01_DATA	DIO	VCC33IO_TDM0
TDM01_LRCK	DIO	VCC33IO_TDM0
TDM01_MCLK	DIO	VCC33IO_TDM0
PCM_CLK	DIO	VCC18IO
PCM_RX	DIO	VCC18IO
PCM_SYNC	DIO	VCC18IO
PCM_TX	DIO	VCC18IO
URXD0	DIO	VCC18IO
UTXD0	DIO	VCC18IO
URXD1	DIO	VCC18IO
UTXD1	DIO	VCC18IO
URXD2	DIO	VCC18IO
UTXD2	DIO	VCC18IO
URXD3	DIO	VCC18IO
UTXD3	DIO	VCC18IO
UCTS3	DIO	VCC18IO
URTS3	DIO	VCC18IO
URXD4	DIO	VCC33IO_UART4B
UTXD4	DIO	VCC33IO_UART4B
UCTS4	DIO	VCC33IO_UART4B
URTS4	DIO	VCC33IO_UART4B

Pin Name	Type	Power Domain
URXD5	DIO	VCC33IO_UART4B
UTXD5	DIO	VCC33IO_UART4B
UCTS5	DIO	VCC33IO_UART4B
URTS5	DIO	VCC33IO_UART4B
KPCOL0	DIO	VCC18IO
KPCOL1	DIO	VCC18IO
KPCOL2	DIO	VCC18IO
KPCOL3	DIO	VCC18IO
KPCOL4	DIO	VCC18IO
KPCOL5	DIO	VCC18IO
KPCOL6	DIO	VCC18IO
KPROW0	DIO	VCC18IO
KPROW1	DIO	VCC18IO
KPROW2	DIO	VCC18IO
KPROW3	DIO	VCC18IO
KPROW4	DIO	VCC18IO
KPROW5	DIO	VCC18IO
KPROW6	DIO	VCC18IO
PWM0	DIO	VCC33IO_GPIO
PWM1	DIO	VCC33IO_GPIO
PWM2	DIO	VCC33IO_GPIO
PWM3	DIO	VCC33IO_GPIO
PWM4	DIO	VCC33IO_GPIO
PWM5	DIO	VCC33IO_GPIO
PWM6	DIO	VCC18IO
PWM7	DIO	VCC18IO
EINT0	DIO	VCC33IO_GPIO
EINT1	DIO	VCC33IO_GPIO
EINT2	DIO	VCC33IO_GPIO
EINT3	DIO	VCC33IO_GPIO
CLKREQN_P0	DIO	VCC33IO_GPIO
PERSTB_P0	DIO	VCC33IO_GPIO
WAKEEN_P0	DIO	VCC33IO_GPIO
CLKREQN_P1	DIO	VCC33IO_GPIO
PERSTB_P1	DIO	VCC33IO_GPIO
WAKEEN_P1	DIO	VCC33IO_GPIO
SPI0_CK	DIO	VCC33IO_SPICTP
SPI0_CSN	DIO	VCC33IO_SPICTP
SPI0_MI	DIO	VCC33IO_SPICTP
SPI0_MO	DIO	VCC33IO_SPICTP
SPI1_CK	DIO	VCC33IO_SPIMCU

Pin Name	Type	Power Domain
SPI1_CSN	DIO	VCC33IO_SPIMCU
SPI1_SI	DIO	VCC33IO_SPIMCU
SPI1_SO	DIO	VCC33IO_SPIMCU
SPI2_CK	DIO	VCC18IO
SPI2_CSN	DIO	VCC18IO
SPI2_MI	DIO	VCC18IO
SPI2_MO	DIO	VCC18IO
SPI3_CK	DIO	VCC18IO
SPI3_CSN	DIO	VCC18IO
SPI3_MI	DIO	VCC18IO
SPI3_MO	DIO	VCC18IO
SPI4_CK	DIO	VCC33IO_SPIMCU
SPI4_CSN	DIO	VCC33IO_SPIMCU
SPI4_MI	DIO	VCC33IO_SPIMCU
SPI4_MO	DIO	VCC33IO_SPIMCU
SPI5_CK	DIO	VCC33IO_SPICTP
SPI5_CSN	DIO	VCC33IO_SPICTP
SPI5_MI	DIO	VCC33IO_SPICTP
SPI5_MO	DIO	VCC33IO_SPICTP
I2C_SCL0	DIO	VCC18IO
I2C_SDA0	DIO	VCC18IO
I2C_SCL1	DIO	VCC18IO
I2C_SDA1	DIO	VCC18IO
I2C_SCL2	DIO	VCC18IO
I2C_SDA2	DIO	VCC18IO
I2C_SCL3	DIO	VCC18IO
I2C_SDA3	DIO	VCC18IO
I2C_SCL4	DIO	VCC18IO
I2C_SDA4	DIO	VCC18IO
I2C_SCL5	DIO	VCC18IO
I2C_SDA5	DIO	VCC18IO
NOR_CK	DIO	VCC33IO_NOR
NOR_CS	DIO	VCC33IO_NOR
NOR_IO0	DIO	VCC33IO_NOR
NOR_IO1	DIO	VCC33IO_NOR
NOR_IO2	DIO	VCC33IO_NOR
NOR_IO3	DIO	VCC33IO_NOR
GBE_COL	DIO	VCC33IO_GBE
GBE_INTR	DIO	VCC33IO_GBE
GBE_MDC	DIO	VCC33IO_GBE
GBE_MDIO	DIO	VCC33IO_GBE

Pin Name	Type	Power Domain
GBE_RXC	DIO	VCC33IO_GBE
GBE_RXD0	DIO	VCC33IO_GBE
GBE_RXD1	DIO	VCC33IO_GBE
GBE_RXD2	DIO	VCC33IO_GBE
GBE_RXD3	DIO	VCC33IO_GBE
GBE_RXDV	DIO	VCC33IO_GBE
GBE_RXER	DIO	VCC33IO_GBE
GBE_TXC	DIO	VCC33IO_GBE
GBE_RXD0	DIO	VCC33IO_GBE
GBE_RXD1	DIO	VCC33IO_GBE
GBE_RXD2	DIO	VCC33IO_GBE
GBE_RXD3	DIO	VCC33IO_GBE
GBE_TXEN	DIO	VCC33IO_GBE
GBE_TXER	DIO	VCC33IO_GBE
NALE	DIO	VCC18IO
NCEB0	DIO	VCC18IO
NCEB1	DIO	VCC18IO
NCLE	DIO	VCC18IO
NF_DQS	DIO	VCC18IO
NREB	DIO	VCC18IO
NWEB	DIO	VCC18IO
MSDCOE_CLK	DIO	VCC18IO
MSDCOE_CMD	DIO	VCC18IO
MSDCOE_DAT0	DIO	VCC18IO
MSDCOE_DAT1	DIO	VCC18IO
MSDCOE_DAT2	DIO	VCC18IO
MSDCOE_DAT3	DIO	VCC18IO
MSDCOE_DAT4	DIO	VCC18IO
MSDCOE_DAT5	DIO	VCC18IO
MSDCOE_DAT6	DIO	VCC18IO
MSDCOE_DAT7	DIO	VCC18IO
MSDCOE_DSL	DIO	VCC18IO
MSDCOE_RSTB	DIO	VCC18IO
MSDC1_CLK	DIO	VCC33IO_MSDC1
MSDC1_CMD	DIO	VCC33IO_MSDC1
MSDC1_DAT0	DIO	VCC33IO_MSDC1
MSDC1_DAT1	DIO	VCC33IO_MSDC1
MSDC1_DAT2	DIO	VCC33IO_MSDC1
MSDC1_DAT3	DIO	VCC33IO_MSDC1
MSDC1_INS	DIO	VCC33IO_MSDC1
MSDC1_PSW	DIO	VCC33IO_MSDC1

Pin Name	Type	Power Domain
MSDC2_CLK	DIO	VCC33IO_MSDC2
MSDC2_CMD	DIO	VCC33IO_MSDC2
MSDC2_DAT0	DIO	VCC33IO_MSDC2
MSDC2_DAT1	DIO	VCC33IO_MSDC2
MSDC2_DAT2	DIO	VCC33IO_MSDC2
MSDC2_DAT3	DIO	VCC33IO_MSDC2
MSDC2_INS	DIO	VCC33IO_MSDC2
MSDC2_PSW	DIO	VCC33IO_MSDC2
MSDC3_CLK	DIO	VCC18IO
MSDC3_CMD	DIO	VCC18IO
MSDC3_DAT0	DIO	VCC18IO
MSDC3_DAT1	DIO	VCC18IO
MSDC3_DAT2	DIO	VCC18IO
MSDC3_DAT3	DIO	VCC18IO
MSDC3_DSL	DIO	VCC18IO
MSDC3_INS	DIO	VCC18IO
AU_MICBIAS0	AIO	AVDD33_AUADC
AU_VIN0_N	AIO	AVDD33_AUADC
AU_VIN0_P	AIO	AVDD33_AUADC
AU_VIN1_N	AIO	AVDD33_AUADC
AU_VIN1_P	AIO	AVDD33_AUADC
USB_DP_P0	AIO	AVDD33_USB_P0
USB_DM_P0	AIO	AVDD33_USB_P0
USB_VBUS_P0	AIO	AVDD18_USB_P0
DRV_VBUS_P0	DIO	VCC33IO_GPIO
IDDIG_P0	DIO	VCC33IO_GPIO
USB_DP_P1	AIO	AVDD33_USB_P1
USB_DM_P1	AIO	AVDD33_USB_P1
USB_VBUS_P1	AIO	AVDD18_USB_P1
DRV_VBUS_P1	DIO	VCC33IO_GBE
IDDIG_P1	DIO	VCC33IO_GBE
SSUSB_RXN_P2	AIO	AVDD10_SSUSB_P2
SSUSB_RXP_P2	AIO	AVDD10_SSUSB_P2
SSUSB_TXN_P2	AIO	AVDD10_SSUSB_P2
SSUSB_TXP_P2	AIO	AVDD10_SSUSB_P2
SSUSB_VRT_P2	AIO	AVDD18_SSUSB_P2
PCIe_CLKN_P2	AIO	AVDD18_SSUSB_P2
PCIe_CLKP_P2	AIO	AVDD18_SSUSB_P2
USB_DP_P2	AIO	AVDD33_USB_P2
USB_DM_P2	AIO	AVDD33_USB_P2
DRV_VBUS_P2	AIO	AVDD18_CVBS

Pin Name	Type	Power Domain
SSUSB_RXN_P3	AIO	AVDD10_SSUSB_P3
SSUSB_RXP_P3	AIO	AVDD10_SSUSB_P3
SSUSB_TXN_P3	AIO	AVDD10_SSUSB_P3
SSUSB_TXP_P3	AIO	AVDD10_SSUSB_P3
SSUSB_VRT_P3	AIO	AVDD18_SSUSB_P3
PCIe_CLKN_P3	AIO	AVDD18_SSUSB_P3
PCIe_CLKP_P3	AIO	AVDD18_SSUSB_P3
USB_DP_P3	AIO	AVDD33_USB_P3
USB_DM_P3	AIO	AVDD33_USB_P3
DRV_VBUS_P3	AIO	AVDD18_CVBS
CVBS_COM	AIO	AVDD18_CVBS
CVBSOP	AIO	AVDD18_CVBS
AUXINO	AIO	AVDD18_AP
AUXIN1	AIO	AVDD18_AP
AUXIN2	AIO	AVDD18_AP
REFP	AIO	AVDD18_PLLGP
XTALI	AIO	AVDD18_PLLGP
XTALO	AIO	AVDD18_PLLGP
AU_TN	AIO	AVDD18_AP
AU_TP	AIO	AVDD18_AP
CSI0_RCN	AIO	AVDD18_MIPIRX
CSI0_RCP	AIO	AVDD18_MIPIRX
CSI0_RDNO	AIO	AVDD18_MIPIRX
CSI0_RDN1	AIO	AVDD18_MIPIRX
CSI0_RDN2	AIO	AVDD18_MIPIRX
CSI0_RDN3	AIO	AVDD18_MIPIRX
CSI0_RDPO	AIO	AVDD18_MIPIRX
CSI0_RDP1	AIO	AVDD18_MIPIRX
CSI0_RDP2	AIO	AVDD18_MIPIRX
CSI0_RDP3	AIO	AVDD18_MIPIRX
CSI1_RCN	AIO	AVDD18_MIPIRX
CSI1_RCP	AIO	AVDD18_MIPIRX
CSI1_RDNO	AIO	AVDD18_MIPIRX
CSI1_RDN1	AIO	AVDD18_MIPIRX
CSI1_RDN2	AIO	AVDD18_MIPIRX
CSI1_RDN3	AIO	AVDD18_MIPIRX
CSI1_RDPO	AIO	AVDD18_MIPIRX
CSI1_RDP1	AIO	AVDD18_MIPIRX
CSI1_RDP2	AIO	AVDD18_MIPIRX
CSI1_RDP3	AIO	AVDD18_MIPIRX
DSI0_TCN	AIO	AVDD18_MIPITX_CH1

Pin Name	Type	Power Domain
DSI0_TCP	AIO	AVDD18_MIPITX_CH1
DSI0_TDN0	AIO	AVDD18_MIPITX_CH1
DSI0_TDN1	AIO	AVDD18_MIPITX_CH1
DSI0_TDN2	AIO	AVDD18_MIPITX_CH1
DSI0_TDN3	AIO	AVDD18_MIPITX_CH1
DSI0_TDPO	AIO	AVDD18_MIPITX_CH1
DSI0_TDP1	AIO	AVDD18_MIPITX_CH1
DSI0_TDP2	AIO	AVDD18_MIPITX_CH1
DSI0_TDP3	AIO	AVDD18_MIPITX_CH1
VRT0	AIO	AVDD18_MIPITX_CH1
DSI1_TCN	AIO	AVDD18_MIPITX_CH1
DSI1_TCP	AIO	AVDD18_MIPITX_CH1
DSI1_TDN0	AIO	AVDD18_MIPITX_CH1
DSI1_TDN1	AIO	AVDD18_MIPITX_CH1
DSI1_TDN2	AIO	AVDD18_MIPITX_CH1
DSI1_TDN3	AIO	AVDD18_MIPITX_CH1
DSI1_TDPO	AIO	AVDD18_MIPITX_CH1
DSI1_TDP1	AIO	AVDD18_MIPITX_CH1
DSI1_TDP2	AIO	AVDD18_MIPITX_CH1
DSI1_TDP3	AIO	AVDD18_MIPITX_CH1
VRT1	AIO	AVDD18_MIPITX_CH1
DSI2_TCN	AIO	AVDD18_MIPITX_CH2
DSI2_TCP	AIO	AVDD18_MIPITX_CH2
DSI2_TDN0	AIO	AVDD18_MIPITX_CH2
DSI2_TDN1	AIO	AVDD18_MIPITX_CH2
DSI2_TDN2	AIO	AVDD18_MIPITX_CH2
DSI2_TDN3	AIO	AVDD18_MIPITX_CH2
DSI2_TDPO	AIO	AVDD18_MIPITX_CH2
DSI2_TDP1	AIO	AVDD18_MIPITX_CH2
DSI2_TDP2	AIO	AVDD18_MIPITX_CH2
DSI2_TDP3	AIO	AVDD18_MIPITX_CH2
VRT2	AIO	AVDD18_MIPITX_CH2
DSI3_TCN	AIO	AVDD18_MIPITX_CH2
DSI3_TCP	AIO	AVDD18_MIPITX_CH2
DSI3_TDN0	AIO	AVDD18_MIPITX_CH2
DSI3_TDN1	AIO	AVDD18_MIPITX_CH2
DSI3_TDN2	AIO	AVDD18_MIPITX_CH2
DSI3_TDN3	AIO	AVDD18_MIPITX_CH2
DSI3_TDPO	AIO	AVDD18_MIPITX_CH2
DSI3_TDP1	AIO	AVDD18_MIPITX_CH2
DSI3_TDP2	AIO	AVDD18_MIPITX_CH2

Pin Name	Type	Power Domain
DSI3_TDP3	AIO	AVDD18_MIPITX_CH2
VRT3	AIO	AVDD18_MIPITX_CH2
CM2MCLK	DIO	VCC18IO
CMMCLK	DIO	VCC18IO
RTN	AIO	AVDD18_RDDR
RTP	AIO	AVDD18_RDDR
REXTDN	AIO	AVDD18_RDDR
ARA0	AIO	DDRV_ARDDR
ARA1	AIO	DDRV_ARDDR
ARA2	AIO	DDRV_ARDDR
ARA3	AIO	DDRV_ARDDR
ARA4	AIO	DDRV_ARDDR
ARA5	AIO	DDRV_ARDDR
ARA6	AIO	DDRV_ARDDR
ARA7	AIO	DDRV_ARDDR
ARA8	AIO	DDRV_ARDDR
ARA9	AIO	DDRV_ARDDR
ARA10	AIO	DDRV_ARDDR
ARA11	AIO	DDRV_ARDDR
ARA12	AIO	DDRV_ARDDR
ARA13	AIO	DDRV_ARDDR
ARBA0	AIO	DDRV_ARDDR
ARBA1	AIO	DDRV_ARDDR
ARBG0	AIO	DDRV_ARDDR
ARCLK	AIO	DDRV_ARDDR
ARCLK#	AIO	DDRV_ARDDR
ARCS0	AIO	DDRV_ARDDR
ARCKE0	AIO	DDRV_ARDDR
ARRAS	AIO	DDRV_ARDDR
ARCAS	AIO	DDRV_ARDDR
ARWE#	AIO	DDRV_ARDDR
ARODT	AIO	DDRV_ARDDR
ARACT#	AIO	DDRV_ARDDR
ARRESET#	AIO	DDRV_ARDDR
ARDQ0	AIO	DDRV_ARDDR
ARDQ1	AIO	DDRV_ARDDR
ARDQ2	AIO	DDRV_ARDDR
ARDQ3	AIO	DDRV_ARDDR
ARDQ4	AIO	DDRV_ARDDR
ARDQ5	AIO	DDRV_ARDDR
ARDQ6	AIO	DDRV_ARDDR

Pin Name	Type	Power Domain
ARDQ7	AIO	DDRV_ARDDR
ARDQ8	AIO	DDRV_ARDDR
ARDQ9	AIO	DDRV_ARDDR
ARDQ10	AIO	DDRV_ARDDR
ARDQ11	AIO	DDRV_ARDDR
ARDQ12	AIO	DDRV_ARDDR
ARDQ13	AIO	DDRV_ARDDR
ARDQ14	AIO	DDRV_ARDDR
ARDQ15	AIO	DDRV_ARDDR
ARDQM0	AIO	DDRV_ARDDR
ARDQM1	AIO	DDRV_ARDDR
ARDQS0	AIO	DDRV_ARDDR
ARDQS0#	AIO	DDRV_ARDDR
ARDQS1	AIO	DDRV_ARDDR
ARDQS1#	AIO	DDRV_ARDDR
BRA0	AIO	DDRV_BRDDR
BRA1	AIO	DDRV_BRDDR
BRA2	AIO	DDRV_BRDDR
BRA3	AIO	DDRV_BRDDR
BRA4	AIO	DDRV_BRDDR
BRA5	AIO	DDRV_BRDDR
BRA6	AIO	DDRV_BRDDR
BRA7	AIO	DDRV_BRDDR
BRA8	AIO	DDRV_BRDDR
BRA9	AIO	DDRV_BRDDR
BRA10	AIO	DDRV_BRDDR
BRA11	AIO	DDRV_BRDDR
BRA12	AIO	DDRV_BRDDR
BRA13	AIO	DDRV_BRDDR
BRBA0	AIO	DDRV_BRDDR
BRBA1	AIO	DDRV_BRDDR
BRBG0	AIO	DDRV_BRDDR
BRCLK	AIO	DDRV_BRDDR
BRCLK#	AIO	DDRV_BRDDR
BRCS0	AIO	DDRV_BRDDR
BRCKE0	AIO	DDRV_BRDDR
BRRAS	AIO	DDRV_BRDDR
BCAS	AIO	DDRV_BRDDR
BRWE#	AIO	DDRV_BRDDR
BRODT	AIO	DDRV_BRDDR
BRACT#	AIO	DDRV_BRDDR

Pin Name	Type	Power Domain
BRRESET#	AIO	DDRV_BRDDR
BRDQ0	AIO	DDRV_BRDDR
BRDQ1	AIO	DDRV_BRDDR
BRDQ2	AIO	DDRV_BRDDR
BRDQ3	AIO	DDRV_BRDDR
BRDQ4	AIO	DDRV_BRDDR
BRDQ5	AIO	DDRV_BRDDR
BRDQ6	AIO	DDRV_BRDDR
BRDQ7	AIO	DDRV_BRDDR
BRDQ8	AIO	DDRV_BRDDR
BRDQ9	AIO	DDRV_BRDDR
BRDQ10	AIO	DDRV_BRDDR
BRDQ11	AIO	DDRV_BRDDR
BRDQ12	AIO	DDRV_BRDDR
BRDQ13	AIO	DDRV_BRDDR
BRDQ14	AIO	DDRV_BRDDR
BRDQ15	AIO	DDRV_BRDDR
BRDQM0	AIO	DDRV_BRDDR
BRDQM1	AIO	DDRV_BRDDR
BRDQS0	AIO	DDRV_BRDDR
BRDQS0#	AIO	DDRV_BRDDR
BRDQS1	AIO	DDRV_BRDDR
BRDQS1#	AIO	DDRV_BRDDR
CRA0	AIO	DDRV_CRDDR
CRA1	AIO	DDRV_CRDDR
CRA2	AIO	DDRV_CRDDR
CRA3	AIO	DDRV_CRDDR
CRA4	AIO	DDRV_CRDDR
CRA5	AIO	DDRV_CRDDR
CRA6	AIO	DDRV_CRDDR
CRA7	AIO	DDRV_CRDDR
CRA8	AIO	DDRV_CRDDR
CRA9	AIO	DDRV_CRDDR
CRA10	AIO	DDRV_CRDDR
CRA11	AIO	DDRV_CRDDR
CRA12	AIO	DDRV_CRDDR
CRA13	AIO	DDRV_CRDDR
CRBA0	AIO	DDRV_CRDDR
CRBA1	AIO	DDRV_CRDDR
CRBG0	AIO	DDRV_CRDDR
CRCLK	AIO	DDRV_CRDDR

Pin Name	Type	Power Domain
CRCLK#	AIO	DDRV_CRDDR
CRCS0	AIO	DDRV_CRDDR
CRCKE0	AIO	DDRV_CRDDR
CRRAS	AIO	DDRV_CRDDR
CRCAS	AIO	DDRV_CRDDR
CRWE#	AIO	DDRV_CRDDR
CRODT	AIO	DDRV_CRDDR
CRACT#	AIO	DDRV_CRDDR
CRRESET#	AIO	DDRV_CRDDR
CRDQ0	AIO	DDRV_CRDDR
CRDQ1	AIO	DDRV_CRDDR
CRDQ2	AIO	DDRV_CRDDR
CRDQ3	AIO	DDRV_CRDDR
CRDQ4	AIO	DDRV_CRDDR
CRDQ5	AIO	DDRV_CRDDR
CRDQ6	AIO	DDRV_CRDDR
CRDQ7	AIO	DDRV_CRDDR
CRDQ8	AIO	DDRV_CRDDR
CRDQ9	AIO	DDRV_CRDDR
CRDQ10	AIO	DDRV_CRDDR
CRDQ11	AIO	DDRV_CRDDR
CRDQ12	AIO	DDRV_CRDDR
CRDQ13	AIO	DDRV_CRDDR
CRDQ14	AIO	DDRV_CRDDR
CRDQ15	AIO	DDRV_CRDDR
CRDQM0	AIO	DDRV_CRDDR
CRDQM1	AIO	DDRV_CRDDR
CRDQS0	AIO	DDRV_CRDDR
CRDQS0#	AIO	DDRV_CRDDR
CRDQS1	AIO	DDRV_CRDDR
CRDQS1#	AIO	DDRV_CRDDR
DRA0	AIO	DDRV_DRDDR
DRA1	AIO	DDRV_DRDDR
DRA2	AIO	DDRV_DRDDR
DRA3	AIO	DDRV_DRDDR
DRA4	AIO	DDRV_DRDDR
DRA5	AIO	DDRV_DRDDR
DRA6	AIO	DDRV_DRDDR
DRA7	AIO	DDRV_DRDDR
DRA8	AIO	DDRV_DRDDR
DRA9	AIO	DDRV_DRDDR

Pin Name	Type	Power Domain
DRA10	AIO	DDRV_DRDDR
DRA11	AIO	DDRV_DRDDR
DRA12	AIO	DDRV_DRDDR
DRA13	AIO	DDRV_DRDDR
DRBA0	AIO	DDRV_DRDDR
DRBA1	AIO	DDRV_DRDDR
DRBG0	AIO	DDRV_DRDDR
DRCLK	AIO	DDRV_DRDDR
DRCLK#	AIO	DDRV_DRDDR
DRCS0	AIO	DDRV_DRDDR
DRCKE0	AIO	DDRV_DRDDR
DRRAS	AIO	DDRV_DRDDR
DRCAS	AIO	DDRV_DRDDR
DRWE#	AIO	DDRV_DRDDR
DRODT	AIO	DDRV_DRDDR
DRACT#	AIO	DDRV_DRDDR
DRRESET#	AIO	DDRV_DRDDR
DRDQ0	AIO	DDRV_DRDDR
DRDQ1	AIO	DDRV_DRDDR
DRDQ2	AIO	DDRV_DRDDR
DRDQ3	AIO	DDRV_DRDDR
DRDQ4	AIO	DDRV_DRDDR
DRDQ5	AIO	DDRV_DRDDR
DRDQ6	AIO	DDRV_DRDDR
DRDQ7	AIO	DDRV_DRDDR
DRDQ8	AIO	DDRV_DRDDR
DRDQ9	AIO	DDRV_DRDDR
DRDQ10	AIO	DDRV_DRDDR
DRDQ11	AIO	DDRV_DRDDR
DRDQ12	AIO	DDRV_DRDDR
DRDQ13	AIO	DDRV_DRDDR
DRDQ14	AIO	DDRV_DRDDR
DRDQ15	AIO	DDRV_DRDDR
DRDQM0	AIO	DDRV_DRDDR
DRDQM1	AIO	DDRV_DRDDR
DRDQS0	AIO	DDRV_DRDDR
DRDQS0#	AIO	DDRV_DRDDR
DRDQS1	AIO	DDRV_DRDDR
DRDQS1#	AIO	DDRV_DRDDR

Pin Name	Type	Description	1.8 V	3.3 V
Analog Power				
AVDD33_USB_P0	P	3.3 V power for USB20 P0	-	V
AVDD33_USB_P1	P	3.3 V power for USB20 P1	-	V
AVDD33_USB_P2	P	3.3 V power for USB20 P2	-	V
AVDD33_USB_P3	P	3.3 V power for USB20 P3	-	V
AVDD18_USB_P0	P	1.8 V power for USB20 P0	V	-
AVDD18_USB_P1	P	1.8 V power for USB20 P1	V	-
AVDD18_USB_P2	P	1.8 V power for USB20 P2	V	-
AVDD18_USB_P3	P	1.8 V power for USB20 P3	V	-
AVDD18_SSUSB_P2	P	1.8 V power for SSUSB P2	V	-
AVDD10_SSUSB_P2	P	1.05 V power for SSUSB P2		
AVDD18_SSUSB_P3	P	1.8 V power for SSUSB P3	V	-
AVDD10_SSUSB_P3	P	1.05 V power for SSUSB P3		
AVDD22_AUADC	P	2.2 V power for AUADC		
AVDD33_AUADC	P	3.3 V power for AUADC	-	V
AVDD18_AP	P	1.8 V power for AUXADC	V	-
AVDD18_PLLGP	P	1.8 V power for PLLs	V	-
AVDD18_CVBS	P	1.8 V power for CVBS	V	-
AVDD18_MIPIRX	P	1.8 V power for MIPIRX	V	-
AVDD18_MIPITX_CH1	P	1.8 V power for MIPITX	V	-
AVDD18_MIPITX_CH2	P	1.8 V power for MIPITX	V	-
Digital Power				
VCC18IO	P	IO Power	V	
VCC18IO_RTC	P	IO Power	V	
VCC33IO_GPIO	P	IO Power	V	V
VCC33IO_I2SI02	P	IO Power	V	V
VCC33IO_I2SI01	P	IO Power	V	V
VCC33IO_I2SO02	P	IO Power	V	V
VCC33IO_MSDC1	P	IO Power	V	V
VCC33IO_MSDC2	P	IO Power	V	V
VCC33IO_NOR	P	IO Power	V	V
VCC33IO_SPICTP	P	IO Power	V	V
VCC33IO_SPIMCU	P	IO Power	V	V
VCC33IO_TDMO	P	IO Power	V	V
VCC33IO_UART4B	P	IO Power	V	V
VCC33IO_GBE	P	IO Power	V	V
VCC18IO_EFUSE	P	IO Power	V	
VCCK_BCPU	P	Core Power		
VCCK_SRAM_BCPU	P	Core Power		
VCCK_LCPU	P	Core Power		
VCCK_SRAM_LCPU	P	Core Power		

Pin Name	Type	Description	1.8 V	3.3 V
VCCK	P	Core Power		
Analog Ground				
AVSS18_AP	G	-	-	-
AVSS_REFN	G	-	-	-
AVSS33_AUADC	G	-		
AVSS18_CVBS	G	-	-	-
AVSS18_MIPIRX	G	-	-	-
AVSS18_MIPITX_CH1	G	-	-	-
AVSS18_MIPITX_CH2	G	-	-	-
Digital Ground				
DVSS	G	-	-	-

2.2 Electrical Characteristics

2.2.1 Absolute Maximum Ratings

Table 2-4 Absolute Maximum Ratings

Symbol or Pin Name	Description	Min.	Max.	Unit
AVDD18_PLLGP	Analog power input 1.8 V for PLL	-0.3	2.1	V
AVDD18_AP	Analog power input 1.8 V for AUXADC & Tsense	-0.3	2.1	V
AVDD18_MIPITX_CH1	Analog power 1.8 V for MIPI CSI	-0.3	2.1	V
AVDD18_MIPITX_CH2	Analog power 1.8 V for MIPI DS1	-0.3	2.1	V
AVDD18_MIPIRX	Analog power 1.8 V for MIPI CS1	-0.3	2.1	V
AVDD33_USB_P0	Analog power 3.3 V for USB	-0.3	3.63	V
AVDD33_USB_P1				
AVDD33_USB_P2				
AVDD33_USB_P3				
AVDD18_USB_P0	Analog power 1.8 V for USB	-0.3	2.1	V
AVDD18_USB_P1				
AVDD18_USB_P2				
AVDD18_USB_P3				
AVDD18_SSUSB_P2	Analog power 1.8 V for SSUSB	-0.3	2.1	V
AVDD18_SSUSB_P3				
AVDD33_AUADC	Analog power 3.3 V for AACD	-0.3	3.63	V
AVDD22_AUADC	Analog power 2.2 V for AACD	-0.3	2.3	V
AVDD18_CVBS	Analog power 1.8 V for CVBS	-0.3	2.1	V
AVDD10_SSUSB_P2	Analog power 1.0 V for SSUSB	-0.3	1.2	V
AVDD10_SSUSB_P3				
VCC18IO	Digital power input for 1.8 V IO	-0.3	2.1	V
VCC18IO_RTC	Digital power input for 1.8 V IO	-0.3	2.1	V

Symbol or Pin Name	Description	Min.	Max.	Unit
VCC33IO_GPIO				
VCC33IO_I2S1O2	Digital power input for 3.3 V IO	-0.3	3.63	V
VCC33IO_I2S1O1				
VCC33IO_I2S0O2				
VCC33IO_MSDC1				
VCC33IO_MSDC2				
VCC33IO_NOR				
VCC33IO_SPICTP				
VCC33IO_SPIMCU	Digital power input for 1.8 V IO	-0.3	2.1	V
VCC33IO_TDMO				
VCC33IO_UART4B				
VCC33IO_GBE				
VCC18IO_EFUSE	Digital power input when eFuse is burning	-0.3	2.1	V
	Digital power input when eFuse is not burning	0.0	0.0	V
DDRV_ARDDR				
DDRV_BRDDR	Analog power input for DDRPHY(LPDDR4)	-0.4	1.5	V
DDRV_CRDDR				
DDRV_DRDDR				
VCCK_BCPU	Digital power input for Big CPU	-0.3	1.2	V
VCCK_SRAM_BCPU	Digital power input for Big CPU SRAM	-0.3	1.2	V
VCCK_LCPU	Digital power input for Small CPU	-0.3	1.2	V
VCCK_SRAM_LCPU	Digital power input for Small CPU SRAM	-0.3	1.2	V
VCCK	Digital power input for CORE&GPU	-0.3	1.2	V
T _{STG}	Storage Temperature	-55	150	°C

Warning: Stressing the device beyond the absolute maximum ratings may cause permanent damage. These are stress ratings only.

Note: Subject to change.

2.2.2 Recommended Operating Conditions

Table 2-5 Recommended Operating Conditions for Power Supply

Symbol or Pin Name	Description	Min.	Typ.	Max.	Unit
AVDD18_PLLGP	Analog power input 1.8 V for PLL	1.71	1.8	1.89	V
AVDD18_AP	Analog power input 1.8 V for AUXADC & Tsense	1.71	1.8	1.89	V
AVDD18_MIPITX_CH1	Analog power 1.8 V for MIPI DSI	1.71	1.8	1.89	V
AVDD18_MIPITX_CH2					
AVDD18_MIPIRX	Analog power 1.8 V for MIPI CSI	1.71	1.8	1.89	V
AVDD33_USB_P0					
AVDD33_USB_P1	Analog power 3.3 V for USB	3.135	3.3	3.465	V
AVDD33_USB_P2					
AVDD33_USB_P3					

Symbol or Pin Name	Description	Min.	Typ.	Max.	Unit
AVDD18_USB_P0					
AVDD18_USB_P1	Analog power 1.8 V for USB	1.71	1.8	1.89	V
AVDD18_USB_P2					
AVDD18_USB_P3					
AVDD18_SSUSB_P2	Analog power 1.8 V for SSUSB	1.71	1.8	1.89	V
AVDD18_SSUSB_P3					
AVDD33_AUADC	Analog power 3.3 V for AACD	3.135	3.3	3.465	V
AVDD22_AUADC	Analog power 2.2 V for AACD	2.05	2.15	2.25	V
AVDD18_CVBS	Analog power 1.8 V for CVBS	1.71	1.8	1.89	V
AVDD10_SSUSB_P2	Analog power 1.0 V for SSUSB	0.9975	1.05	1.1025	V
AVDD10_SSUSB_P3					
VCC18IO	Digital power input for 1.8 V IO	1.7	1.8	1.9	V
VCC18IO_RTC	Digital power input for 1.8 V IO	1.62	1.8	1.98	V
VCC33IO_GPIO					
VCC33IO_I2SIO2					
VCC33IO_I2SIO1	Digital power input for 3.3 V IO	2.97	3.3	3.63	V
VCC33IO_I2S002					
VCC33IO_MSDC1					
VCC33IO_MSDC2					
VCC33IO_NOR					
VCC33IO_SPICTP					
VCC33IO_SPIMCU					
VCC33IO_TDMO					
VCC33IO_UART4B					
VCC33IO_GBE					
VCC18IO_EFUSE	Digital power input when eFuse is burning	1.7	1.8	1.98	V
	Digital power input when eFuse is not burning	0.0	0.0	0.0	V
DDR_V_ARDDR					
DDR_V_BRDDR					
DDR_V_CRDDR	Analog power input for DDRPHY(LPDDR4)	1.06	1.10	1.17	V
DDR_V_DRDDR					
VCCK_BCPU (MT2712S/A/H only)	Digital power input for Big CPU	1.00	1.05	1.10	V
VCCK_SRAM_BCPU (MT2712S/A/H only)	Digital power input for Big CPU SRAM	1.00	1.05	1.10	V
VCCK_LCPU (MT2712S/A/H only)	Digital power input for Small CPU	0.95	1.0	1.10	V
VCCK_SRAM_LCPU (MT2712S/A/H only)	Digital power input for Small CPU SRAM	0.95	1.0	1.10	V
VCCK_BCPU (None MT2712S/A/H)	Digital power input for Big CPU	0.95	1.0	1.05	V
VCCK_SRAM_BCPU (None MT2712S/A/H)	Digital power input for Big CPU SRAM	0.95	1.0	1.05	V

Symbol or Pin Name	Description	Min.	Typ.	Max.	Unit
VCCK_LCPU (None MT2712S/A/H)	Digital power input for Small CPU	0.95	1.0	1.05	V
VCCK_SRAM_LCPU (None MT2712S/A/H)	Digital power input for Small CPU SRAM	0.95	1.0	1.05	V
VCCK	Digital power input for CORE&GPU(Active mode & Deep Idle mode & Suspend B mode)	0.95	1.0	1.05	V
	Digital power input for CORE&GPU(Suspend A mode)	0.665	0.7	0.735	V

Warning: System power design for MT2712 needs to pay attention to system-level power integrity and PCB PDN Mask, including Rdc, remote and equivalent RLC, to make sure the power supply to SoC is under the recommended requirement range to ensure functionality. If the power supply is out of specification, MT2712 may not work, especially VCCK_BCPU/VCCK_SRAM_BCPU/VCCK_LCPU/VCCK_SRAM_LCPU/VCCK. Customers can confirm power by power integrity simulation, or work with MediaTek technical support engineers to confirm system power design.

Note: Subject to change.

2.2.3 Storage Conditions

- Shelf life in sealed bag: 12 months at < 40°C and < 90% relative humidity (RH).
- After the bag is opened, devices subjected to infrared reflow, vapor-phase reflow or equivalent processing must be:
 - Mounted within 168 hours in factory condition of 30°C/60% RH, or
 - Stored at 20% RH
- Devices require baking before being mounted, if they are placed
 - For 192 hours at 40°C +5°C/-0°C and < 5% RH in low temperature device containers.
 - For 24 hours at 125°C +5°C/-0°C in high temperature device containers.

2.2.4 AC Electrical Characteristics and Timing Diagram

2.2.4.1 External Memory Interface for LPDDR4

The External Memory Interface (EMI), shown in Figure 2-2, Figure 2-3 and Figure 2-4, is used to connect LPDDR4 device for MT2712. It includes pins CLK_T, CLK_C, CKE, CS, DQS_T[1:0], DQS_C[1:0], CA[5:0] and DQ[15:0]. Table 2-6 summarizes the symbol definition and the related timing specifications.

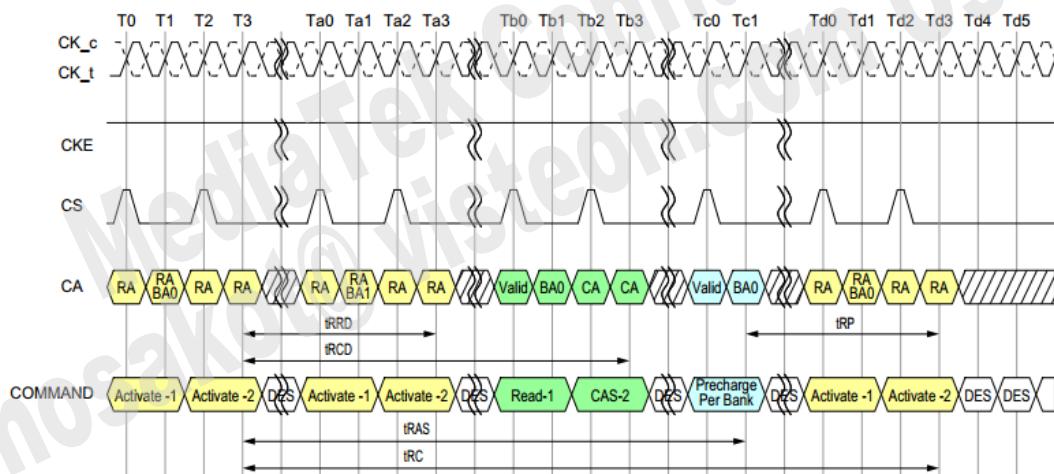


Figure 2-2 Basic Timing Parameter for LPDDR4 Commands

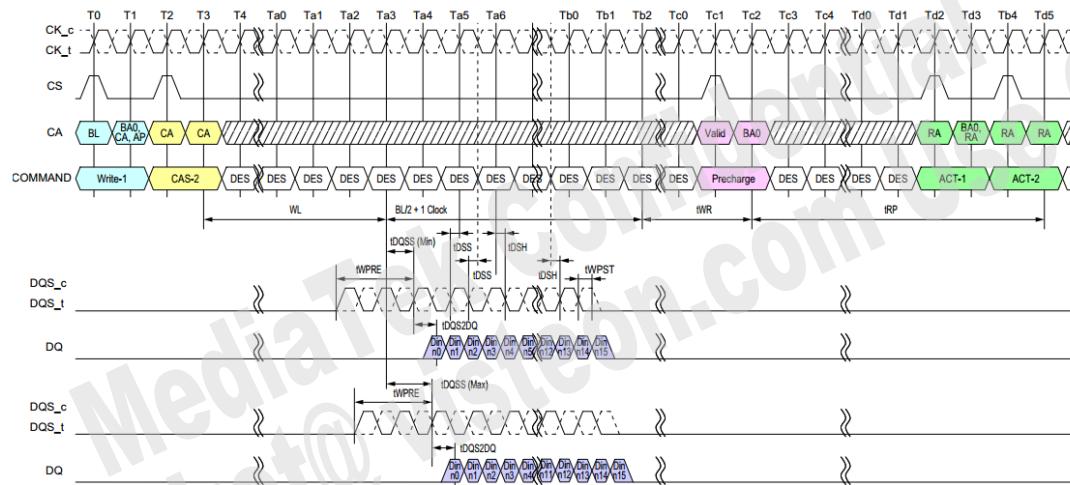


Figure 2-3 Basic Timing Parameter for LPDDR4 Write

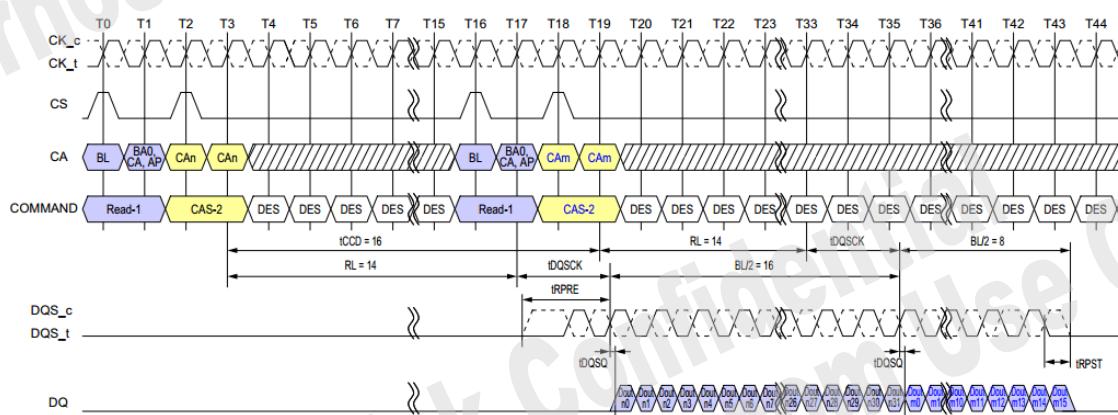


Figure 2-4 Basic Timing Parameter for LPDDR4 Read

Table 2-6 LPDDR4 AC Timing Parameter for External Memory Interface

Symbol	Description	Min.	Typ.	Max.	Unit
Clock Timing					
tCK	Clock cycle time	0.625	-	100	ns
tCH	Clock high level width	0.46	-	0.54	tCK(avg)
tCL	Clock low level width	0.46	-	0.54	tCK(avg)
Core Timing					
tRC	ACTIVATE-to-ACTIVATE command period (same bank)	tRAS + tRPab (with all-bank precharge) tRAS + tRPpb (with per-bank precharge)	-	-	ns
tSR	Minimum Self-Refresh Time (Entry to Exit)	Max(15ns, 3nCK)	-	-	ns
tXSR	SELF REFRESH exit to next valid command delay	Max(tRFCab + 7.5ns, 2nCK)	-	-	ns
tXP	Exit Power-Down to next valid command delay	Max(7.5ns, 3nCK)	-	-	ns
tCCD	CAS-to-CAS delay	8	-	-	tCK(avg)
tRTP	Internal READ to PRECHARGE command delay	Max(7.5ns, 8nCK)	-	-	ns
tDQSK	DQS output access time from CK_t/CK_c (derated)	1.5	-	3.5	ns
tRCD	RAS-to-CAS delay	Max(18ns, 4nCK)	-	-	ns
tRPpb	Row precharge time (single bank)	Max(18ns, 3nCK)	-	-	ns
tRPpb	Row precharge time (all banks)	Max(21ns, 3nCK)	-	-	ns
tRAS	Row active time	Max(42ns, 3nCK)	-	Min(9 * tREFI * Refresh Rate, 70.2) us (Refresh Rate is specified by MR4, OP[2:0])	ns
tWR	WRITE recovery time	Max(18ns, 4nCK)	-	-	ns
tWTR	WRITE-to-READ delay	Max(10ns, 8nCK)	-	-	ns
tRRD	Active bank-A to active bank-B	Max(10ns, 4nCK)	-	-	ns
tPPD	Precharge to Precharge Delay	4	-	-	tCK(avg)
tFAW	Four-bank ACTIVATE window	40	-	-	ns
Read Data Timing					
tDQSQ	DQS_t,DQS_c to DQ Skew total, per group, per access (DBI-Disabled)	-	-	0.18	UI
tQH	DQ output hold time total from DQS_t, DQS_c (DBI-Disabled)	Min(tQSH, tQSL)	-	-	UI

Symbol	Description	Min.	Typ.	Max.	Unit
tQW_total	DQ output window time total, per pin (DBI-Disabled)	0.7	-	-	UI
tQH_DB1	DQ output hold time total from DQS_t, DQS_c (DBI-Enabled)	Min(tQSH_DB1, tQSL_DB1)	-	-	UI
tQSL	DQS, DQS# differential output low time (DBI-Disabled)	tCL(abs) -0.05	-	-	tCK(avg)
tQSH	DQS, DQS# differential output high time (DBI-Disabled)	tCH(abs) -0.05	-	-	tCK(avg)
tQSL_DB1	DQS, DQS# differential output low time (DBI-Enabled)	tCL(abs) -0.045	-	-	tCK(avg)
tQSH_DB1	DQS, DQS# differential output high time (DBI-Enabled)	tCH(abs) -0.045	-	-	tCK(avg)
Writing Timing					
tDQS2DQ	DQ to DQS offset	0.2	-	0.8	ns
tDQDQ	DQ to DQ offset	-	-	0.03	ns
tDQS2DQ_temp	DQ to DQS offset temperature variation	-	-	0.6	ps/°C
tDQS2DQ_volt	DQ to DQS offset voltage variation	-	-	30	ps/mv

*Unit UI = tCK(avg)min/2

2.2.4.2 1.8V IO

Table 2-7 General 1.8V IO DC Timing Parameters

Parameter	Description	Min.	Typ.	Max.	Unit	Note
VCC18IO	Supply voltage of IO power	1.7	1.8	1.9	V	-
V _{IH}	Input logic high voltage	0.65*VDDIO	-	VDDIO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.35*VDDIO	V	-
R _{pu}	Input pull-up resistance	40	75	190	KΩ	Control pin PU = 1
R _{pd}	Input pull-down resistance	40	75	190	KΩ	Control pin PD = 1
V _{OH} (DC)	DC Output logic high voltage	0.75*VDDIO	-	-	V	-
V _{OL} (DC)	DC Output logic low voltage	-	-	0.25*VDDIO	V	-
I _{IN}	Input leakage current (any input 0V < V _{IN} < VDDIO)	-5	-	5	μA	-
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	-

2.2.4.3 3.3V IO

Table 2-8 General 3.3V IO DC Timing Parameters

AC&DC Operating Conditions of 3.3V Applications						
Parameter	Description	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	2.97	3.3	3.63	V	-
V _{OH} (DC)	DC Output high voltage	VCC33IO-0.4	-	VCC33IO+0.3	V	
V _{OL} (DC)	DC Output low voltage	-0.3	-	0.4	V	
V _{IH}	Input logic high voltage	2.0	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.8	-	-
R _{pu}	Input pull-up resistance	40	75	190	KΩ	Control pin PU = 1
R _{pd}	Input pull-down resistance	40	75	190	KΩ	Control pin PD = 1
AC&DC Operating Conditions of 1.8V Applications						
Parameter	Description	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	1.7	1.8	1.9	V	1
V _{OH} (DC)	DC Output logic high voltage	VCC33IO-0.2	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC Output logic low voltage	-0.3	-	0.2	V	-
V _{IH}	Input logic high voltage	1.27	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.58	V	-
R _{pu}	Input pull-up resistance	10	50	100	KΩ	Control pin PU = 1
R _{pd}	Input pull-down resistance	10	50	100	KΩ	Control pin PD = 1
I _{IN}	Input leakage current(any input 0V < V _{IN} < V _{CCK})	-5	-	5	μA	3
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	3

2.2.4.4 MSDC IO**Table 2-9 VCC33IO_MSDC1/2 DC Timing Parameters**

AC&DC Operating Conditions of 3.3V Applications						
Parameters	Description	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	2.97	3.3	3.63	V	-
V _{OH} (DC)	DC Output logic high voltage	0.75*VCC33IO	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC Output logic low voltage	-0.3	-	0.125*VCC33IO	V	-
V _{IH}	Input logic high voltage	0.625*VCC33IO	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.25*VCC33IO	V	-

AC&DC Operating Conditions of 3.3V Applications						
Parameters	Description	Min.	Typ.	Max.	Unit	Note
R _{pu1}	Input pull-up resistance	10	50	100	KΩ	Refer to Table 2-11, control pin R0 = 0, R1 = 1
R _{pd1}	Input pull-down resistance	10	50	100	KΩ	Refer to Table 2-11, control pin R0 = 0, R1 = 1
R _{pu0}	Input pull-up resistance	5	7.5	10	KΩ	Refer to Table 2-11, control pin R0 = 1, R1 = 0
R _{pd0}	Input pull-down resistance	5	7.5	10	KΩ	Refer to Table 2-11, control pin R0 = 1, R1 = 0
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC33IO)	-5	-	5	μA	-
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	-
AC&DC Operating Conditions of 1.8V Applications						
Parameters	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	1.7	1.8	1.9	V	-
V _{OH(DC)}	DC Output logic high voltage	1.4	-	VCC33IO+0.3	V	-
V _{OL(DC)}	DC Output logic low voltage	-0.3	-	0.45	V	-
V _{IH}	Input logic high voltage	1.27	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.58	V	-
R _{pu1}	Input pull-up resistance	10	50	100	KΩ	Refer to Table 2-11, control pin R0 = 0, R1 = 1
R _{pd1}	Input pull-down resistance	10	50	100	KΩ	Refer to Table 2-11, control pin R0 = 0, R1 = 1
R _{pu0}	Input pull-up resistance	5	7.5	10	KΩ	Refer to Table 2-11, control pin R0 = 1, R1 = 0
R _{pd0}	Input pull-down resistance	5	7.5	10	KΩ	Refer to Table 2-11, control pin R0 = 1, R1 = 0
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC33IO)	-5	-	5	μA	-
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	-

Table 2-10 VCC18IO for MSDCOE/MSDC3 DC Timing Parameters

Parametes	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC18IO	Supply voltage of IO power	1.7	1.8	1.9	V	-
V _{IH}	Input logic high voltage	VCC18IO-0.45	-	VCC18IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.58	V	-
R _{pu1}	Input pull-up resistance	10	50	90	KΩ	Refer to Table 2-11, control pin R0 = 0, R1 = 1

Parametes	Descriptions	Min.	Typ.	Max.	Unit	Note
R_{pd1}	Input pull-down resistance	10	50	90	KΩ	Refer to Table 2-11, control pin R0 = 0, R1 = 1
R_{pu0}	Input pull-up resistance	5	7.5	10	KΩ	Refer to Table 2-11, control pin R0 = 1, R1 = 0
R_{pd0}	Input pull-down resistance	5	7.5	10	KΩ	Refer to Table 2-11, control pin R0 = 1, R1 = 0
$V_{OH(DC)}$	DC Output logic high voltage	1.4	-	-	V	-
$V_{OL(DC)}$	DC Output logic low voltage	-	-	0.45	-	-
C_{load}	external capacitance loading	3	-	15	pF	-
IIN	Input leakage current (any input $0V < V_{IN} < VDDIO$)	-2	-	2	μA	-
IOZ	Tri-state output leakage current	-2	-	2	μA	-

Table 2-11 MSDC Note

PUPD	R1	R0	Resistance Value
0	0	0	High-Z
0	0	1	PU10 Kohm
0	1	0	PU50 Kohm
0	1	1	PU10 Kohm//50 Kohm
1	0	0	High-Z
1	0	1	PD10 Kohm
1	1	0	PD50 Kohm
1	1	1	PD10 Kohm//50 Kohm

Table 2-12 MSDC Overshoot/Undershoot

Parameter	Description	Min.	Typ.	Max.	Unit	Note
Overshoot	Maximum peak amplitude allowed for overshoot area	-	-	0.9	V	-
Undershoot	Maximum peak amplitude allowed for undershoot area	-	-	0.9	V	-

2.3 System Configuration

2.3.1 Constant Tied Pins

Table 2-13 Constant Tied Pins

Pin Name	Description
TESTMODE	Test mode (tied to GND)

Pin Name	Description
FSOURCE_P	eFuse blowing (tied to GND)

2.4 Power-On Sequence

The power-on/off sequence with XTAL is shown in Figure 2-5.

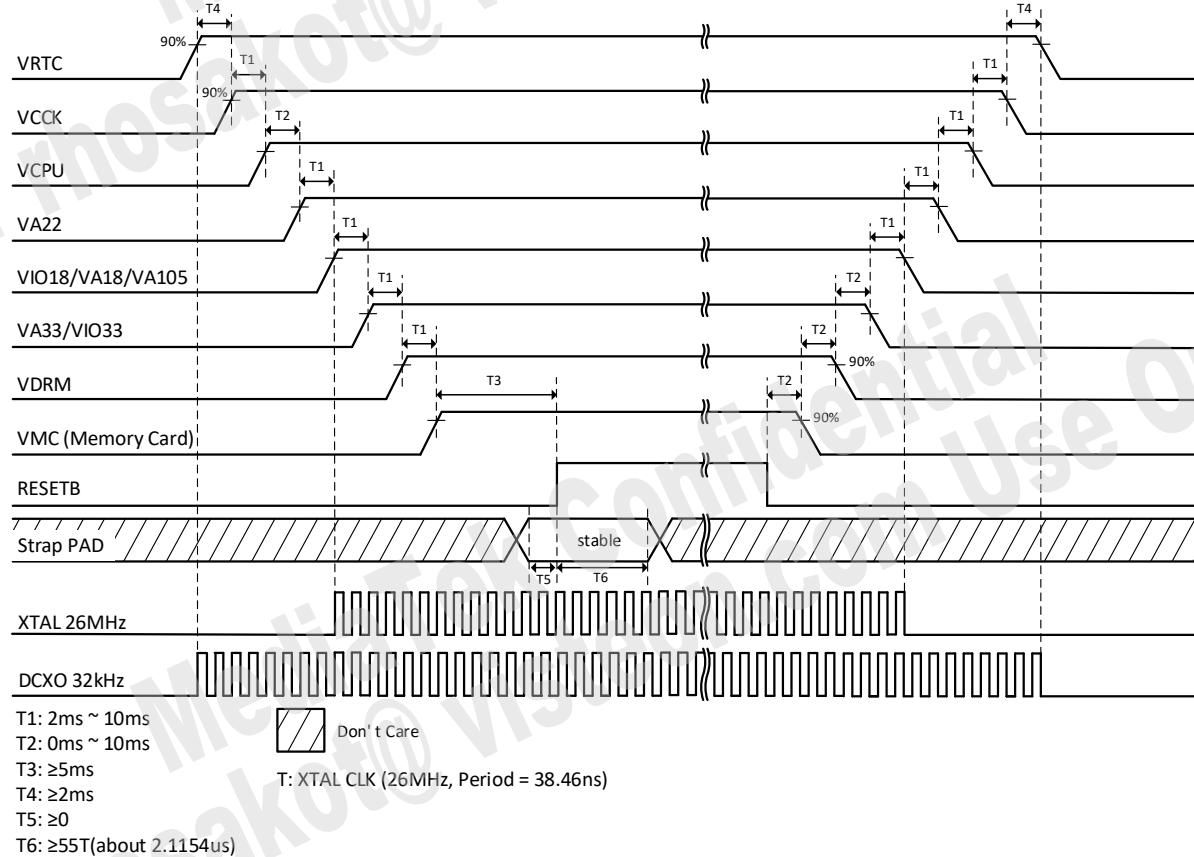


Figure 2-5 Power-On/Off Sequence with XTAL

Note that Figure 2-5 only shows one power-on/off condition with XTAL. The external circuit handles the power-on and power-off of the device.

For details about the signals shown in Figure 2-5, please refer to Table 2-14.

Table 2-14 Signals Details

PWRKEY	User Define
VRTC	Voltage of RTC: VCC18IO_RTC

PWRKEY	User Define
VCKK	Voltage of Core: VCKK Voltage of GPU (same with VCORE) VCKK Voltage of Analog Core (same with VCORE): VCKK
VCPU	Voltage of Processor: VCKK_BCPU; VCKK_SRAM_BCPU; VCKK_LCPU; VCKK_SRAM_LCPU
VA22	Voltage of Analog22: AVDD22_AUADC
VIO18	Voltage of IO: VCC18IO; VCC18IO_EFUSE;
VA18	Voltage of Analog18: AVDD18_USB_P0; AVDD18_USB_P1; AVDD18_USB_P2; AVDD18_USB_P3; AVDD18_SSUSB_P0; AVDD18_SSUSB_P1; AVDD18_SSUSB_P2; AVDD18_SSUSB_P3; AVDD18_AP; AVDD18_PLLGP; AVDD18_CVBS; AVDD18_MIPIRX; AVDD18_MIPITX_CH1; AVDD18_MIPITX_CH2
VA105	Voltage of Analog: AVDD10_SSUSB_P2 AVDD10_SSUSB_P3
VA33	Voltage of Analog33: AVDD33_USB_P0; AVDD33_USB_P1; AVDD33_USB_P2; AVDD33_USB_P3; AVDD33_AUADC
VIO33	Voltage of IO33: VCC33IO_GPIO VCC33IO_I2SI02; VCC33IO_I2SI01; VCC33IO_I2SO02; VCC33IO_NOR; VCC33IO_SPICTP; VCC33IO_SPIMCU; VCC33IO_TDMD; VCC33IO_UART4B; VCC33IO_GBE;
VDRM	Voltage of DRAM: DDRV_ARDDR; DDRV_BRDDR; DDRV_CRDDR; DDRV_DRDDR;
VMC (Memory Card)	Voltage of MSDC: VCC33IO_MSDC1; Voltage of MSDC: VCC33IO_MSDC2;
XTAL	Clock input: XTALI; XTALO
RESETB	Reset Pin: RESET_

2.5 Analog IP Information

2.5.1 Introduction

To communicate with analog blocks, a common control interface for all analog blocks is implemented. In addition, there are some dedicated interfaces for data transfer. The common control interface translates the APB bus write and read cycle for specific addresses related to analog front-end control. In the write or read of any of these control registers, there is a latency associated with the transfer of data to or from the analog front-end. Dedicated data interface of each analog block is implemented in the corresponding digital block.

- Auxiliary ADC: Provides an ADC for the thermal sensor and other auxiliary analog functions monitoring.
- Clock generation: One clock-squarer for shaping the input sin-wave clock and 17 PLLs providing clock signals to base-band LVDSTx, VDEC, VENC, AUDIO, DSP, MCU, USB, MSDC units.

2.5.2 Analog Block Features

The analog blocks include the following analog functions:

- AUXADC
- Phase-locked loop (PLL)
- Temperature sensor
- AADC

2.5.3 AUXADC Block

2.5.3.1 Block Descriptions

The Auxiliary Analog-Digital Converter (AUXADC) includes the following functional blocks:

- Analog multiplexer: Selects signal from one of the auxiliary input channels. There are 16 input channels of AUXADC. Some are for internal voltage measurement and some for external voltage measurement. Environmental messages to be monitored, e.g. temperature, should be transferred to the voltage domain.
- 12-bit A/D converter: Converts the multiplexed input signal to 12-bit digital data.

See Table 2-15 for brief descriptions of AUXADC input channels.

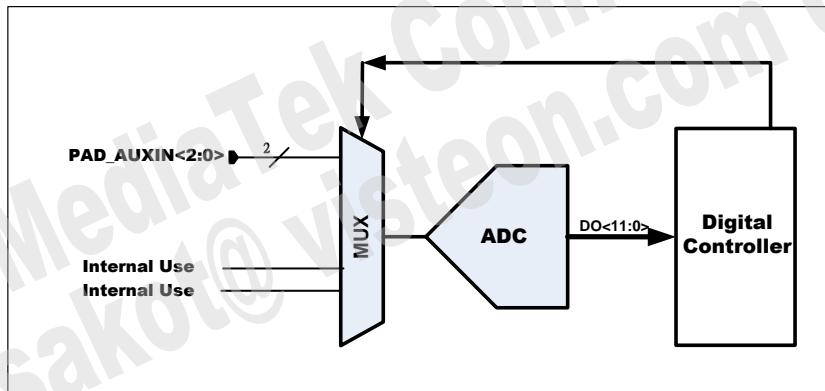


Figure 2-6 AUXADC Block Diagram

Table 2-15 Definitions of AUXADC Channels

AUXADC Channel ID	Description
Channel 0	External use (AUX_IN0)
Channel 1	External use (AUX_IN1)
Channel 2	NA
Channel 3	NA
Channel 4	NA
Channel 5	NA
Channel 6	NA
Channel 7	NA
Channel 8	NA
Channel 9	NA
Channel 10	Internal use
Channel 11	Internal use
Channel 12	External use (AUX_IN2)
Channel 13	NA
Channel 14	NA
Channel 15	NA

2.5.3.2 Auxiliary ADC Functional Specifications

See Table 2-16 for the functional specifications of AUXADC.

Table 2-16 AUXADC Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
N	Resolution	-	12	-	Bit
FC	Clock rate	-	26/8	-	MHz
FS	Sampling rate@N-Bit	-	FC/(N+4)	-	msps

Symbol	Parameter	Min.	Typ.	Max.	Unit
-	Input swing	0	-	1.5	V
RIN	Input resistance Unselected channel	- 1	-	-	MΩ
	Clock latency	-	N+4	-	1/FC
DNL	Differential nonlinearity	-	+1.0/-1.0	-	LSB
INL	Integral nonlinearity	-	+2.0/-2.0	-	LSB
SINAD	Signal to noise and distortion ratio (1 kHz full swing input & 1.0833 MHz clock rate)	56	62	-	dB
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption Power-up Power-down	-	- 480 1	-	μA μA

2.5.4 XTAL

2.5.4.1 XTAL Block Descriptions

The XTAL provides a reference clock source for the Phase-Locked Loop Group (PLLGP), and the XTAL's output clock waveform is sinusoidal with small amplitude (about several hundred mV).

2.5.4.2 XTAL Functional Specifications

See Table 2-17 for the functional specifications of XTAL.

Table 2-17 XTAL Specifications

Parameter	Min.	Typ.	Max.	Unit	Note
Power Consumption	-	2	-	mA	-
XTAL Frequency	-	26	-	MHz	-
Frequency tolerance	-	+/-10	-	ppm	EX xtal spec @25c
Frequency stability	-	+/-30	-	ppm	-40°C~85°C Ta
XTAL shunt capacitor	-	0.646329	-	pF	-
Load capacitor	-	12	-	pF	-
XTAL series resistor (Normal case)	-	-	40	Ohm	-

2.5.5 Phase-Locked Loop (PLL)

2.5.5.1 PLL Block Descriptions

There are a total of 17 PLLs in PLL macro that are separated into two groups, providing several clocks for CPU, BUS, MSDC and image-sensor.

PLLGP_TOP

MTK Confidential A

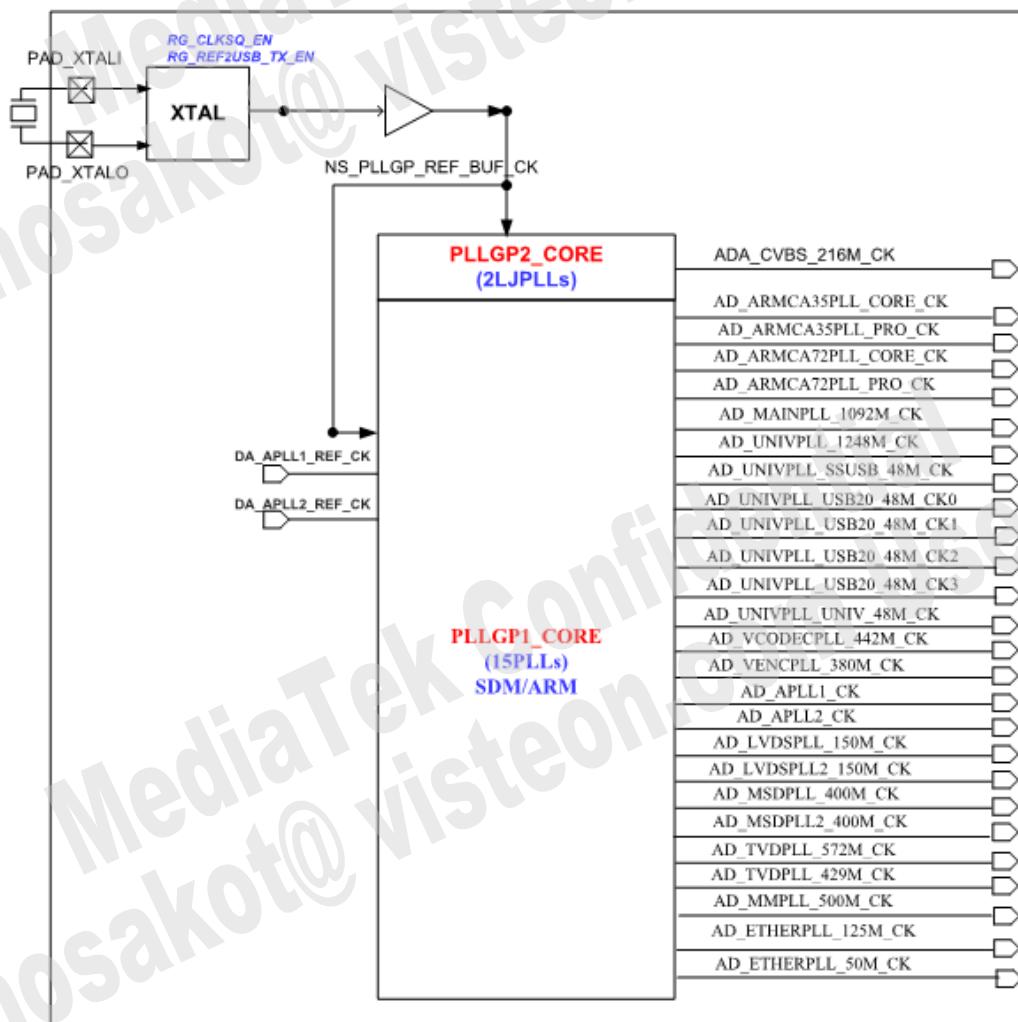


Figure 2-7 PLLGP Block Diagram

2.5.5.2 PLL Functional Specifications

See Table 2-18 to Table 2-34 for the functional specifications of PLL.

Table 2-18 ARMCA35PLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F _{in}	Input clock frequency	-	26	-	MHz
F _{out}	Output clock frequency	-	1200	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V

Symbol	Parameter	Min.	Typ.	Max.	Unit
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1.2	-	mA
-	Power-down current consumption	-	-	28	µA

Table 2-19 ARMCA72PLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	1600	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1.2	-	mA
-	Power-down current consumption	-	-	28	µA

Table 2-20 MAINPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	1092	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1.2	-	mA
-	Power-down current consumption	-	-	28	µA

Table 2-21 MMPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	500	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	µA

Table 2-22 UNIVPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	1248	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA

Symbol	Parameter	Min.	Typ.	Max.	Unit
-	Power-down current consumption	-	-	28	µA

Table 2-23 MSDCPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	400	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	µA

Table 2-24 MSDCPLL2 Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	400	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	µA

Table 2-25 VENCPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	380	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	µA

Table 2-26 VCODECPPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	442	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	µA

Table 2-27 TVDPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	572/429	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	μ A

Table 2-28 MMPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	500	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	μ A

Table 2-29 APLL1 Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	196.608	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	μ A

Table 2-30 APLL2 Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	180.6336	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	μ A

Table 2-31 LVDSPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{out}	Output clock frequency	-	150	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	μ A

Table 2-32 ETHERPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	150	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	μ A

Table 2-33 CVBSPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	216	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	μ A

Table 2-34 CVBSREFPLL Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{in}	Input clock frequency	-	26	-	MHz
F_{out}	Output clock frequency	-	702	-	MHz
DVDD	Digital power supply	0.95	1.0	1.05	V
AVDD	Analog power supply	1.71	1.8	1.89	V
-	Current consumption	-	1	-	mA
-	Power-down current consumption	-	-	28	μ A

2.5.6 Temperature Sensor

2.5.6.1 Block Descriptions

In order to monitor the temperature of CPUs, several temperature sensors are provided. The temperature sensor is made of substrate BJT in the CMOS process. The voltage output of temperature sensor is measured by AUXADC.

2.5.6.2 Functional Specifications

See Table 2-35 for the functional specifications of temperature sensor.

Table 2-35 Temperature Sensor Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
-	Resolution	-	0.3	-	°C
-	Accuracy@85°C	-3.5	-	3.5	°C
-	Accuracy@105°C	-2	-	2	°C
-	Accuracy@125°C	-0.5	-	0.5	°C
-	Active current	-	60	-	µA
-	Quiescent current	-	10	-	µA

2.5.7 AADC

2.5.7.1 Features

2-ch differential-ended AADC input, with 16-bit ADC data output.

2.5.7.2 Specification

Table 2-36 AADC Specification

MT2712 Audio Uplink	Specification				Note
	Min.	Typ.	Max.	Unit	
Analog Power Supply1	2.97	3.3	3.63	V	-
Analog Power Supply2	2.05	2.15	2.25	V	For ADC Core
Digital Power Supply	0.95	1	1.05	V	-
Temperature	-40	-	125	°C	-
Single Channel Current	-	2	-	mA	PGA+ADC+VREF
Resolution	-	16	-	Bit	-
Clock Frequency (FCK)	-	6.5	-	MHz	-
Input Signal Level Differential-ended Input Amplitude	-	0.4	-	Vrms	-
Total Harmonic Distortion + Noise (THD + N) Input Level:0.4 mVrms (PGA gain = 0 dB, Input signal frequency 1 kHz)	-	-25	-	dB	-

	Specification				
MT2712 Audio Uplink	Min.	Typ.	Max.	Unit	Note
Total Harmonic Distortion + Noise (THD + N) Input Level: 0.4 Vrms (PGA gain = 0 dB, Input signal frequency 1 kHz)	-	-82	-	dB	-
Signal-to-noise Ratio (Input differential signal Amplitude: 0.4 Vrms PGA gain = 0 dB, Unweighted, 20 Hz~20 kHz)	-	83	-	dB	Noise bandwidth is 20 Hz~20 kHz
Input PGA Gain Step	-	6	-	dB	-
Input PGA Gain	-6	0	24	dB	-
Channel Number	-	2	-	-	-
Input Impedance	13	20	27	Kohm	-
MICphone Biasing Voltage	1.9	-	2.2	V	-
Current draw from microphone bias	-	2	-	mA	-

2.5.7.3 Block Diagram

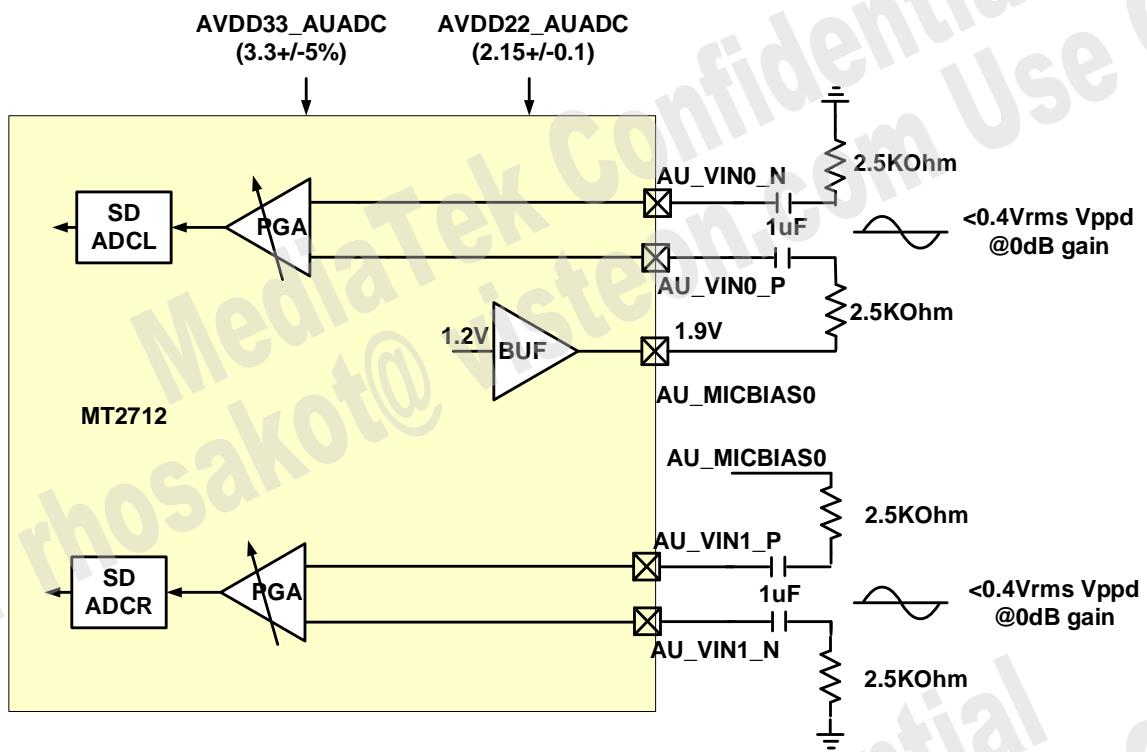


Figure 2-8 AADC Input Specification and Connection with Chip MT2712

2.5.8 CVBS

2.5.8.1 Introduction

CVBS supports the worldwide NTSC/PAL/SECAM video signal processing with 1channel analog front-end that integrates digital clamping, AGC loop operation, anti-aliasing filter, and one 10-bit ADC at single 1.8V power supply.

2.5.8.2 Features

1x single-ended CVBS input, 10-bit digital data output.

2.5.8.3 SPEC

1. Connector input max swing: 1.3Vpp
2. Matching impedance: 75 ohm, divider to 0.54x for 0.7Vpp swing AC couple input
3. AC Cap: 47nF
4. CVBS_COM: 1uF cap to ground
5. Power supply(AVDD18_CVBS): 1.8 V +/-5%

2.5.8.4 Block Diagram

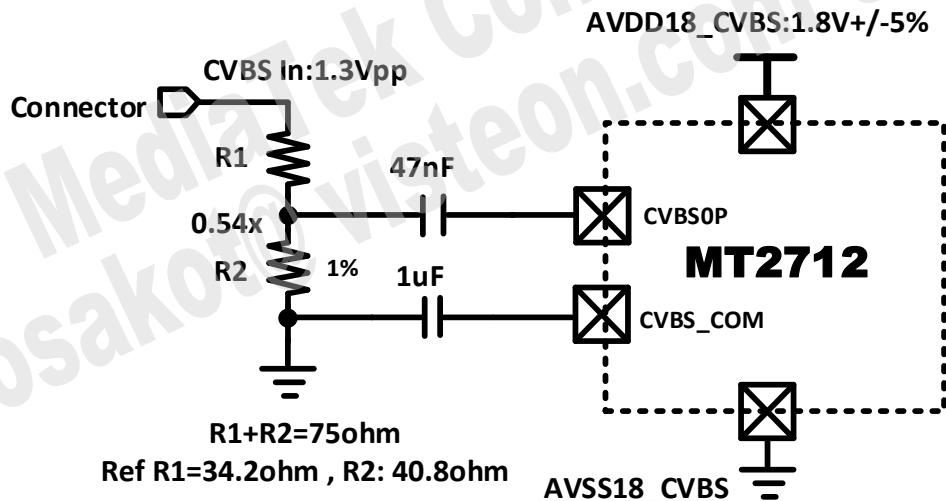
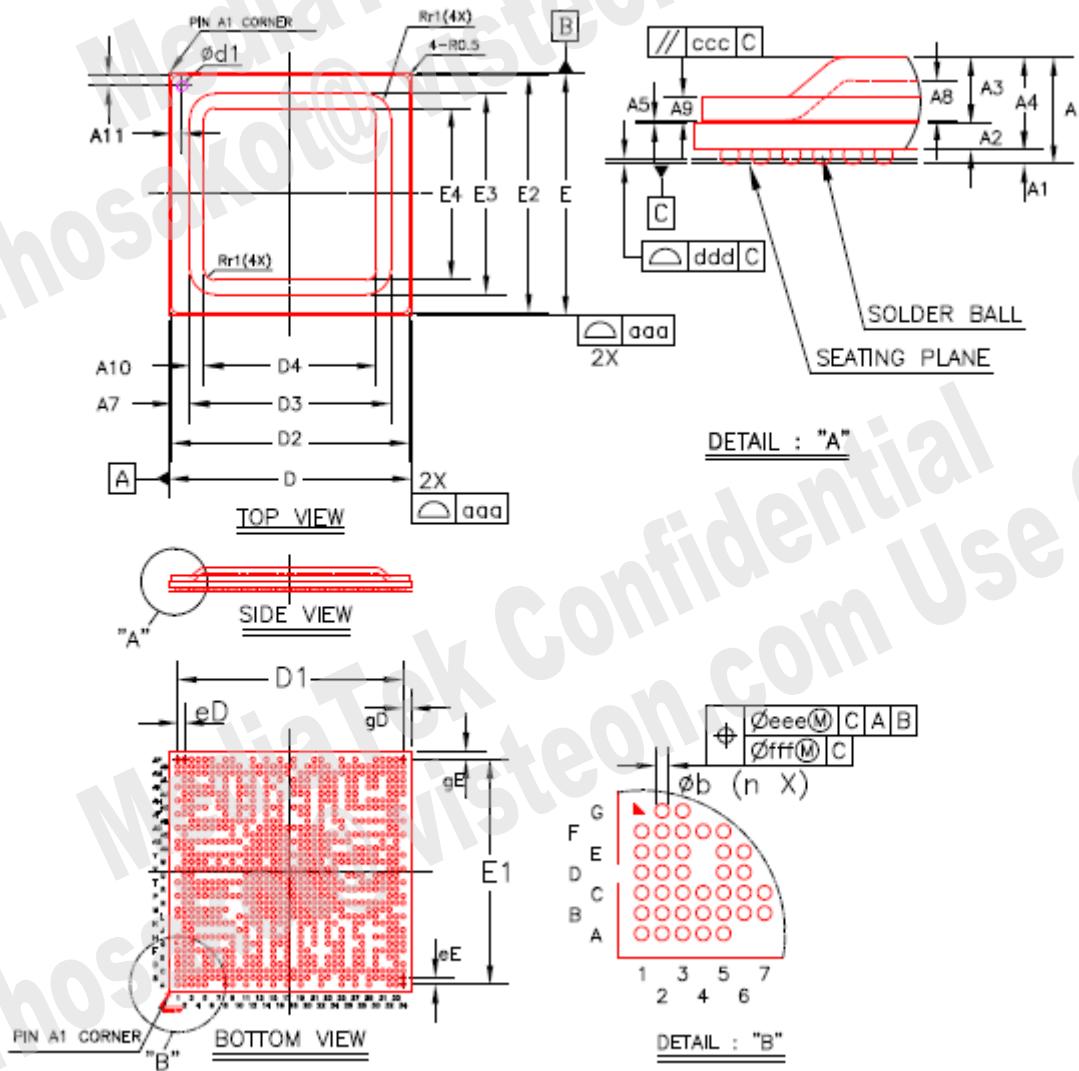


Figure 2-9 CVBS Application Reference Circuit

2.6 Package Information

2.6.1 Package Outlines



Item	Symbol	Common Dimensions		
		MIN.	NOM.	MAX.
Package Type		HFC FBGA		
Body Size	X	D	22.90	23.00
	Y	E	22.90	23.00
Ball Pitch	X	eD	0.65	
	Y	eE	0.65	
Substrate Thickness		A2	0.47	0.57
Adhesive Thickness		A5	0.05 Ref.	
H/S Height+Adhesive Thickness		A3	1.40 Ref.	
H/S Height+Adhesive+Substrate Thickness		A4	1.82	1.97
Total Thickness		A	2.13	2.29
Ball Diameter			0.42	
Ball Stand Off		A1	0.27	0.32
Ball Width		b	0.39	0.44
Package Edge Tolerance		aaa	0.10	
H/S Flatness		ccc	0.25	
Coplanarity		ddd	0.10	
Ball Offset (Package)		eee	0.25	
Ball Offset (Ball)		fff	0.10	
Ball Count		n	852	
Edge Ball Center to Center	X	D1	21.45	
	Y	E1	21.45	
Edge Ball Center to Package Edge	X	gD	0.775	
	Y	gE	0.775	

HEAT SPREADER(H/S) INFORMATION					
H/S Size	X	D2	22.55	22.60	22.65
	Y	E2	22.55	22.60	22.65
H/S Cavity Size	X	D3	18.60 Ref.		
	Y	E3	18.60 Ref.		
H/S Lid Size	X	D4	16.55	16.60	16.65
	Y	E4	16.55	16.60	16.65
H/S Width		A7	2.00 Ref.		
H/S Depth		A8	0.80	0.85	0.90
H/S Thickness		A9	0.45	0.50	0.55
H/S Oblique Width		A10	1.00 Ref.		
Pin A1 Through Hole Center to H/S Edge		A11	1.00 Ref.		
Pin A1 Through Hole Length		d1	0.75	1.00	1.25
Fillet Radius		r1	1.00 Ref.		

TITLE Azalea PACKAGE OUTLINE			
HFC FBGA 852 L 23X 23X 2.29mm			
DWG. NO.	REV.	SHEET	UNIT
MT-P00670	A	1 OF 2	MM

Figure 2-10 Outlines and Dimensions of HFC FBGA 23mm*23mm, 852-ball, 0.65mm Pitch Package

2.6.2 Thermal Operating Specifications

Table 2-37 Thermal Operating Specifications

Symbol	Description	Value	Unit
Tjunc	Max. operating junction temperature	125	°C
Theta-JA	Package thermal resistances of junction-to-ambient in natural convection	12.36	°C/Watt
Theta-JB	Package thermal resistance of junction-to-board	5.45	°C/Watt
Theta-JC	Package thermal resistances of junction-to- case	0.89	°C/Watt

Note: The device is mounted on MTK 6-Layer PCB and modeled per JEDEC condition.

2.6.3 Lead-free Packaging

The chip is provided in a lead-free package and meets RoHS requirements.

2.7 Ordering Information

Table 2-38 Scalable Platform Solution

Part #	Ordering Part #	Big Cores	Little Cores	Peak DMIPS	DRAM	GPU- ARM Mali T880	VIDEO Codec	Display Output
		A72	A35		LPDDR4			
MT2712S	MT2712Q/SOZC	2 x 1.5 GHz	4 x 1.2 GHz	22.6K	4-ch x 16b	MP4	FHD	x3
MT2712A	MT2712Q/AOZC	2 x 1.4 GHz	4 x 1.2 GHz	21.7K	4-ch x 16b	MP4	FHD	x3
MT2712C	MT2712Q/COZC	2 x 1.2 GHz	4 x 1.0 GHz	18.4K	4-ch x 16b	MP4	FHD	x3
MT2712H	MT2712Q/HOZC	1 x 1.4 GHz	4 x 1.2 GHz	15.1K	4-ch x 16b	MP4	FHD	x3
MT2712M	MT2712Q/MOZC	1 x 1.2 GHz	2 x 1.2 GHz	9.9K	4-ch x 16b	MP2	FHD	x3
MT2712E	MT2712Q/EOZC	1 x 1.0 GHz	2 x 0.8 GHz	7.5K	4-ch x 16b	MP2	FHD	x3

2.7.1 Top Marking Definition



MT2712Q: Part No.
DDDD: Date code, %%%%%%: internal code
@@@ -XX: Lot ID, -XX : Segment code
: internal code

Figure 2-11 Top Mark of MT2712

3 Clock and Power Control

3.1 Crystal Oscillator

3.1.1 Introduction

Crystal Oscillator (XO) provides a reference clock source for PLLGP group. Its output clock waveform is sinusoidal with small amplitude (about several hundred mV).

3.1.2 Features

See Table 3-1 for the XO functional specifications.

Table 3-1 XO Specifications

Parameter	Min.	Typ.	Max.	Unit	Notes
Nominal Frequency	-	26	-	MHz	-
Total Frequency Variation	-40	-	40	ppm	1. Frequency tolerance at 25°C 2. Frequency stability over temperature at -40°C~85°C 3. Aging variation: 10 years
Equivalent Series Resistance	-	-	50	Ohm	-
Load Capacitance	-	12	-	pF	-
Shunt Capacitance C0	-	-	3	pF	-
Operating Temperature	-40	-	85	°C	-

3.2 Application Processor Mixed Mode Control System

3.2.1 Introduction

Application Processor Mixed Mode Control System (Apmixedsys) generates PLL module control signals and controls Analog module working conditions. Apmixedsys configures registers to control PLL frequency via APB bus in normal mode or via cci pads in occmode.

3.2.2 Features

Apmixedsys receives the command from CPU or CCI to control PLL work:

- Control PLL output frequency
- Protect PLL setting from over high frequency

3.2.3 Block Diagram

The Apmixedsys block diagram is shown in Figure 3-1.

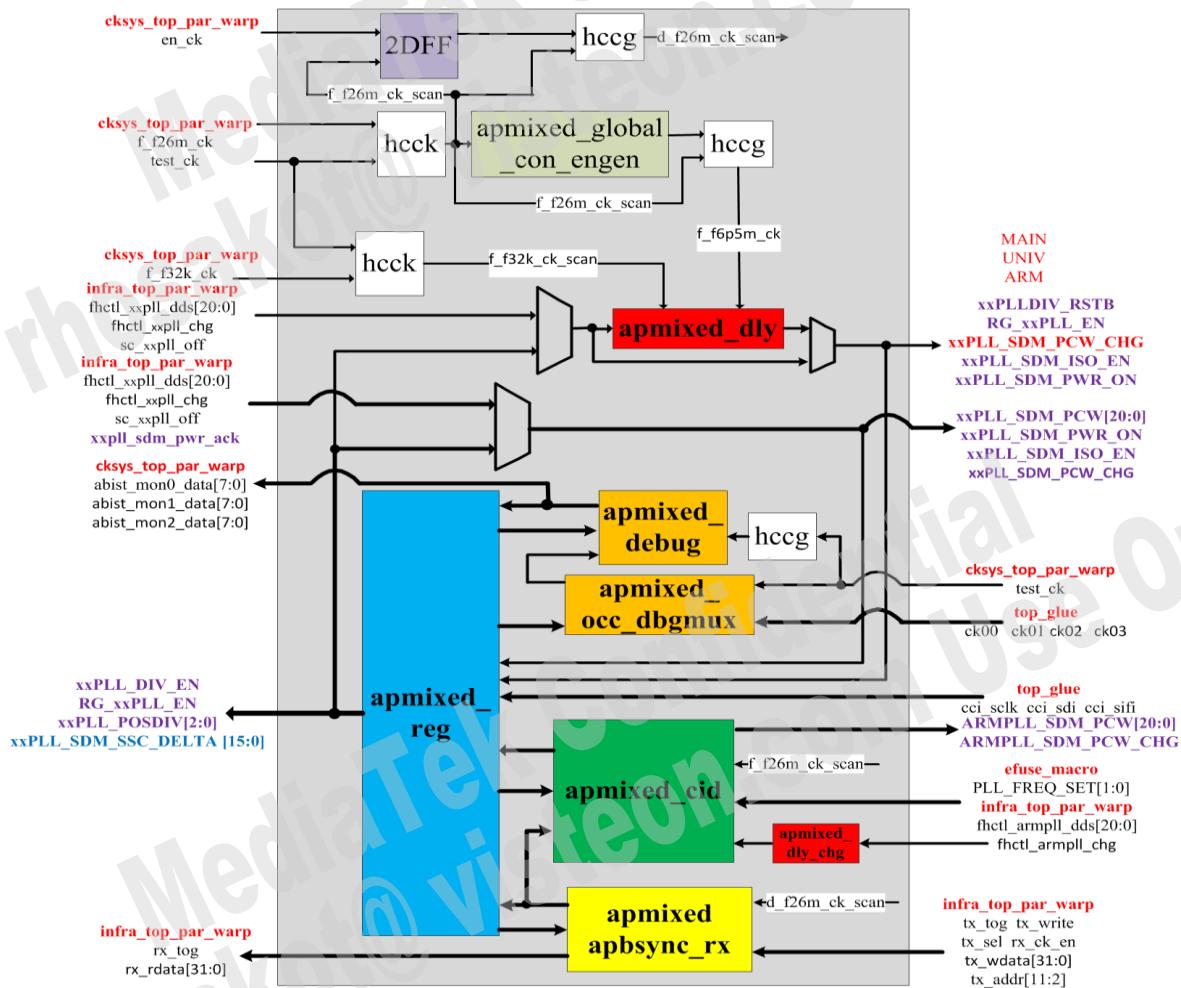


Figure 3-1 Apmixedsys Block Diagram

3.2.4 Register Definition

For register details, please refer to Chapter 1.2 of “MT2712 IVI Application Processor Registers”.

3.2.5 Programming Guide

3.2.5.1 PLL Frequency Calculation

$$CKOUT = freq * \left(1 + \frac{NCPO_PCW < 30:0>}{2^{24}}\right) * \frac{2^{DDS_PREDIV+2}}{2^{POSDIV}}$$

Frequency, the reference clock of PLL, is 26 MHz in MT2712. NCPO_PCW is xxPLL_SDM_PCW. POSDIV is xxPLL_POSDIV.

3.2.5.2 Software Control Sequence

General control sequence is as follows:

1. Enable PLL
2. Enable PWR_ON
3. Clear ISO_EN
4. Set all parameters first (SDM_PCW, POSDIV and so on)
5. Set PLL_EN = 1
6. Disable PLL
7. Set PLL_EN = 0
8. Set ISO_EN
9. Clear PWR_ON

Detailed control sequence:

- ARMCA35PLL Control Sequence
- ARMCA72PLL Control Sequence
- VENCPLL Control Sequence
- LVDSPLL Control Sequence
- LVDSPLL2 Control Sequence
- MAINPLL Control Sequence
- UNIVPLL Control Sequence
- VCODECPLL Control Sequence
- MMPLL Control Sequence
- APLL1 Control Sequence
- MSDCPLL Control Sequence
- MSDCPLL2 Control Sequence

ARMCA35PLL

Table 3-2 ARMCA35PLL Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μ s	-	-	-
SKIP	ARMCA35PLL setting	-	-	-
SYSTEM	DA_ARMCA35PLL_SDM_PWR_ON	1	10209110	00000003
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	DA_ARMCA35PLL_SDM_ISO_EN	0	10209110	00000001
SYSTEM	RG_ARMCA35PLL_SDM_FRA_EN	1	10209100	F0000110
SYSTEM	RG_ARMCA35PLL_SDM_PCW_CHG	1	10209104	BE000000

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SYSTEM	RG_ARMCA35PLL_SDM_PCW[30:0]	1011100010011101100010011101100	10209104	BE000000
SYSTEM	RG_ARMCA35PLL_POSDIV[2:0]	001	10209100	F0000110
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	RG_ARMCA35PLL_EN	1	10209100	F0000111
SKIP	Delay 20 μs	-	-	-

ARMCA72PLL**Table 3-3 ARMCA72PLL Control Sequence**

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μs	-	-	-
SKIP	ARMCA72PLL setting	-	-	-
SYSTEM	DA_ARMCA72PLL_SDM_PWR_ON	1	10209220	00000003
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	DA_ARMCA72PLL_SDM_ISO_EN	0	10209220	00000001
SYSTEM	RG_ARMCA72PLL_SDM_FRA_EN	1	10209210	00000100
SYSTEM	RG_ARMCA72PLL_SDM_PCW_CHG	1	10209214	A6800000
SYSTEM	RG_ARMCA72PLL_SDM_PCW[30:0]	0110101110110001001110110001010	10209214	A6800000
SYSTEM	RG_ARMCA72PLL_POSDIV[2:0]	000	10209210	00000100
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	RG_ARMCA72PLL_EN	1	10209210	00000101
SKIP	Delay 20 μs	-	-	-

VENCPLL**Table 3-4 VENCPLL Control Sequence**

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	delay 2000 μs	-	-	-
SKIP	VENCPLL setting	-	-	-
SYSTEM	DA_VENCPLL_SDM_PWR_ON	1	1020928C	00000003
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	DA_VENCPLL_SDM_ISO_EN	0	1020928C	00000001
SYSTEM	RG_VENCPLL_SDM_FRA_EN	1	10209280	00000120
SYSTEM	RG_VENCPLL_SDM_PCW_CHG	1	10209284	BA762762
SYSTEM	RG_VENCPLL_SDM_PCW[30:0]	0111010011101100010011101100010	10209284	BA762762
SYSTEM	RG_VENCPLL_POSDIV[2:0]	010	10209280	00000120
SKIP	Delay 0.1 μs	-	-	-

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SYSTEM	RG_VENCPLL_EN	1	10209280	00000121
SKIP	Delay 20 μ s	-	-	-

LVDSPLL**Table 3-5 LVDSPLL Control Sequence**

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μ s	-	-	-
SKIP	LVDSPLL setting	-	-	-
SYSTEM	DA_LVDSPLL_SDM_PWR_ON	1	1020937C	00000003
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	DA_LVDSPLL_SDM_ISO_EN	0	1020937C	00000001
SYSTEM	RG_LVDSPLL_SDM_FRA_EN	1	10209370	00000130
SYSTEM	RG_LVDSPLL_SDM_PCW_CHG	1	10209374	AE276276
SYSTEM	RG_LVDSPLL_SDM_PCW[30:0]	0101110001001110110001001110110	10209374	AE276276
SYSTEM	RG_LVDSPLL_POSDIV[2:0]	011	10209370	00000130
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	RG_LVDSPLL_EN	1	10209370	00000131
SKIP	Delay 20 μ s	-	-	-

LVDSPLL2**Table 3-6 LVDSPLL2 Control Sequence**

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μ s	-	-	-
SKIP	LVDSPLL2 setting	-	-	-
SYSTEM	DA_LVDSPLL2_SDM_PWR_ON	1	1020939C	00000003
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	DA_LVDSPLL2_SDM_ISO_EN	0	1020939C	00000001
SYSTEM	RG_LVDSPLL2_SDM_FRA_EN	1	10209390	00000130
SYSTEM	RG_LVDSPLL2_SDM_PCW_CHG	1	10209394	AE276276
SYSTEM	RG_LVDSPLL2_SDM_PCW[30:0]	0101110001001110110001001110110	10209394	AE276276
SYSTEM	RG_LVDSPLL2_POSDIV[2:0]	011	10209390	00000130
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	RG_LVDSPLL2_EN	1	10209390	00000131
SKIP	Delay 20 μ s	-	-	-

MAINPLL

Table 3-7 MAINPLL Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 µs	-	-	-
SKIP	MAINPLL setting	-	-	-
SYSTEM	DA_MAINPLL_SDM_PWR_ON	1	1020923C	00000003
SKIP	Delay 0.1 µs	-	-	-
SYSTEM	DA_MAINPLL_SDM_ISO_EN	0	1020923C	00000001
SYSTEM	RG_MAINPLL_SDM_FRA_EN	1	10209230	F0000110
SYSTEM	RG_MAINPLL_SDM_PCW_CHG	1	10209234	D4000000
SYSTEM	RG_MAINPLL_SDM_PCW[30:0]	10101000000000000000000000000000	10209234	D4000000
SYSTEM	RG_MAINPLL_POSDIV[2:0]	001	10209230	F0000110
SKIP	delay 0.1 µs	-	-	-
SYSTEM	RG_MAINPLL_EN	1	10209230	F0000111
SKIP	Delay 20 µs	-	-	-

UNIVPLL

Table 3-8 UNIVPLL Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 µs	-	-	-
SKIP	UNIVPLL setting	-	-	-
SYSTEM	DA_UNIVPLL_SDM_PWR_ON	1	1020924C	00000003
SKIP	Delay 0.1 µs	-	-	-
SYSTEM	DA_UNIVPLL_SDM_ISO_EN	0	1020924C	00000001
SYSTEM	RG_UNIVPLL_SDM_FRA_EN	1	10209240	FE000110
SYSTEM	RG_UNIVPLL_SDM_PCW_CHG	1	10209244	E0000000
SYSTEM	RG_UNIVPLL_SDM_PCW[30:0]	11000000000000000000000000000000	10209244	E0000000
SYSTEM	RG_UNIVPLL_POSDIV[2:0]	001	10209240	FE000110
SKIP	Delay 0.1 µs	-	-	-
SYSTEM	RG_UNIVPLL_EN	1	10209240	FE000111
SKIP	DIV26 enable	-	-	-
SYSTEM	RG_USB20_MON48M_EN	1	10209904	FFFFFD7
SYSTEM	RG_UNIV_48M_EN	1	10209240	FE000111
SKIP	Delay 20 µs	-	-	-

VCODECPLL

Table 3-9 VCODECPLL Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μs	-	-	-
SKIP	VCODECPLL setting	-	-	-
SYSTEM	DA_VCODECPLL_SDM_PWR_ON	1	1020932C	00000003
SKIP	delay 0.1 μs	-	-	-
SYSTEM	DA_VCODECPLL_SDM_ISO_EN	0	1020932C	00000001
SYSTEM	RG_VCODECPLL_SDM_FRA_EN	1	10209320	C0000120
SYSTEM	RG_VCODECPLL_SDM_PCW_CHG	1	10209324	C4000000
SYSTEM	RG_VCODECPLL_SDM_PCW[30:0]	10001000000000000000000000000000	10209324	C4000000
SYSTEM	RG_VCODECPLL_POSDIV[2:0]	010	10209320	C0000120
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	RG_VCODECPLL_EN	1	10209320	C0000121
SKIP	Delay 20 μs	-	-	-

MMPLL

Table 3-10 MMPLL Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μs	-	-	-
SKIP	MMPLL setting	-	-	-
SYSTEM	DA_MMPLL_SDM_PWR_ON	1	10209260	00000003
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	DA_MMPLL_SDM_ISO_EN	0	10209260	00000001
SYSTEM	RG_MMPLL_SDM_FRA_EN	1	10209250	00000120
SYSTEM	RG_MMPLL_SDM_PCW_CHG	1	10209254	CCEC4EC5
SYSTEM	RG_MMPLL_SDM_PCW[30:0]	1001100111011000100111011000101	10209254	CCEC4EC5
SYSTEM	RG_MMPLL_POSDIV[2:0]	010	10209250	00000120
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	RG_MMPLL_EN	1	10209250	00000121
SKIP	Delay 20 μs	-	-	-

APLL1**Table 3-11 APLL1 Control Sequence**

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μ s	-	-	-
SKIP	APLL1 setting	-	-	-
SYSTEM	DA_APOLL1_SDM_PWR_ON	1	10209340	00000003
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	DA_APOLL1_SDM_ISO_EN	0	10209340	00000001
SYSTEM	RG_APOLL1_SDM_FRA_EN	1	10209330	00000130
SYSTEM	RG_APOLL1_SDM_PCW_CHG	1	10209334	BC7EA932
SYSTEM	RG_APOLL1_SDM_PCW[30:0]	011110001111101010100100110010	10209334	BC7EA932
SYSTEM	RG_APOLL1_POSDIV[2:0]	011	10209330	00000130
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	RG_APOLL1_EN	1	10209330	00000131
SKIP	Delay 20 μ s	-	-	-

APLL2**Table 3-12 APLL2 Control Sequence**

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μ s	-	-	-
SKIP	APLL2 setting	-	-	-
SYSTEM	DA_APOLL2_SDM_PWR_ON	1	10209360	00000003
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	DA_APOLL2_SDM_ISO_EN	0	10209360	00000001
SYSTEM	RG_APOLL2_SDM_FRA_EN	1	10209350	00000130
SYSTEM	RG_APOLL2_SDM_PCW_CHG	1	10209354	B7945EA6
SYSTEM	RG_APOLL2_SDM_PCW[30:0]	0110111100101000101111010100110	10209354	B7945EA6
SYSTEM	RG_APOLL2_POSDIV[2:0]	011	10209350	00000130
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	RG_APOLL2_EN	1	10209350	00000131
SKIP	Delay 20 μ s	-	-	-

MSDCPLL

Table 3-13 MSDCPLL Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μs	-	-	-
SKIP	MSDCPLL setting	-	-	-
SYSTEM	DA_MSDCPLL_SDM_PWR_ON	1	1020927C	00000003
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	DA_MSDCPLL_SDM_ISO_EN	0	1020927C	00000001
SYSTEM	RG_MSDCPLL_SDM_FRA_EN	1	10209270	00000120
SYSTEM	RG_MSDCPLL_SDM_PCW_CHG	1	10209274	BD89D89E
SYSTEM	RG_MSDCPLL_SDM_PCW[30:0]	0111101100010011101100010011110	10209274	BD89D89E
SYSTEM	RG_MSDCPLL_POSDIV[2:0]	010	10209270	00000120
SKIP	delay 0.1 μs	-	-	-
SYSTEM	RG_MSDCPLL_EN	1	10209270	00000121
SKIP	Delay 20 μs	-	-	-

MSDCPLL2

Table 3-14 MSDCPLL2 Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μs	-	-	-
SKIP	MSDCPLL2 setting	-	-	-
SYSTEM	DA_MSDCPLL2_SDM_PWR_ON	1	1020941C	00000003
SKIP	Delay 0.1 μs	-	-	-
SYSTEM	DA_MSDCPLL2_SDM_ISO_EN	0	1020941C	00000001
SYSTEM	RG_MSDCPLL2_SDM_FRA_EN	1	10209410	00000120
SYSTEM	RG_MSDCPLL2_SDM_PCW_CHG	1	10209414	BD89D89E
SYSTEM	RG_MSDCPLL2_SDM_PCW[30:0]	0111101100010011101100010011110	10209414	BD89D89E
SYSTEM	RG_MSDCPLL2_POSDIV[2:0]	010	10209410	00000120
SKIP	delay 0.1 μs	-	-	-
SYSTEM	RG_MSDCPLL2_EN	1	10209410	00000121
SKIP	Delay 20 μs	-	-	-

ETHERPLL

Table 3-15 ETHERPLL Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	delay 2000 μ s	-	-	-
SKIP	ETHERPLL setting	-	-	-
SYSTEM	DA_ETHERPLL_SDM_PWR_ON	1	1020930C	00000003
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	DA_ETHERPLL_SDM_ISO_EN	0	1020930C	00000001
SYSTEM	RG_ETHERPLL_SDM_FRA_EN	1	10209300	C0000120
SYSTEM	RG_ETHERPLL_SDM_PCW_CHG	1	10209304	CCEC4EC5
SYSTEM	RG_ETHERPLL_SDM_PCW[30:0]	10011001110110001001110110001 01	10209304	CCEC4EC5
SYSTEM	RG_ETHERPLL_POSDIV[2:0]	010	10209300	C0000120
SKIP	Delay 0.1 μ s	-	-	-
SYSTEM	RG_ETHERPLL_EN	1	10209300	C0000121
SKIP	Delay 20 μ s	-	-	-

CVBSPLL

Table 3-16 CVBSPLL Control Sequence

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μ s	-	-	-
SKIP	BG current enable	-	-	-
SYSTEM	RG_BGR_PI_PWD	0	10209600	00000000
SKIP	CVBSREFPLL setting	-	-	-
SYSTEM	RG_CVBSREFPLL_EN	1	10209318	B50113C9
SYSTEM	RG_CVBSREFPLL_BIAS_EN	1	1020931C	90800220
SYSTEM	RG_CVBSREFPLL_FBKDIV[6:0]	110101	10209318	B50113C9
SYSTEM	RG_CVBSREFPLL_FBKSEL[1:0]	00	10209318	B50113C9
SYSTEM	RG_CVBSREFPLL_POSDIV[1:0]	01	10209318	B50113C9
SYSTEM	RG_CVBSREFPLL_MONCK_EN	0	1020931C	90800220
SKIP	LDO enable	-	-	-
SYSTEM	RG_PLLGP_RESERVE[15]	1	10209040	FF002000
SKIP	CVBSREFPLL_DIV26 setting	-	-	-
SYSTEM	RG_CVBSREFPLL_DIV26_EN	1	1020931C	90800220
SKIP	Delay 10 μ s	-	-	-
SKIP	CVBSPLL setting	-	-	-
SYSTEM	RG_CVBSPLL_EN	1	10209310	BF0A13C9
SYSTEM	RG_CVBSPLL_BIAS_EN	1	10209314	01020005

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CVBSPLL_FBKDIV[6:0]	111111	10209310	BF0A13C9
SYSTEM	RG_CVBSPLL_FBKSEL[1:0]	00	10209310	BF0A13C9
SYSTEM	RG_CVBSPLL_POSDIV[1:0]	10	10209310	BF0A13C9
SYSTEM	RG_CVBSPLL_MONCK_EN	1	10209314	01020025
SKIP	DIV2 setting	-	-	-
SYSTEM	RG_CVBSPLL_DIV2_EN	1	10209314	01020025
SKIP	Delay 20 μ s	-	-	-

CVBSREFPLL**Table 3-17 CVBSREFPLL Control Sequence**

Options	REG_Name	REG_Value (BIN, e.g. 01110)	Address	Value
SKIP	XTAL setting	-	-	-
SYSTEM	RG_CLKSQ_EN	1	10209000	0001A171
SKIP	Delay 2000 μ s	-	-	-
SKIP	BG current enable	-	-	-
SYSTEM	RG_BGR_PI_PWD	0	10209600	00000000
SKIP	CVBSREFPLL setting	-	-	-
SYSTEM	RG_CVBSREFPLL_EN	1	10209318	B50113C9
SYSTEM	RG_CVBSREFPLL_BIAS_EN	1	1020931C	90800220
SYSTEM	RG_CVBSREFPLL_FBKDIV[6:0]	110101	10209318	B50113C9
SYSTEM	RG_CVBSREFPLL_FBKSEL[1:0]	00	10209318	B50113C9
SYSTEM	RG_CVBSREFPLL_POSDIV[1:0]	01	10209318	B50113C9
SYSTEM	RG_CVBSREFPLL_MONCK_EN	1	1020931C	90801220
SKIP	Delay 20 μ s	-	-	-

3.3 Top Clock Generator**3.3.1 Introduction**

This chapter introduces the Top Clock Generator (TOPCKGEN) and the clock architecture.

3.3.2 Features

TOPCKGEN is responsible for generating the following clock signals:

- Free clock generation for the whole chip
- Infrastructure and peripheral system clock, including the top level AXI fabric clock
- Multimedia system clock
- Pad macro clocks to be synchronized with one of the above systems

The module TOPCKGEN provides clock source selection. Each clock has several clock source selections and can be turned off. When switching certain clock from frequency A to frequency B, make sure frequency A and B are available.

It comprises glitch-free clock MUX and digital clock divider to generate various clock frequencies.

3.3.3 Block Diagram

3.3.3.1 Clock Architecture

There are clock generators not only in the top level hierarchy, but also in every partition/system. Figure 3-2 shows the location of the top level clock generator.

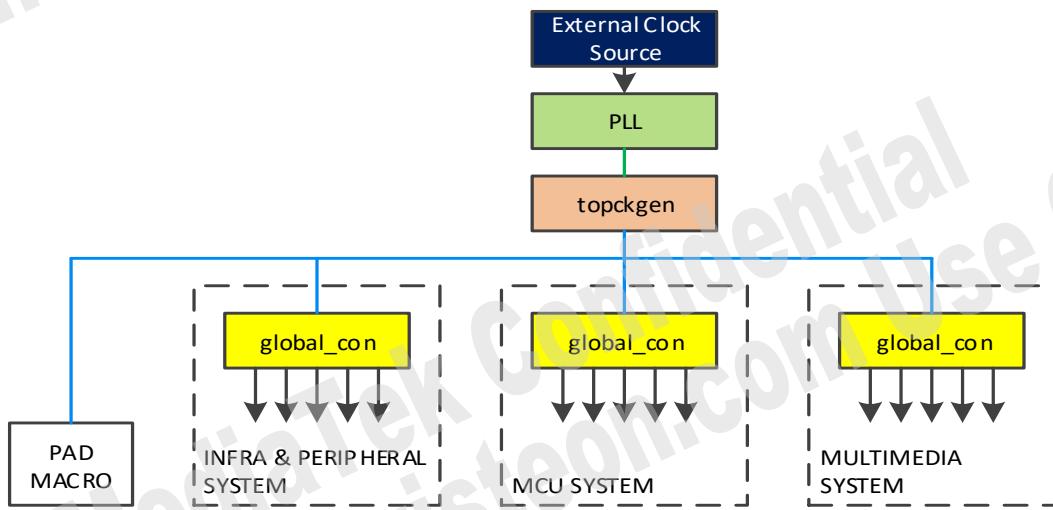


Figure 3-2 Clock Architecture Block Diagram

3.3.3.2 Clock Multiplexer

Clock selection and generation have a similar structure (see Figure 3-3). Several clock sources are provided. Users can choose one by specified register setting. The turn-off bit is provided as well to stop the clock output.

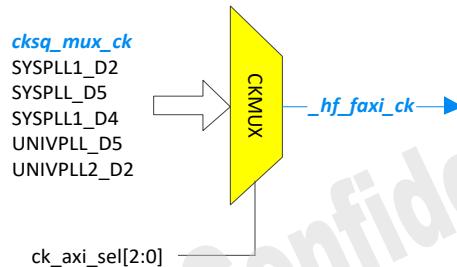


Figure 3-3 Example of Clock Multiplexer

3.3.3.3 Clock PLL

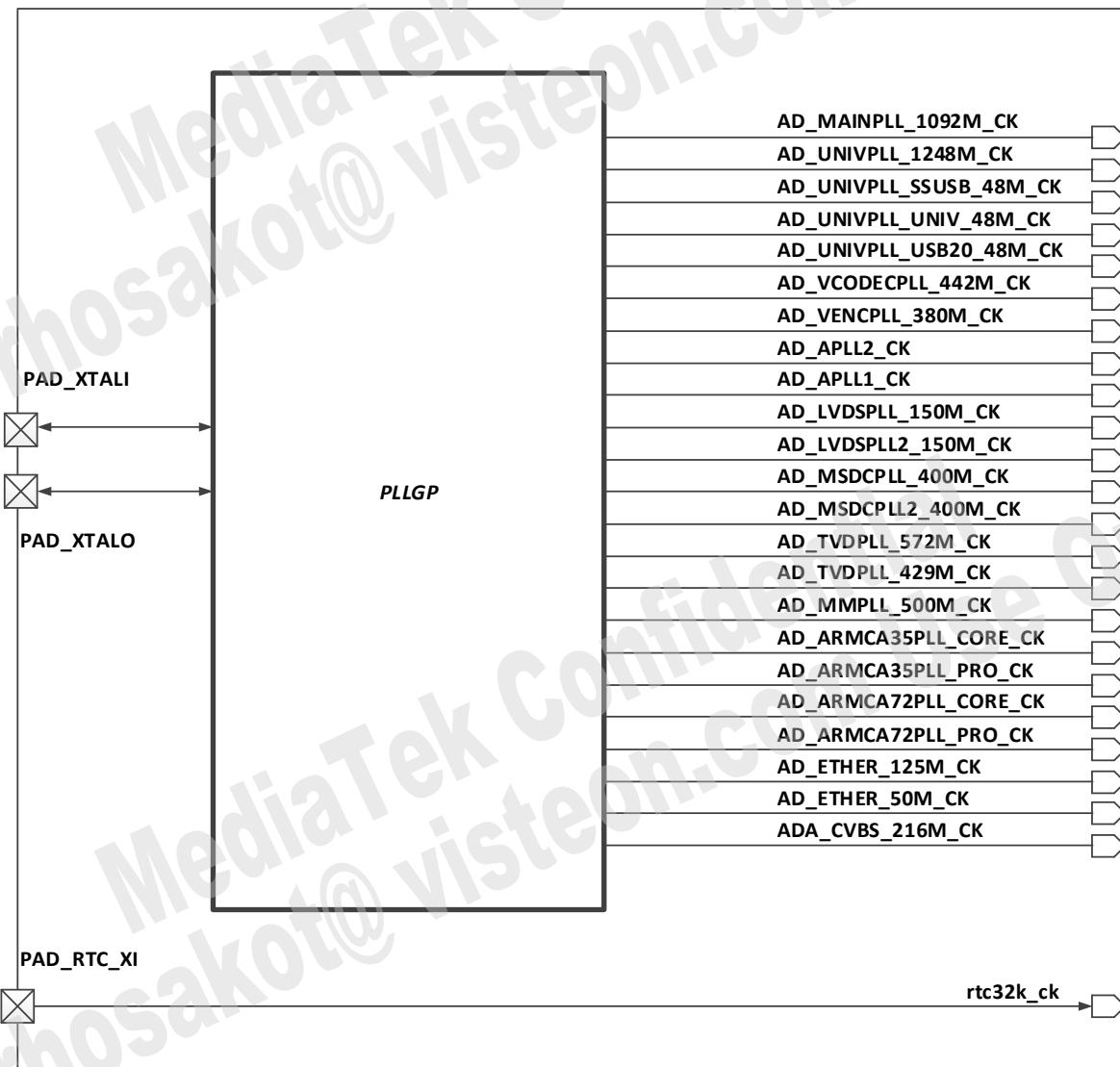


Figure 3-4 PLLGP Block Diagram

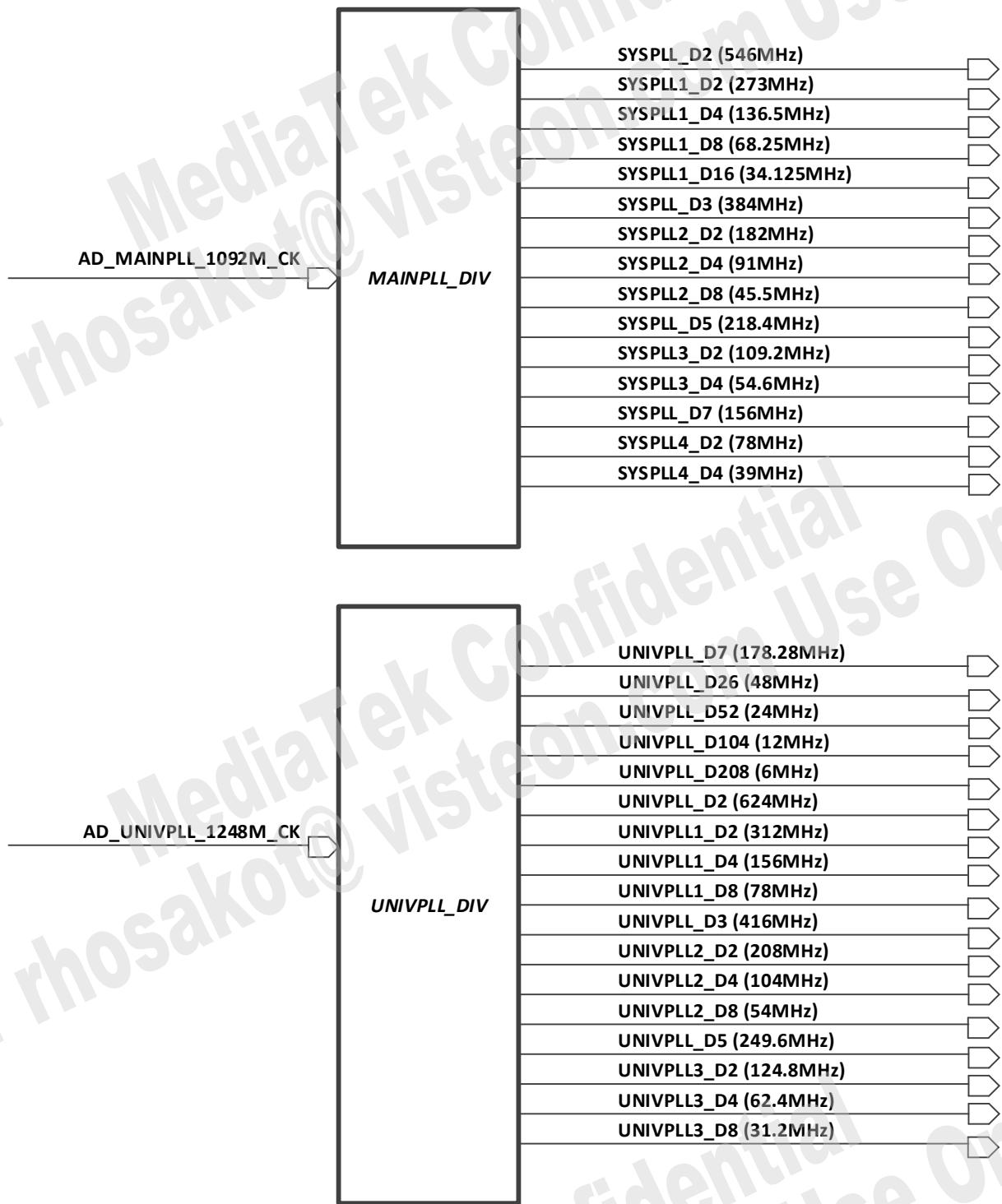


Figure 3-5 PLL Divider1 Block Diagram

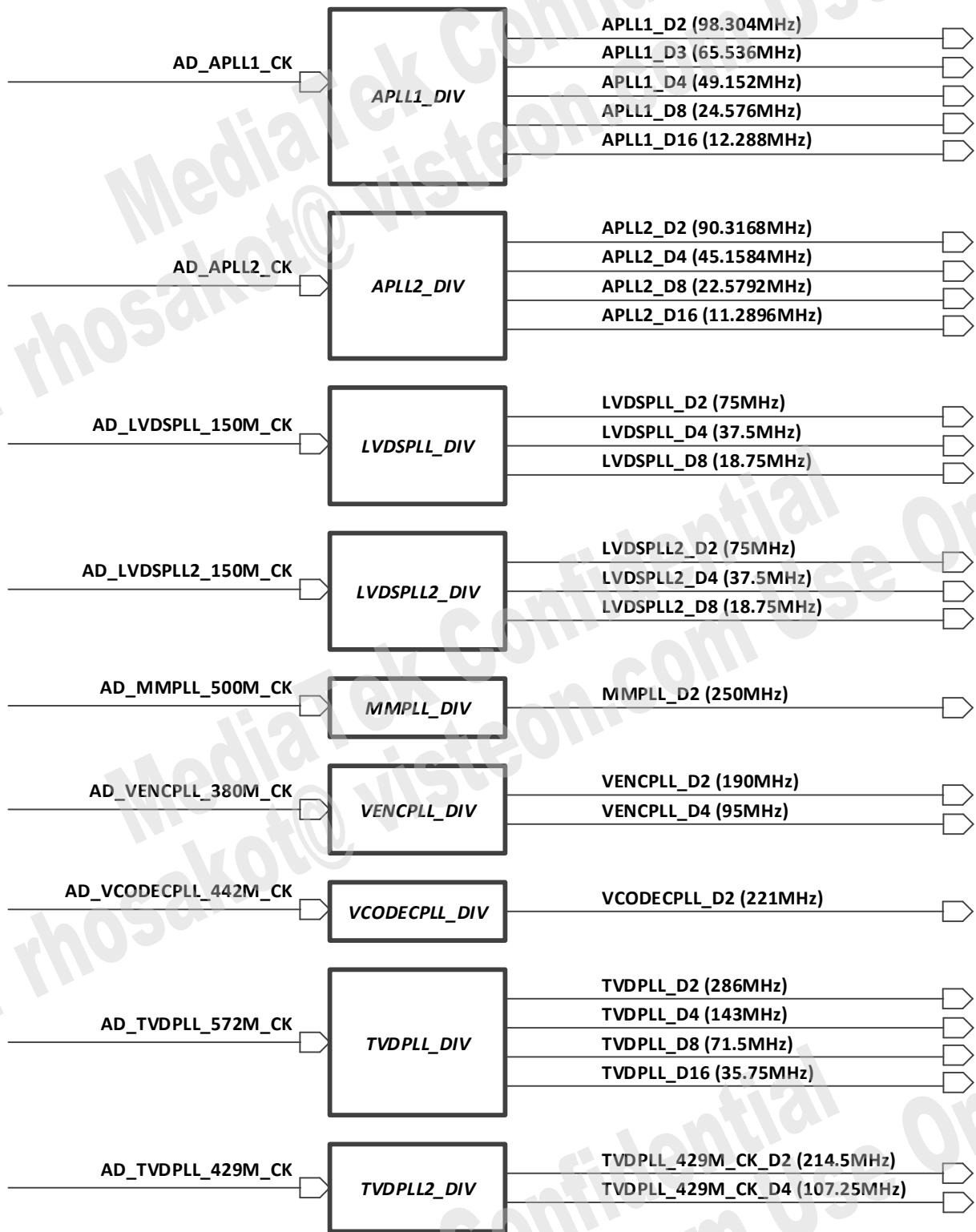


Figure 3-6 PLL Divider2 Block Diagram

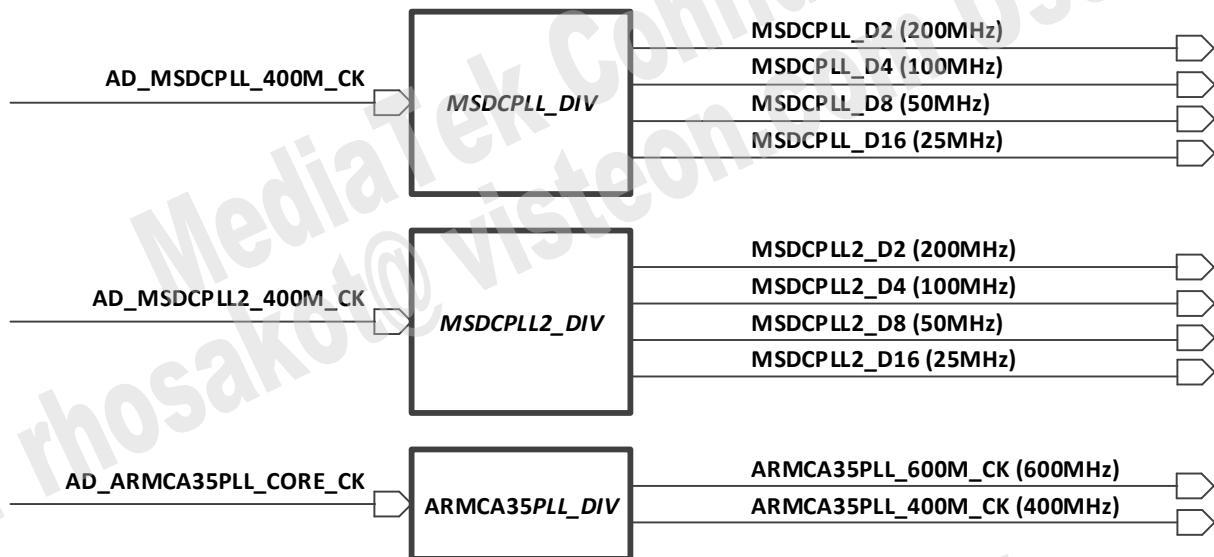


Figure 3-7 PLL Divider3 Block Diagram

3.3.3.4 PLL Related Control

All PLLs inside the application system are listed in Table 3-18.

The enabling of PLL can be switched between software control and hardware control. The hardware control is from SCPSYS.

The hopping and SSC features can be switched between software control and hardware control. The hardware control is from FHCTL.

Table 3-18 PLL Related Control

PLL	Capability	Controlled by FHCTL	Controlled by SPM
ARMCA35PLL	Hopping, SSC	Y	Y
ARMCA72PLL	Hopping, SSC	Y	Y
MAINPLL	Hopping, SSC	Y	Y
MMPPLL	Hopping, SSC	Y	N
VENCPPLL	Hopping, SSC	Y	N
UNIVPLL	Fix	N	Y
MSDCPLL	Hopping, SSC	Y	N
MSDCPLL2	Hopping, SSC	Y	N
TVDPLL	Hopping, SSC	Y	N
VCODECPPLL	Hopping, SSC	Y	N
APLL1	Fix	N	N

PLL	Capability	Controlled by FHCTL	Controlled by SPM
APLL2	Fix	N	N
MIPI	SSC	N	N
USB20_PHYA	Fix	N	N
SSUSB_PHYA	Fix	N	N
LVDSPLL	SSC	Y	N
LVDSPLL2	SSC	Y	N
ETHERPLL	Fix	N	N
CVBSPLL	Fix	N	N

3.3.3.5 Clock Gating

The clock gating for module TOPCKGEN is listed in Table 3-19, where DCM and turn-off settings are also provided.

Table 3-19 Clock Gating Settings

Register Name	Bit	Default	Function Name	Description
DCM_CFG	[7]	1'b0	dcm_enable	Enables hf_faxi_ck DCM
CLK_SCP_CFG_0	[0]	1'b0	sc_26ck_off_en	Turns on scpsys control path to gate 26 MHz
	[2]	1'b0	sc_axick_off_en	Turns on scpsys control path to gate hf_faxi_ck
	[9]	1'b0	sc_mac_26m_off_en	Turns on scpsys control path to gate MIPI 26 MHz
CLK_SCP_CFG_1	[0]	1'b0	sc_axi_26m_sel_en	Turns on scpsys control path to switch hf_faxi_ck to 26 MHz
	[4]	1'b0	sc_axick_dcm_dis_en	Turns on scpsys control path to disable DCM of hf_faxi_ck

3.3.3.6 Frequency Meter

There are two frequency meters inside TOPCKGEN. One, called abist_fmter, is for PLLs and TEST clock. The other, called ckggen_fmter, is for clocks generated from TOPCKGEN.

Both structures have PAD output that can observe frequency directly instead of reading results from the frequency meter. abist_fmter is output to DEBUG_MON[0] and ckggen_fmter is output to DEBUG_MON[2].

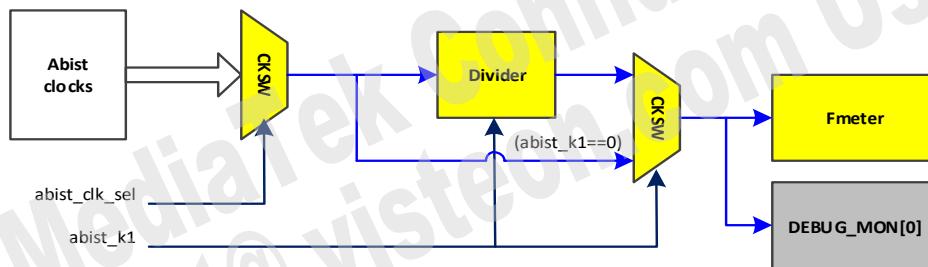


Figure 3-8 ABIST FMETER Structure

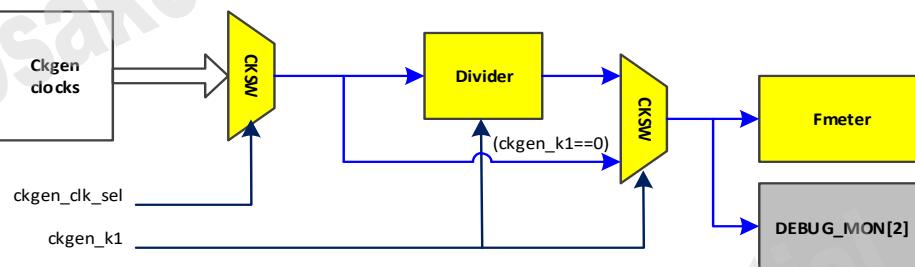


Figure 3-9 CKGEN Structure

3.3.4 Register Definition

For register details, please refer to Chapter 1.3 of “MT2712 IVI Application Processor Registers”.

3.3.5 Programming Guide

3.3.5.1 Clock Off

- The clock can be turned on/off by changing the value of `pdn_*`. However, this control cannot be switched along with `clk_*_sel` and `clk_*_inv`.
- It is recommended to change `pdn_*` with SET and CLEAR function provided by `CLK_CFG_*_SET` and `CLK_CFG_*_CLR`, because there may be clock with multi-bit `pdn_*` which is planned to prevent read-modify-write from different subsystems at the same time.
- SET and CLEAR function of `CLK_CFG_*` is a solution to prevent read-modify-write from different subsystems.

3.3.5.2 Clock Switching

- Make sure clock A and clock B are available before changing the setting of `clk_*_sel`. If switched to a non-existent clock, the clock switch will be stuck until the non-existent clock is turned on to release the clock switch.
- Support multi-clock switching at the same time (without changing `pdn_*`).

Switching from clock A to clock B

1. Make sure clock B is ready.
2. Change clk_*_sel.
3. Wait until chg_sta = 1'b0 (optional).
4. Turn off clock A (optional).

3.3.5.3 Switch AXI to 26 MHz by SCPSYS

The reflection time is about 17T 26 MHz, if all clocks are counted as 26 MHz. Refer to the following formula:

$$4T \text{ Bus Clock}_{\text{*1}} + 4T \text{ Current Clock}_{\text{*2}} + 5T \text{ Reference Clock}_{\text{*3}} + 1T \text{ Bus Clock}_{\text{*4}} + 3T \text{ Target Clock}_{\text{*5}}$$

Table 3-20 Clock Delay Comment

Comment	Description
Bus clock	26 MHz (in current project)
Ref clock	26 MHz, balance with bus clock
*1	2T sync + 1T control
*2	3T sync
*3	4T sync
*4	1T control. For async CLKSW like hf_fmem_ck used, it will be 2T sync.
*5	2T sync

3.3.5.3.1 Frequency Meter

There are two frequency meters embedded inside TOPCKGEN.

1. Set fmeter_en to 1'b1.
2. Choose target clock by changing abist_clk_sel/ckgen_clk_sel.
3. Change abist_k1/ckgen_k1 for dividing target clock (optional).
4. Change reference clock by changing abist_clk_exc/ckgen_clk_exc (optional).
5. Change abist_load_cnt/ckgen_load_cnt (optional).
6. Trigger frequency meter by setting abist_tri_cal/ckgen_tri_cal = 1b'1.
7. Wait until abist_tri_cal/ckgen_tri_cal = 1'b0.
8. Read frequency meter result from abist_cal_cnt/ckgen_cal_cnt.
 - freq(target) = (abist_k1 + 1)*[freq(reference clock) * abistcal_cnt]/(abist_load_cnt + 1)
 - freq(target) = (ckgen_k1 + 1)*[freq(reference clock) * ckgen_cal_cnt]/(ckgen_load_cnt + 1)

3.3.5.4 Clock Mux Table

Table 3-21 Clock Mux

Clock Name	Provided Clock Source
hf_faxi_ck	CLK26M
	SYSPLL1_D2
	SYSPLL_D5
	SYSPLL1_D4
	UNIVPLL_D5
	UNIVPLL2_D2
	MSDCPLL2_CK
hf_fmem_ck	CLK26M
	DMPLL_CK
hf_fddrphycfg_ck	CLK26M
	SYSPLL1_D8
hf_fmm_ck	CLK26M
	VENCPLL_CK
	SYSPLL_D3
	SYSPLL1_D2
	SYSPLL_D5
	SYSPLL1_D4
	UNIVPLL1_D2
f_fpwm_ck	UNIVPLL2_D2
	UNIVPLL1_D4
	UNIVPLL3_D2
	UNIVPLL2_D4
hf_fvdec_ck	CLK26M
	VCODECPLL_CK
	TVDPLL_429M_CK
	UNIVPLL_D3

Clock Name	Provided Clock Source
hf_fvenc_ck	VENCPLL_CK
	SYSPLL_D3
	UNIVPLL1_D2
	MMPLL_D2
	SYSPLL3_D2
	TVDPLL_CK
hf_fmfq_ck	CLK26M
	UNIVPLL1_D2
	MMPLL_D2
	TVDPLL_D2
	SYSPLL1_D2
	UNIVPLL_D5
	VCODECPPLL_D2
	UNIVPLL2_D2
	SYSPLL3_D2
	CLK26M
	MMPLL_CK
	UNIVPLL_D3
	CLK26M
	SYSPLL_D3
	SYSPLL1_D2
	SYSPLL_D5
	UNIVPLL_D3
	UNIVPLL1_D2

Clock Name	Provided Clock Source
hf_fcamtg_ck	UNIVPLL_D5
	UNIVPLL2_D2
	CLK26M
	UNIVPLL_D52
	UNIVPLL_D208
	UNIVPLL_D104
	CLK26M_D2
	UNIVPLL_D26
	UNIVPLL2_D8
	SYSPLL3_D4
f_fuart_ck	SYSPLL3_D2
	UNIVPLL1_D4
hf_fspi_ck	UNIVPLL2_D2
	CLK26M
	UNIVPLL2_D4
	UNIVPLL1_D4
	UNIVPLL2_D2
	UNIVPLL3_D2
f_fusb20_ck	UNIVPLL1_D8
	CLK26M
	UNIVPLL3_D4
f_fusb30_ck	CLK26M
	UNIVPLL3_D2
	UNIVPLL3_D4
	UNIVPLL2_D4
hf_fmsdc50_0_hclk_ck	CLK26M

Clock Name	Provided Clock Source
	SYSPLL1_D2
	SYSPLL2_D2
	SYSPLL4_D2
	UNIVPLL_D5
	UNIVPLL1_D4
hf_fmsdc50_0_ck	CLK26M
	MSDCPLL_CK
	MSDCPLL_D2
	UNIVPLL1_D4
	SYSPLL2_D2
	MSDCPLL_D4
	VENCPLL_D2
	UNIVPLL1_D2
	MSDCPLL2_CK
	MSDCPLL2_D2
	MSDCPLL2_D4
hf_fmsdc30_1_ck	CLK26M
	UNIVPLL2_D2
	MSDCPLL_D2
	UNIVPLL1_D4
	SYSPLL2_D2
	UNIVPLL_D7
	VENCPLL_D2
hf_fmsdc30_2_ck	CLK26M
	UNIVPLL2_D2
	MSDCPLL_D2
	UNIVPLL1_D4
	SYSPLL2_D2
	UNIVPLL_D7

Clock Name	Provided Clock Source
	VENCPLL_D2
hf_fmsdc30_3_ck	CLK26M MSDCPLL2_CK MSDCPLL2_D2 UNIVPLL2_D2 MSDCPLL2_D4 UNIVPLL1_D4 SYSPLL2_D2 SYSPLL_D7 UNIVPLL_D7 VENCPLL_D2 MSDCPLL_CK MSDCPLL_D2 MSDCPLL_D4
hf_faudio_ck	CLK26M SYSPLL3_D4 SYSPLL4_D4 SYSPLL1_D16
hf_faud_intbus_ck	CLK26M SYSPLL1_D4 SYSPLL4_D2 UNIVPLL3_D2 UNIVPLL2_D8 SYSPLL3_D2 SYSPLL3_D4
hf_fpmicspi_ck	CLK26M SYSPLL1_D8 SYSPLL3_D4 SYSPLL1_D16

Clock Name	Provided Clock Source
hf_fdpilvds1_ck	UNIVPLL3_D4
	UNIVPLL_D26
	SYSPLL3_D4
hf_fatb_ck	CLK26M
	LVDSPLL2_CK
	LVDSPLL2_D2
	LVDSPLL2_D4
	LVDSPLL2_D8
	FPC_CK
hf_fnr_ck	CLK26M
	SYSPLL1_D2
	UNIVPLL_D5
	SYSPLL_D5
hf_fnfi2x_ck	CLK26M
	UNIVPLL1_D4
	SYSPLL2_D2
	SYSPLL1_D4
	UNIVPLL1_D8
	UNIVPLL3_D2
	UNIVPLL2_D2
	SYSPLL_D5
	CLK26M
	SYSPLL4_D4
	UNIVPLL3_D4
	UNIVPLL1_D8
	SYSPLL2_D4
	UNIVPLL3_D2
	SYSPLL_D7
	SYSPLL2_D2

Clock Name	Provided Clock Source
	UNIVPLL2_D2
	SYSPLL_D5
	SYSPLL1_D2
hf_firda_ck	CLK26M
	UNIVPLL2_D4
	SYSPLL2_D4
	UNIVPLL2_D8
hf_fcci400_ck	CLK26M
	VENCPLL_CK
	ARMCA35PLL_600M_CK
	ARMCA35PLL_400M_CK
	UNIVPLL_D2
	SYSPLL_D2
	MSDCPLL_CK
hf_faud_1_ck	UNIVPLL_D3
	CLK26M
	APLL1_CK
	UNIVPLL2_D4
hf_faud_2_ck	UNIVPLL2_D8
	CLK26M
	APLL2_CK
	UNIVPLL2_D4
hf_fmem_mfg_in_as_ck	UNIVPLL2_D8
	CLK26M
	MMPLL_CK
hf_faxi_mfg_in_as_ck	UNIVPLL_D3
	CLK26M
	hd_faxi_ck
	UNIVPLL_D5

Clock Name	Provided Clock Source
hf_fscam_ck	CLK26M
	SYSPLL3_D2
	UNIVPLL2_D4
	SYSPLL2_D4
hf_fnfiecc_ck	CLK26M
	hf_fnfiecc_ck
	SYSPLL_D7
	SYSPLL2_D2
	UNIVPLL2_D2
	UNIVPLL_D5
hf_fnfiecc_ck	SYSPLL1_D2
	hf_fnfiecc_ck_D2
hf_fpe2_mac_p0_ck	CLK26M
	SYSPLL1_D8
	SYSPLL4_D2
	SYSPLL2_D4
	UNIVPLL2_D4
	SYSPLL3_D2
hf_fpe2_mac_p1_ck	CLK26M
	SYSPLL1_D8
	SYSPLL4_D2
	SYSPLL2_D4
	UNIVPLL2_D4
	SYSPLL3_D2
hf_fdpilvds_ck	CLK26M
	LVDSPLL_CK
	LVDSPLL_D2
	LVDSPLL_D4
	LVDSPLL_D8

Clock Name	Provided Clock Source
	FPC_CK
hf_fmsdc50_3_hclk_ck	CLK26M SYSPLL1_D2 SYSPLL2_D2 SYSPLL4_D2 UNIVPLL_D5 UNIVPLL1_D4
hf_fhdcp_ck	CLK26M SYSPLL4_D2 SYSPLL3_D4 UNIVPLL2_D4
hf_fhdcp_24m_ck	CLK26M UNIVPLL_D26 UNIVPLL_D52 UNIVPLL2_D8
hf_fclkrte	clkrtc_int clkrtc_ext CLK26M UNIVPLL3_D8
hf_fspinor_ck	CLK26M CLK26M_D2 SYSPLL4_D4 UNIVPLL2_D8 UNIVPLL3_D4 SYSPLL4_D2 SYSPLL2_D4 UNIVPLL2_D4 ETHERPLL_125M_CK SYSPLL1_D4

Clock Name	Provided Clock Source
hf_fapll_ck	CLK26M
	APLL1_CK
	APLL1_D2
	APLL1_D4
	APLL1_D8
	APLL1_D16
	APLL2_CK
	APLL2_D2
	APLL2_D4
	APLL2_D8
	APLL2_D16
	CLK26M
	CLK26M
hf_fapll2_ck	CLK26M
	APLL1_CK
	APLL1_D2
	APLL1_D4
	APLL1_D8
	APLL1_D16
	APLL2_CK
	APLL2_D2
	APLL2_D4
	APLL2_D8
	APLL2_D16
	CLK26M
	CLK26M
hf_fa1sys_hp_ck	CLK26M
	APLL1_CK
	APLL1_D2

Clock Name	Provided Clock Source
hf_fa2sys_hp_ck	APLL1_D4
	APLL1_D8
	APLL1_D3
hf_fasm_l_ck	CLK26M
	APLL2_CK
	APLL2_D2
	APLL2_D4
	APLL2_D8
	APLL2_D3
hf_fasm_m_ck	CLK26M
	UNIVPLL2_D4
	UNIVPLL2_D2
	SYSPLL_D5
hf_fasm_h_ck	CLK26M
	UNIVPLL2_D4
	UNIVPLL2_D2
	SYSPLL_D5
hf_fi2so1_mck	CLK26M
	APLL1_CK
	APLL2_CK
hf_fi2so2_mck	CLK26M
	APLL1_CK
	APLL2_CK
hf_fi2so3_mck	CLK26M
	APLL1_CK

Clock Name	Provided Clock Source
	APLL2_CK
hf_ftdmo0_mck	CLK26M
	APLL1_CK
	APLL2_CK
hf_ftdmo1_mck	CLK26M
	APLL1_CK
	APLL2_CK
hf_fi2si1_mck	CLK26M
	APLL1_CK
	APLL2_CK
hf_fi2si2_mck	CLK26M
	APLL1_CK
	APLL2_CK
hf_fi2si3_mck	CLK26M
	APLL1_CK
	APLL2_CK
hf_fether_125m_ck	CLK26M
	ETHERPLL_125M_CK
	UNIVPLL3_D2
hf_fether_50m_ck	CLK26M
	ETHERPLL_50M_CK
	APLL1_D3
	UNIVPLL3_D4
hf_fjpgdec_ck	CLK26M
	UNIVPLL_D3
	TVDPLL_429M_CK
	VENCPLL_CK
	SYSPLL_D3
	VCODECPLL_CK

Clock Name	Provided Clock Source
	UNIVPLL1_D2
	ARMCA35PLL_400M_CK
	TVDPLL_429M_CK_D2
	TVDPLL_429M_CK_D4
hf_fspislv_ck	CLK26M
	UNIVPLL2_D4
	UNIVPLL1_D4
	UNIVPLL2_D2
	UNIVPLL3_D2
	UNIVPLL1_D8
	UNIVPLL1_D2
	UNIVPLL_D5
hf_fether_50m_rmii_ck	CLK26M
	ETHERPLL_50M_CK
	UNIVPLL_D26
hf_fcam2tg_ck	CLK26M
	UNIVPLL_D52
	UNIVPLL_D208
	UNIVPLL_D104
	CLK26M_D2
	UNIVPLL_D26
	UNIVPLL2_D8
	SYSPLL3_D4
	SYSPLL3_D2
	UNIVPLL1_D4
	UNIVPLL2_D2
hf_fdi_ck	CLK26M
	TVDPLL_D2
	TVDPLL_D4

Clock Name	Provided Clock Source
	TVDPLL_D8
	VENCPLL_CK
	VENCPLL_D2
	CVBS_CK
	CVBS_D2
hf_ftvd_ck	CLK26M
	CVBS_D2
	UNIVPLL2_D8
hf_fi2c_ck	CLK26M
	UNIVPLL_D26
	UNIVPLL2_D4
	UNIVPLL3_D2
	UNIVPLL1_D4
hf_fpwm_infra_ck	CLK26M
	UNIVPLL2_D4
	UNIVPLL3_D2
	UNIVPLL1_D4
hf_fmsdcOp_aes_ck	CLK26M
	SYSPLL_D2
	UNIVPLL_D3
	VCODECPPLL_CK
hf_fcmsys_ck	CLK26M
	UNIVPLL_D3
	SYSPLL_D3
	SYSPLL1_D2
	SYSPLL2_D2
hf_fgcpu_ck	CLK26M
	SYSPLL_D3
	SYSPLL1_D2

Clock Name	Provided Clock Source
	UNIVPLL1_D2
	UNIVPLL_D5
	UNIVPLL3_D2
	UNIVPLL_D3

3.4 Frequency Hopping Controller

3.4.1 Introduction

Frequency Hopping Controller (FHCTL) helps AP to resolve the de-sense issues. The RF victims are 2G, 3G, BT, FM, Wi-Fi, GPS, etc. The aggressor in AP is the clock generated from PLL. The harmonic of all clock frequencies may de-sense the band of RF system.

3.4.2 Features

FHCTL receives the command from CPU to trigger two mechanisms:

- Spread spectrum clocking
- Frequency hopping

In MT2712, there are 12 hopping PLLs.

Table 3-22 Hopping PLLs

Name	Capability	Range
ARMCA7PLL	Hopping, SSC	{-8%,0}
ARMCA15PLL	Hopping, SSC	{-8%,0}
MAINPLL	Hopping, SSC	{-8%,0}
MMPLL	Hopping, SSC	{-8%,0}
VENCPLL	Hopping, SSC	{-8%,0}
MSDCPLL	Hopping, SSC	{-8%,0}
MSDCPLL2	Hopping, SSC	{-8%,0}
TVDPLL	Hopping, SSC	{-8%,0}
VCODECPPLL	Hopping, SSC	{-8%,0}
MPLL	Hopping, SSC	{-8%,0}
LVDSPLL	Hopping, SSC	{-8%,0}
LVDSPLL2	Hopping, SSC	{-8%,0}

3.4.3 Block Diagram

Figure 3-10 is the system level block diagram of FHCTL.

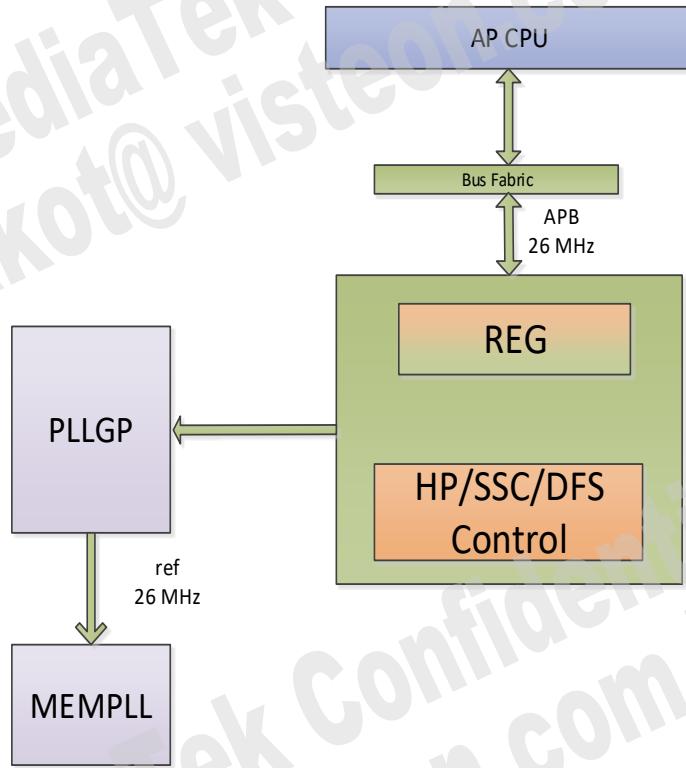


Figure 3-10 FHCTL Block Diagram

3.4.4 Register Definition

For register details, please refer to Chapter 1.4 of “MT2712 IVI Application Processor Registers”.

3.4.5 Programming Guide

3.4.5.1 Initialize FHCTL for Each PLL

- A. Initialize PLL by Apmixedsys
- B. Initialize FHCTL basic setting
 - Turn on the clock of each FHCTL_CORE
 - FHCTL_CLK_CON[11:0]
 - Release software-reset of each FHCTL_CORE
 - FHCTL_RST_CON[11:0]
 - Align frequency (SDM_PCW[21:0]) with Apmixedsys/DDRPHY
 - Setting

- Setting **FHCTLx_DDS (x=0~11)** will update the SDM_PCW value into FHCTL_CORE FSM and trigger one TGL signal for PLL to latch this value
 - Mapping of FHCTL core (0~11)
- Switch to FHCTL_CORE controller
 - FHCTL_HP_EN[11:0]
- Turn on FHCTL_CORE to make DFS/SSC/Dynamic SSC work
 - FHCTLx_EN
 - If this bit is turned off, all functions (DFS/SSC/Dynamic SSC) will not work and frequency will jump to the original frequency in **FHCTLx_DDS[21:0]**

3.4.5.2 Enable/Disable Hopping

- Enable hopping
 - Turn on PLL based-on Apmixedsys/DDRPHY PLL related registers
 - Two setting are controlled by top clock owner
 - Configure FHCTL to align the PLL frequency (SDM_PCW[21:0])
 - Switch control by pll_en from 1'b0 to 1'b1
 - FHCTL_HP_EN[11:0]
- Disable hopping
 - Disable FHCTL all features to keep the SDM_PCW stable
 - SSC/DFS/Dynamic SSC
 - Configure Apmixedsys/DDRPHY register to align SDM_PCW value
 - Switch control by pll_en from 1'b1 to 1'b0
 - FHCTL_HP_EN[11:0]

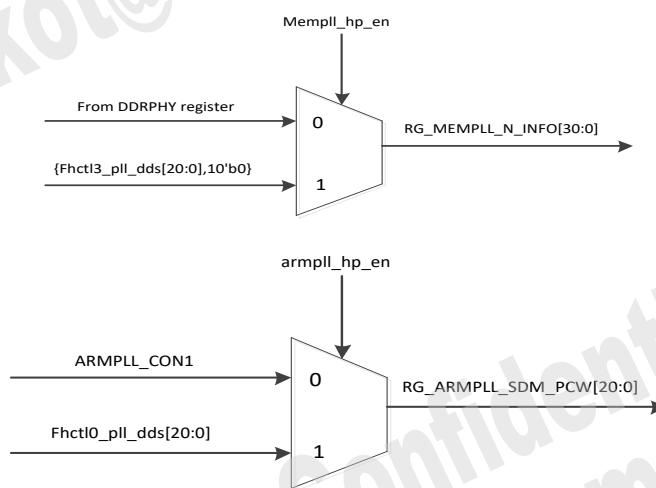


Figure 3-11 Hopping Enable Control

3.5 Top Reset Generation Unit

3.5.1 Introduction

Top Reset Generation Unit (TOPRGU) generates reset signals and distributes them to each system. A watchdog timer is also included in this module.

3.5.2 Features

TOPRGU supports the following features:

- Hardware reset signals for the whole chip
- Software controllable reset for each system (Except for Infrastructure and Apmixedsys System)
- Watchdog timer
- Reset output signals for companion chips

3.5.3 Block Diagram

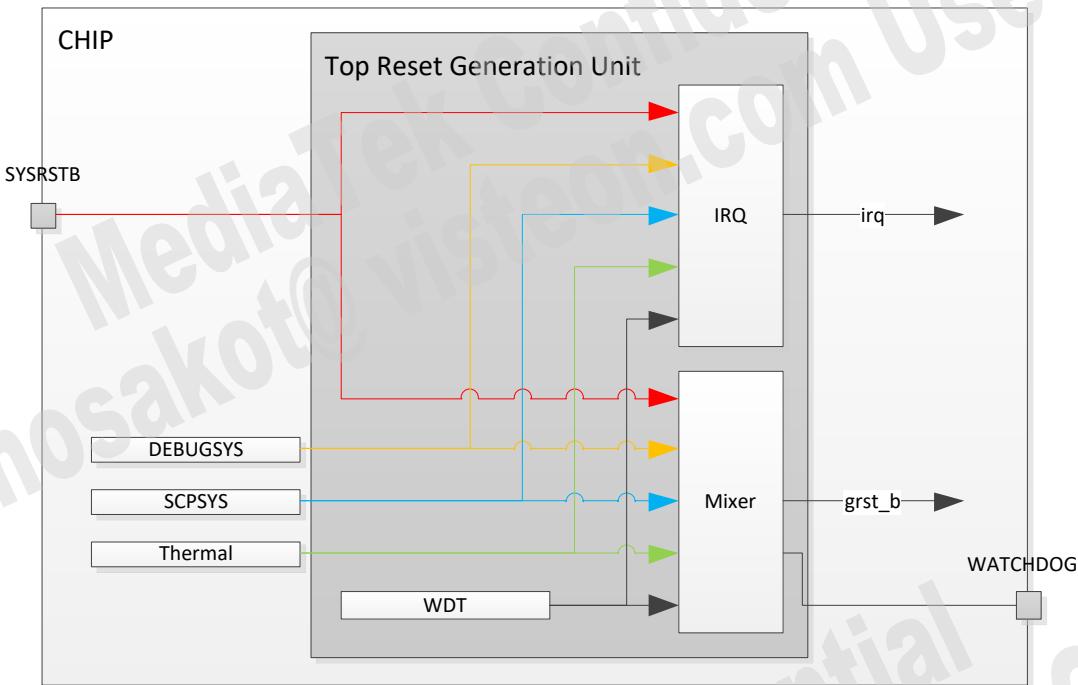


Figure 3-12 Block Diagram of TOPRGU

3.5.4 Register Definition

For register details, please refer to Chapter 1.5 of “MT2712 IVI Application Processor Registers”.

3.5.5 Programming Guide

3.5.5.1 TOPRGU Initial

Enable dual mode reset when TOPRGU is first initialized, because WDT_MODE will not be reset and the dual mode will be disabled if system reset is triggered through WDT_SWRST. These registers will only be reset by SYSRSTB.

The following registers will not be reset by TOPRGU.

- WDT_MODE
- WDT_STA
- WDT_NONRST_REG
- WDT_NONRST_REG2
- WDT_REQ_MODE
- WDT_REQ_IRQ_EN
- WDT_DEBUG_CTL
- WDT_DFD_CTL

3.5.5.2 Watchdog Timer

- Trigger WDT_RESTART right after WDT_LENGTH is updated.
- WDT_SWRST can be triggered without wdt_en set to 1'b1.
- It is recommended to trigger WDT_RESTART before setting wdt_en to 1'b1.

3.5.5.3 IRQ Mode

Dual mode reset is by default on. Therefore, all reset requests are by default with IRQ mode enabled. This means the interrupt, instead of the system reset, is triggered immediately. If users would like to trigger system reset instead of interrupt, change the corresponding configuration of each reset request.

Each reset request can be configured as reset or IRQ separately.

3.5.5.4 Dual Mode Reset

Dual mode reset is system reset after TOPRGU triggers interrupt. The watchdog timer needs to be enabled to complete this function.

In this mode, the watchdog timer will be automatically restarted after interrupt is triggered. AP needs to clear WDT_STA after receiving interrupt from TOPRGU, or system reset will be triggered after the watchdog timer expires.

- Set `wdt_en` = 1'b1.
- Set `dual_mode` = 1'b1.
- Set `wdt_irq`, `thermal_ctl_irq`, `pcm_irq`, `spm_irq` or `debug_irq` to 1'b1.

3.5.5.5 DDR Protect

DDR protect (`rg_ddr_protect_en`) is useless when DDR reserved mode is enabled.

3.5.5.6 DDR Reserved Mode Reset

DDR reserved mode keeps data in DDR during system reset. In order to complete this function, DRAMC, DRMC_CONF, DDRPHY_CONF and EMI_CONF (optional) will not be reset.

- Enable DDR reserved mode when initializing TOPRGU.
- Wait for system reset to be triggered.
- [Optional] Check DDR reserved mode status (`ddr_reserve_sta`).
- After system reset, release DRAMC_CONF protect (set `rg_dramc_conf_iso` = 1'b0).
- Ensure that the related clocks of EMI, DRAMC, DDRPHY are ready (including PLL).
- Wait for `ddr_sref_sta` = 1'b1.
- Release DRAMC protect (set `rg_dramc_iso` = 1'b0).
- Release DRAMC self-refresh control (set `rg_dramc_sref` = 1'b0).
- Wait for `ddr_sref_sta` = 1'b0.

3.5.5.7 DFD3.0

- Trigger DFD enable (set `rgdfd_en` = 1'b1).
- Set reset delay timer (set `rgdfd_timeout`).
- Set WDT_MODE.
- Set WDT_SWRST.
- Wait timeout of reset delay timer (~1 ms).
- HW WDT reset is asserted.

3.5.5.8 History

- WDT_LENGTH needs to be reset, or SYSTEM will enter RESET loop if WDT_LENGTH has been set to very short before resetting trigger.

3.6 MTCMOS Domains and DFS

3.6.1 Introduction

This chapter introduces the Multi-threshold CMOS (MTCMOS) power domains and Dynamic Frequency Scaling (DFS) information in MT2712.

3.6.2 Features

- Coarse-grained MTCMOS control and DFS control are provided to achieve high performance with low power in MCU System (MCUSYS). There is one cluster in MCUSYS with 4-core CA35 & 2-core CA72, a total of six MTCMOS domains.
- Four MTCMOS domains in INFRA and Graphics Processing Unit (GPU), five MTCMOS domains in DRAM Controller (DRAMC), and only one MTCMOS domain in Video Decoder (VDEC), Video Encoder (VENC), Image System (IMGSYS), and Multimedia System (MMSYS).
- There are CPU and Graphics Processing Unit (GPU) DFS in MT2712, but no DFS in the SOC. For both CA35 and CA72, the minimum frequency is 600 MHz. The maximum performance of CA35 is running at 1.2 GHz, and the maximum performance of CA72 is running at 1.5 GHz.
- For high performance scenarios, GPU can be set at a maximum frequency of 520 MHz; in light-loading scenarios, GPU should be set at a frequency of 200 MHz.

3.6.3 Block Diagram

Figure 3-13 shows all power blocks in MT2712, including CPU, GPU, DRAMC, VDEC, VENC, IMAGE, MMSYS and INFRA.

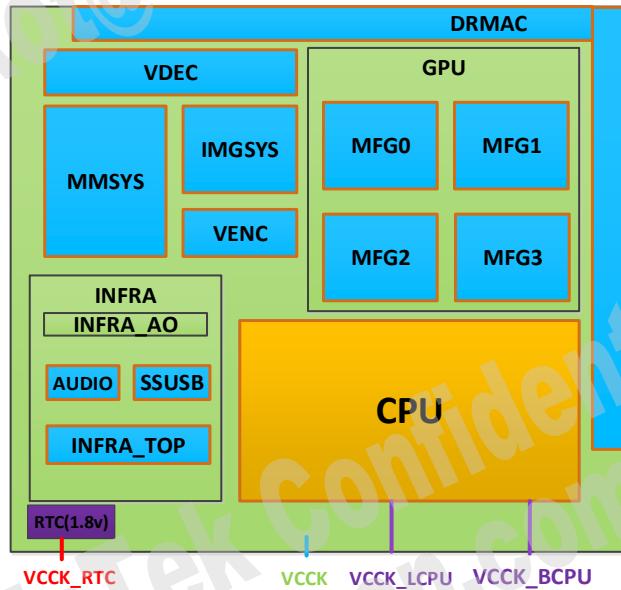


Figure 3-13 Block Diagram of MT2712 Power Domain

There are five different power sources in CPU: VCCK_LCPU, VCCK_SRAM_LCPU, VCCK, VCCK_SRAM_BCPU, and VCCK_BCPU. VCCK_LCPU and VCCK_SRAM_LCPU are for CA35. VCCK_SRAM_BCPU and VCCK_BCPU are for CA72. Other power blocks are in VCORE power domain with supply net VCCK. All the power sources are driven from external power supply and the voltage is 1 V. A total of six MTCMOS domains are listed in Table 3-23.

Table 3-23 Power Domains in MCUSYS

Domain	Description	Boot-up Power State
CA35-CPU0	CA35 CPU0 power domain	ON
CA35-CPU1	CA35 CPU1 power domain	ON
CA35-CPU2	CA35 CPU2 power domain	ON
CA35-CPU3	CA35 CPU3 power domain	ON
CA72-CPU0	CA72 CPU0 power domain	OFF
CA72-CPU1	CA72 CPU1 power domain	OFF

All the other MTCMOS domains reside in the VCORE power domain.

Table 3-24 VCORE MTCMOS Domains

Domain	Description	Boot-up Power State
VDEC	Video decoder power domain	OFF
IMG	Image power domain	OFF
VENC	Video encoder power domain	OFF
AUDIO	Audio power domain	OFF
SSUSB	ssusb power domain	ON
INFRA_TOP	Infrasys power domain	ON
MMSYS	Multimedia system power domain	OFF
MFG0	GPU MFG0 power domain	OFF
MFG1	GPU MFG1 power domain	OFF
MFG2	GPU MFG2 power domain	OFF
MFG3	GPU MFG3 power domain	OFF
DRAMC	DRAM controller power domain	ON

To further reduce the system-level power consumption, a System Power Manager (SPM) is implemented to control the top level power by different power scenarios. Table 3-25 shows the power mode scenarios of MT2712.

Table 3-25 Power Mode Scenarios

Function Blocks	Power Mode				
	HW Power Off	SW Control (MTCMOS control)			
		Power Off	Cold Sleep of WFI with 32K XTAL (Suspend A)	Cold Sleep with 26M XTAL (Suspend B)	Warm Sleep (Deep Idle)
CPU CA72	OFF	OFF	OFF	OFF	ON
CPU CA35	OFF	OFF	OFF	Sleep, L2 dormant	ON

Function Blocks	Power Mode				
	HW Power Off	SW Control (MTCMOS control)			
		Cold Sleep of WiFi with 32K XTAL (Suspend A)	Cold Sleep with 26M XTAL (Suspend B)	Warm Sleep (Deep Idle)	Active
CA72 PLL	OFF	OFF	OFF	OFF	ON
CA35 PLL	OFF	OFF	OFF	OFF	ON
GPU Mali 880	OFF	OFF	OFF	OFF	ON
DDR	OFF	Self refresh (PSR)	Self refresh (PSR)	Self refresh (PSR)	ON
DDR Clock	OFF	OFF	OFF	OFF	ON
VDEC	OFF	OFF	OFF	OFF	ON
VENC	OFF	OFF	OFF	OFF	ON
Infra	OFF	AO part ON, non-AO part Off	AO part ON, non-AO part Off	ON	ON
Misc.	OFF	ON (WiFi, SW control)	ON (need control SPI1/SPI4/UART5)	ON	ON
SPM	OFF	ON	ON	ON	ON
RTC	ON	ON	ON	ON	ON
Power Mode Description	Power Off, only RTC alive	Deep sleep mode; wait for external interrupt	Deep sleep mode; SPI/UART co-work with the external MCU	Normal sleep mode, with higher power than deep sleep mode, and a faster warm boot-up speed	Active mode

3.7 Real Time Clock

3.7.1 Introduction

Real Time Clock (RTC) provides time and alarm functions. RTC's clock is 32.768 kHz and input is from PAD_RTC_XI. In addition to providing timing data, an alarm interrupt also can be generated. The wakeup Interrupt Request (IRQ) can be sent to System Power Management (SPM) and External MCU (EMCU). RTC interrupts corresponding to seconds, minutes, hours and days can be generated whenever the time counter value reaches a maximum value (e.g. 59 for seconds and minutes, 23 for hours, etc.). The year span up to 2127 is supported. The maximum day-of-month values depend on the leap year condition, and are stored in the RTC block.

Table 3-26 RTC Clock Specification

Parameters	Value	Unit
Nominal Frequency	32.768	kHz

Parameters	Value	Unit
Frequency Tolerance(full range ⁽¹⁾)	± 100	ppm
Operation Temperature Range	-40~+85	°C
Operation Voltage	1.8	V

(1) Initial accuracy, temperature drift, aging and voltage effects should be combined when evaluating RTC clock. Maximum RTC error is about 8.64 seconds per day.

3.7.2 Features

RTC has the following features:

- Work in isolation power domain (1.8 V)
- 32 kHz clock input from PAD_RTC_XI and output to chip
- Support time and alarm function
- Generate IRQ to SPM and EMCU

3.7.3 RTC Block Diagram

RTC contains rtc and rtc_macro. The block diagram is shown in Figure 3-14. The rtc is a bridge between APB bus and rtc_macro. The rtc_macro is a digital macro with different power domains to core. So rtc_marco can work independently if core power is down.

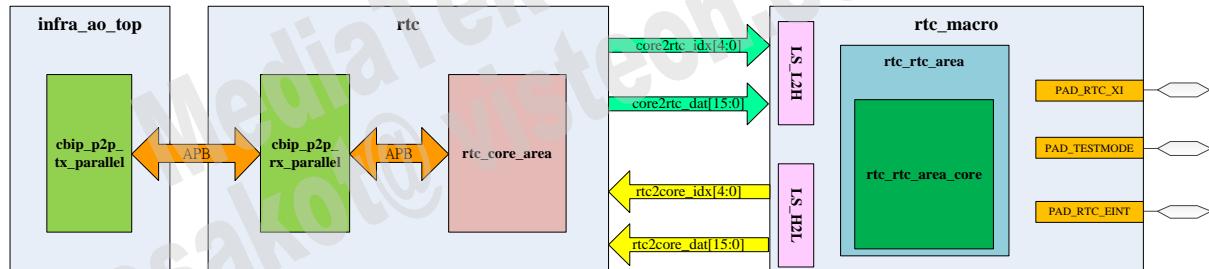


Figure 3-14 RTC Block Diagram

3.7.4 Register Definition

For register details, please refer to Chapter 1.7 of “MT2712 IVI Application Processor Registers”.

3.7.5 Programming Guide

RTC's registers are brief, as shown in Figure 3-2. A bus write is acceptable only when KEY_BBU=0x43. After modifying all the RTC registers they want to modify, users must write RTC_WRTGR to 1 to trigger the transfer. If the powerkey is matched, i.e., RTC_POWERKEY1 = 0xA357 and RTC_POWERKEY2 = 0x67D2, this shows that the channel between rtc and rtc_macro is locked, and users have to perform the unlock flow, as shown in Figure 3-15, to enable the writing interface.

ADDR	IDX		Bits	Name
1001_1000h+ 00h	1	RTC_BBPU	15:8	KEY_BBPU KEY_BBPU=0x43
	2	RTC_IRQ_STA	6	CBUSY
	3	RTC_IRQ_EN	5	RELOAD
	4	RTC_CILEN	4	CLRPKY
10h	5	RTC_AL_MASK		
	6	RTC_TC_SEC		
	7	RTC_TC_MIN		
	8	RTC_TC_HOU		
20h	9	RTC_TC_DOM		
	10	RTC_TC_DOW		
	11	RTC_TC_MTH		
	12	RTC_TC_YEA		
30h	13	RTC_AL_SEC		
	14	RTC_AL_MIN		
	15	RTC_AL_HOU		
	16	RTC_AL_DOM		
40h	17	RTC_AL_DOW		
	18	RTC_AL_MTH		
	19	RTC_AL_YEA		
50h	20	RTC_POWERKEY1		
	21	RTC_POWERKEY2		
	22	RTC_DIFF		
	23	RTC_CON0		
60h	24	RTC_CON1		
	25	RTC_PDN1		
	26	RTC_PDN2		
	27	RTC_SPAR1		
70h	28	RTC_SPAR2(RTC_PROT)		
	29			
	30			
78h	31	RTC_WRTGR		

RTC_DIFF ranges from 0x800 (-2048) to 0x7fd (2045)

0xA357

0x67D2

Unlock flow: Step1: *RTC_PROT=0x9136; // compare 15 bits: bit 15~1
Step2: *RTC_WRTRG=1;
Step3: while(*RTC_BBPU & 0x40) {} // Timeout period: 120usec
Step4: *RTC_PROT=0x586A; // compare 15 bits: bit 15~1
Step5: *RTC_WRTRG=1;
Step6: while(*RTC_BBPU & 0x40) {} // Timeout period: 120usec

0x9136 -> 0x586A

Figure 3-15 RTC Register Map Sketch

4 MCU and Bus Fabric

4.1 MCU System

4.1.1 Introduction

The MCU System (MCUSYS) is a subsystem responsible for running operating system and application programs in MT2712. It consists of:

- Little cluster: 4 × CA35
- Big cluster: 2 × CA72
- MCISA Bus Interface Unit (in-house MTK bus) Design for Debug (DFD) Controller 3.0
- Phase-Locked Loop (PLL) Divider
- GIC-400 with 220 interrupt sources

The MCUSYS supports Dynamic Frequency Scaling (DFS) technology which allows the CPU to run at different frequency configurations for various application requirements. In standby/dormant mode, little cluster or big cluster can be completely shut down to further save power.

4.1.2 Features

4.1.2.1 Big Little Cluster Unit Specifications

- Quad-core ARM® Cortex-CA35 MPCore™ operating up to 1.2 GHz, 32 KB L1, 512 KB L2 with ECC
- Dual-core ARM® Cortex-CA72 MPCore™ operating up to 1.4 GHz, 1MB L2 with ECC
- MCISA bus operating at 650 MHz
- Support L2 256 KB sharing in little cluster
- Support SPMC to control power sequence
- Support DFD 3.0
- Support coherency

4.1.3 MCUSYS Block Diagram

Figure 4-1 shows an overview of MCUSYS.

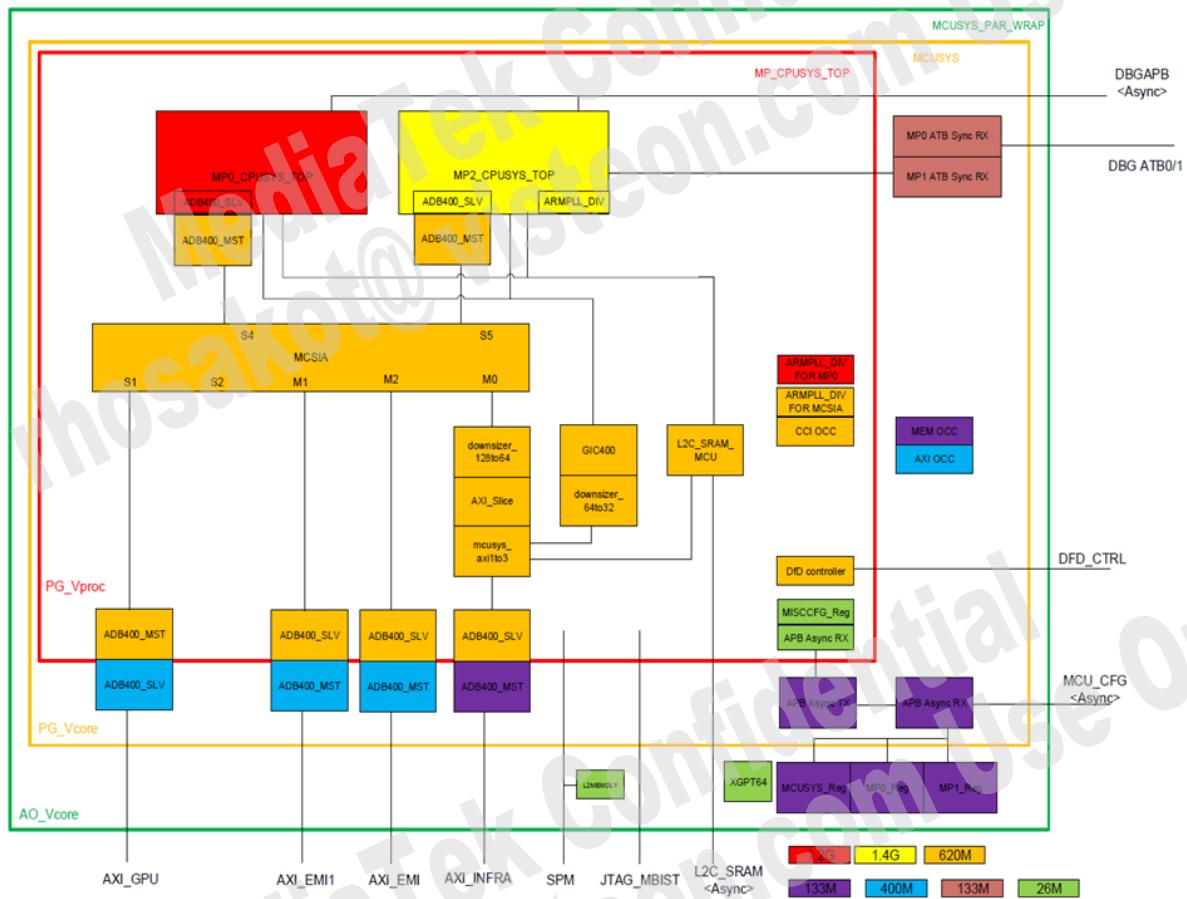


Figure 4-1 MCUSYS Block Diagram

4.1.4 Register Definition

For register details, please refer to Chapter 2.1 of “MT2712 IVI Application Processor Registers”.

4.1.5 MCUSYS Interrupts

In MT2712, the Interrupt bus input to MCUSYS is as wide as 220 bits (IRQS[291:0]). For the detailed interrupt map, please refer to Table 4-1. Since all the interrupt signals are connected as Shared Peripheral Interrupt (SPI) of GIC-400, the interrupt ID starts from 32.

Table 4-1 Interrupt Request List for MCUSYS

GIC ID	Interrupt Source/Name	Polarity	Trigger Type	SPI/IRQS	irqs_ext
32	mp0_nIRQOUT[0]	L	Level	0	-
33	mp0_nIRQOUT[1]	L	Level	1	-
34	mp0_nIRQOUT[2]	L	Level	2	-
35	mp0_nIRQOUT[3]	L	Level	3	-
36	0	-	-	4	-
37	0	-	-	5	-

GIC ID	Interrupt Source/Name	Polarity	Trigger Type	SPI/IRQS	irqs_ext
38	0	-	-	6	-
39	0	-	-	7	-
40	mp0_nPMUIRQ[0]	L	Level	8	-
41	mp0_nPMUIRQ[1]	L	Level	9	-
42	mp0_nPMUIRQ[2]	L	Level	10	-
43	mp0_nPMUIRQ[3]	L	Level	11	-
.....	-	-
.....	-	-
.....	-	-
.....	-	-
80	mp0_nEXTERRIRQ	L	Level	48	-
81	mp0_nINTERRIRQ	L	Level	49	-
82	mp0_CTIIRQ_sync[0]	L	Level	50	-
83	mp0_CTIIRQ_sync[1]	L	Level	51	-
84	mp0_CTIIRQ_sync[2]	L	Level	52	-
85	mp0_CTIIRQ_sync[3]	L	Level	53	-
86	0	-	-	54	-
87	0	-	-	55	-
88	0	-	-	56	-
89	0	-	-	57	-
90	CCI_nEVNTCNTOVERFLOW[0]	L	Level	58	-
91	CCI_nEVNTCNTOVERFLOW[1]	L	Level	59	-
92	CCI_nEVNTCNTOVERFLOW[2]	L	Level	60	-
93	CCI_nEVNTCNTOVERFLOW[3]	L	Level	61	-
94	CCI_nEVNTCNTOVERFLOW[4]	L	Level	62	-
95	CCI_nERRORIRQ	L	Level	63	-
96	xgpt_irq_int[0]	-	-	64	-
97	xgpt_irq_int[1]	-	-	65	-
98	xgpt_irq_int[2]	-	-	66	-
99	xgpt_irq_int[3]	-	-	67	-
100	xgpt_irq_int[4]	-	-	68	-
101	xgpt_irq_int[5]	-	-	69	-
102	xgpt_irq_int[6]	-	-	70	-
103	xgpt_irq_int[7]	L	Level	71	-
104	irqs_ext[0]	H	Level	72	0
105	irqs_ext[1]	H	Level	73	1
106	irqs_ext[2]	H	Level	74	2
.....	H	Level
.....	H	Level
.....	H	Level
.....	H	Level

GIC ID	Interrupt Source/Name	Polarity	Trigger Type	SPI/IRQS	irqs_ext
.....	H	Level
.....	H	Level
403	irqs_ext[219]	H	Level	291	219
404	sec_vio_abort_n	L	Level	292	-
405	big_nIRQOUT[0]	L	Level	293	-
406	big_nIRQOUT[1]	L	Level	294	-
407	big_nIRQOUT[2]	L	Level	295	-
408	big_nIRQOUT[3]	L	Level	296	-
409	big_nPMUIRQ[0]	L	Level	297	-
410	big_nPMUIRQ[1]	L	Level	298	-
411	big_nPMUIRQ[2]	L	Level	299	-
412	big_nPMUIRQ[3]	L	Level	300	-
.....	L	Level
.....	L	Level
.....	L	Level
.....	L	Level
429	big_nEXTERRIRQ	L	Level	317	-
430	big_CTIIRQ_sync[0]	-	-	318	-
431	big_CTIIRQ_sync[1]	-	-	319	-
432	big_CTIIRQ_sync[2]	-	Level	320	-
433	big_CTIIRQ_sync[3]	-	Level	321	-
434	big_xgpt_irq[0]	-	-	322	-
435	big_xgpt_irq[1]	-	-	323	-
436	big_xgpt_irq[2]	-	-	324	-
437	big_xgpt_irq[3]	-	-	325	-
438	big_nINTERRIRQ	L	Level	326	-
439	big_nAXIERRIRQ	L	Level	327	-
440	big_sec_vio_abort_n	L	Level	328	-
441	big_CPUTOP_IRQ[0]	-	-	329	-
442	big_CPUTOP_IRQ[1]	-	-	330	-
443	big_CPUTOP_IRQ[2]	-	-	331	-
444	mp0_ptp3_ocp_irq0	-	-	332	-
445	mp0_ptp3_ocp_irq1	-	-	333	-
446	mp0_ptp3_ocp_irq2	-	-	334	-
447-463	0	-	-	335-351	-

4.1.6 MCUSYS Power Architecture

4.1.6.1 Instruction

MCUSYS supports MTCMOS, as shown below.

Table 4-2 Power State

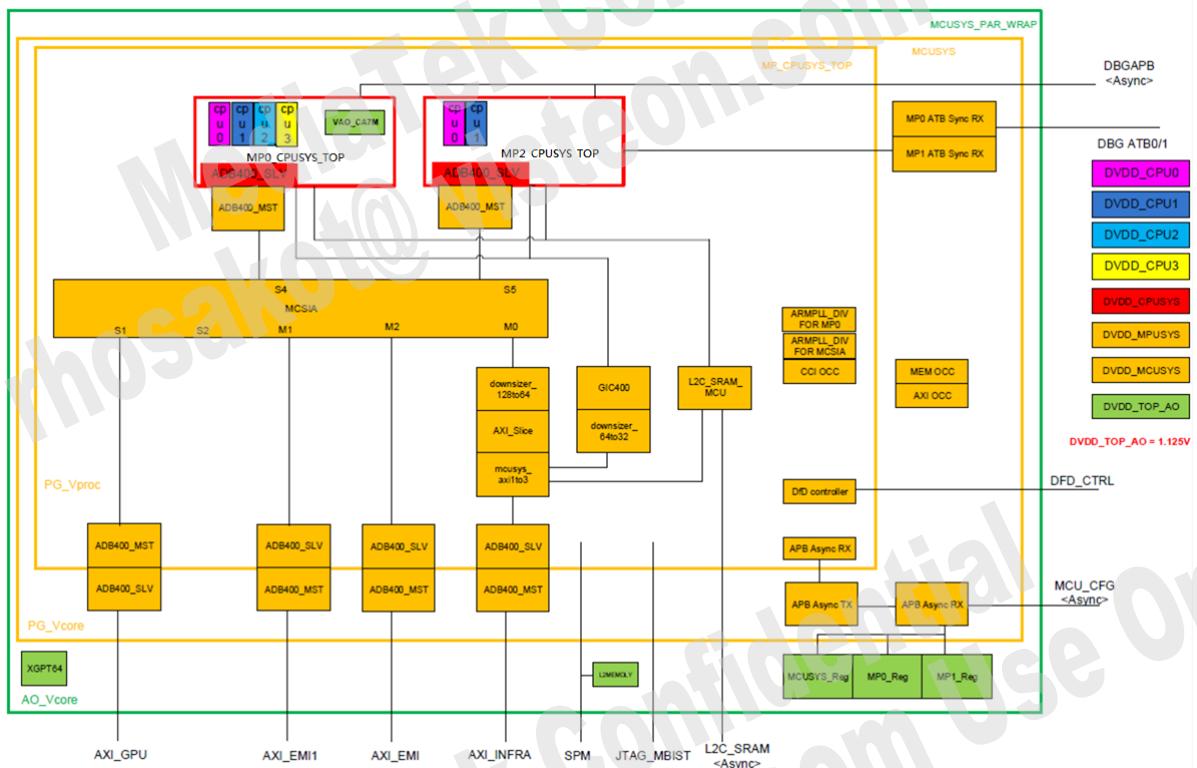


Figure 4-2 Power Domain

4.2 Debug System

4.2.1 Introduction

The debug and trace system in MT2712 is built on ARM CoreSight architecture and components to provide flexible low latency debugging for multi-processor system and high speed logging for all SW/HW trace functions. In this document, the focus is on target's debugging and trace architecture design, so that the existing Debug and Trace (DnT) framework and the commercial ARM debugger can be used concurrently.

4.2.2 Reference

There is a significant amount of literature from ARM on the CoreSight architecture and components. Please refer to the following documents for the specific details of any CoreSight components. Register descriptions and more detailed functional specifications can also be found in the following ARM specifications:

ARM Debug Interface V5.0

- [CoreSight Technology System Design Guide](#)
- [CoreSight Component Technical Reference Manual](#)
- [CoreSight Architecture Specification](#)

4.2.3 Features

The debug subsystem is responsible for control and observation of the target system. Devices use external emulators to halt the processor. Then the processor cores execute the code step by step, observe system state, and download code to the target device. In MT2712, debug system may use a single 2-pin interface to control all processor cores in the system via the Debug Access Port (DAP). In addition, legacy JTAG interfaces are also available for emulation. Three Debug Communication Channels (DCC) are provided and can be used in numerous ways to support the exchange of data between the target and external host. The channels are intended for the APMCU so that this processing system has its own communication channel. The DCCs also share the same 2-pin interface as the emulators allowing for a highly efficient pin count.

4.2.4 Block Diagram

In MT2712, DnT System is designed to provide debug and trace mechanism for MCUSYS. The debug and trace architecture is shown in Figure 4-3.

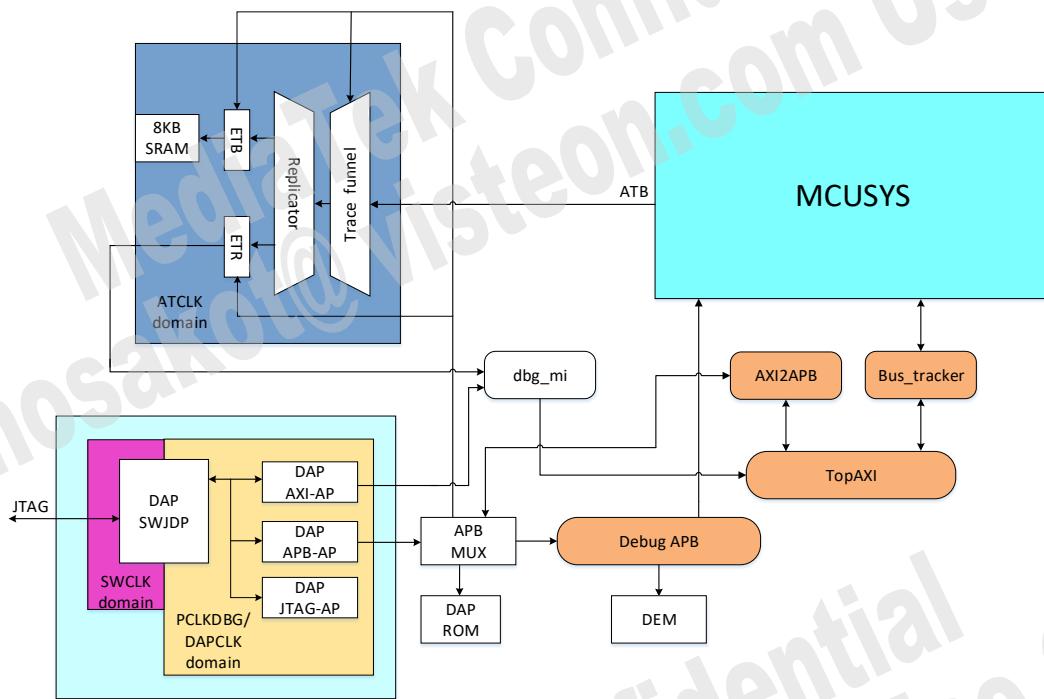


Figure 4-3 Debug System Block Diagram in MT2712

The debug system contains the following components – CSSYS, debug APB and Cross Trigger Interface/Matrix which operate in different clock/power domains. The trace subsystem contains ETR, Funnel, Replicator, and ETB.

4.2.5 JTAG AC Timing

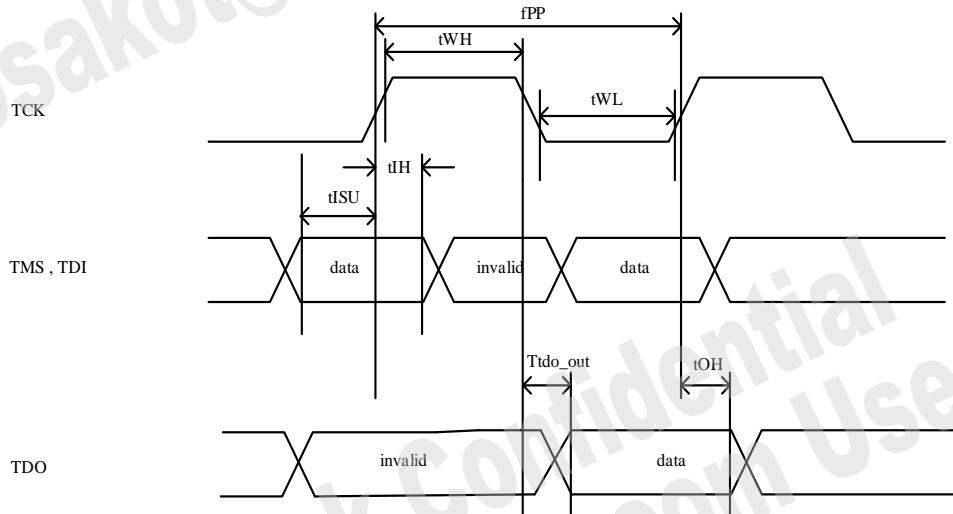


Figure 4-4 JTAG Timing Diagram

Table 4-3 JTAG Timing Parameter

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK frequency	fPP	-	9.995	MHz
CLK low time	tWL	50.392	-	ns
CLK high time	tWH	49.655	-	ns
Host CMD/DAT output (refer to CLK)				
Input setup time	tISU	44.313	-	ns
Input hold time	tIH	49.936	-	ns
Host CMD/DAT input (refer to CLK)				
TDO output delay	Tdo_out	-	11.675	ns
Input hold time	tOH	62.507	-	ns

4.2.6 Application Processor Debug Subsystem

In general, modules belong to Application Processor (AP) system, are not included in the modem system. This includes APMCU subsystem, multimedia subsystem, peripherals and system bus fabric. The following table lists the clock frequency of the major AP system bus fabric.

Table 4-4 Clock Frequency of Major AP System Bus Fabric

Name	Frequency
AP AXI bus	266 MHz
AP APB bus	66 MHz
AP Debug APB bus	66 MHz

APMCU subsystem is a multi-core system. It has its own power plane for DFS technology. CoreSight components such as ETM and CTI/CTM are integrated inside this multi-core system. The operating frequency will be changed with the core's clock speed. The debugging of APB and ATB needs asynchronous bridges to connect to AP's debug system.

4.2.7 Register Definition

For register details, please refer to Chapter 2.2 of "MT2712 IVI Application Processor Registers".

4.2.8 Programming Guide

Please refer to the users' In-Circuit Emulator (ICE) tool operating, such as Codeviser (CVD).

4.3 System Interrupt Controller

4.3.1 Introduction

For processors like CA35 or CA72 which have embedded interrupt controllers (Generic Interrupt Controller (GIC)), part of the MCUSYS will need to keep feeding clock and power to make the interrupt functional. However, due to power/leakage overhead introduced by higher clock ratio and deep submicron processes, reserving an always-on (or frequently turned-on) domain in MCUSYS has led to ineffective power consumption. The system interrupt controller (SYS_CIRQ) is a low power interrupt controller designed to work outside MCUSYS as a second-level interrupt controller. With SYS_CIRQ, MCUSYS can be completely turned off to reduce the system power consumption without losing interrupts.

4.3.2 Features

SYS_CIRQ supports up to 219 interrupts which can configure the following attributes individually:

- Polarity inversion
- Edge/level trigger selection

The 219 interrupts will be fed through SYS_CIRQ and be connected to GIC in MCUSYS. When SYS_CIRQ is enabled, it will record the edge-sensitive interrupts and generate a pulse signal to CPU GIC when the flush command is executed.

4.3.3 Block Diagram

The block diagram of SYS_CIRQ in MT2712 is shown in Figure 4-5.

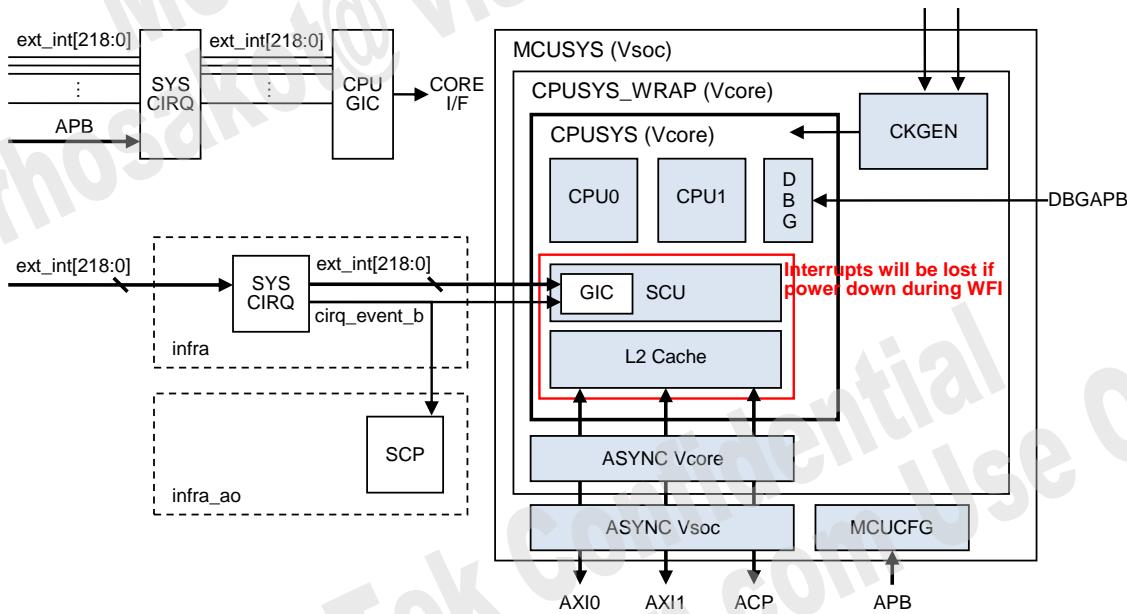


Figure 4-5 SYS_CIRQ System Level Block Diagram

The SYS_CIRQ controller is integrated between MCUSYS and other interrupt sources as the second-level interrupt controller. All interrupts are fed through SYS_CIRQ controller and then bypassed to MCUSYS. In normal mode (where MCUSYS GIC is active), SYS_CIRQ is disabled and interrupts will be directly issued to MCUSYS. When MCUSYS enters sleep mode, GIC is powered down. SYS_CIRQ controller will be enabled and then monitors all edge-triggered interrupts (only edge-triggered interrupt will be lost in this scenario). When an edge-triggered interrupt is triggered, it will be recorded in the SYS_CIRQ_STA register and can be restored to GIC by SW context restoration or the SYS_CIRQ flush function.

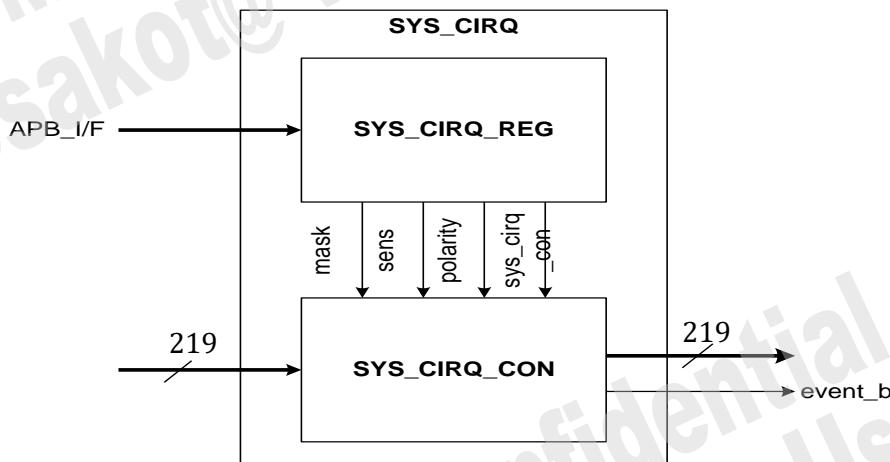


Figure 4-6 System Interrupt Controller Architecture Block Diagram

As shown in Figure 4-6, SYS_CIRQ_REG stores the mask/sensitivity/polarity attributes of each interrupt signals, and SYS_CIRQ_CON is used to mask and detect edge-triggered interrupts.

4.3.4 Register Definition

For register details, please refer to Chapter 2.3 of “MT2712 IVI Application Processor Registers”.

4.3.5 Programming Guide

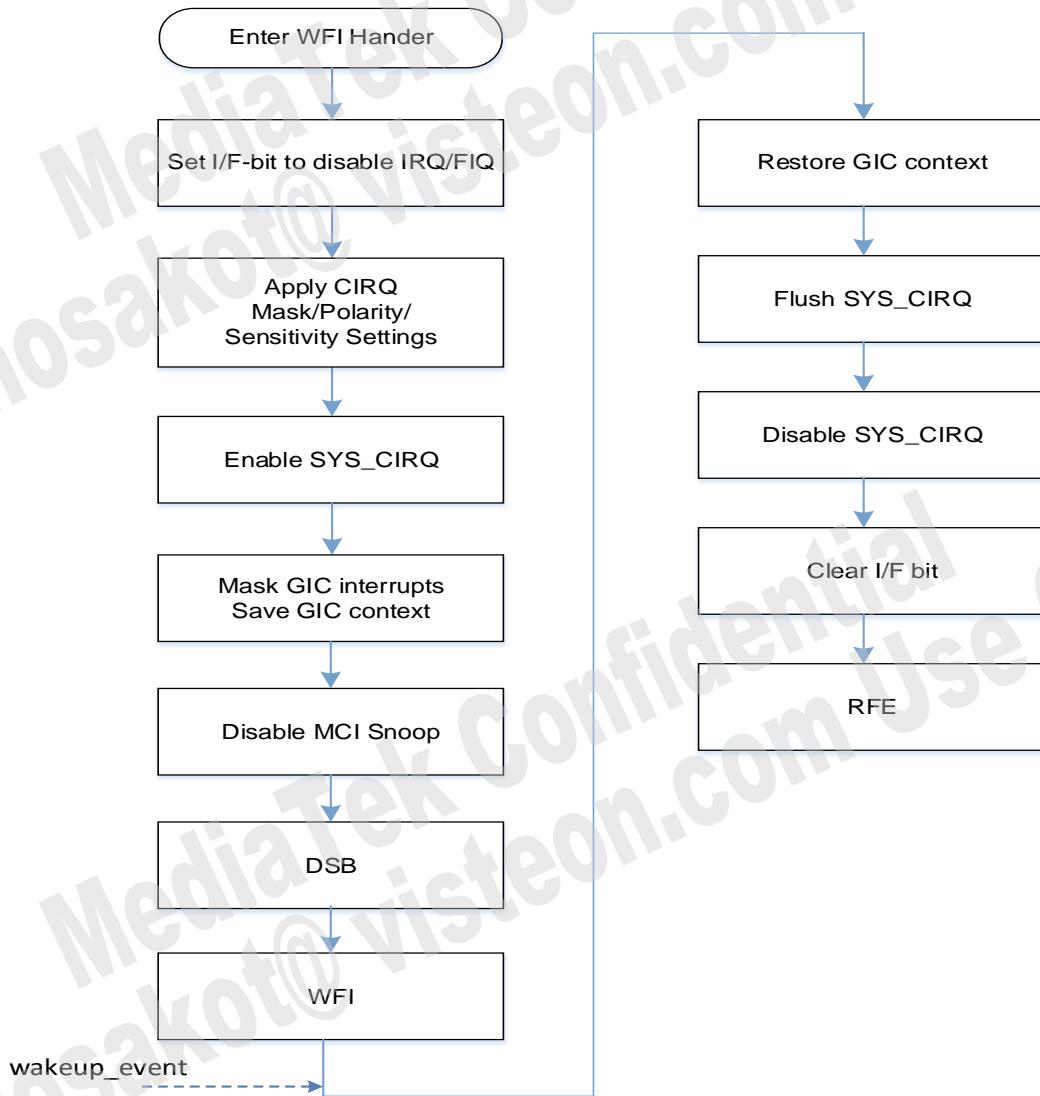


Figure 4-7 Software Programming Flow Chart

- DSB: Data Synchronization Barrier (ARM instruction)
- WFI: Wait For Interrupt (ARM instruction)
- RFE: Return From an Exception (ARM instruction)

4.4 External Interrupt Controller

4.4.1 Introduction

External Interrupt Controller (EINTC) processes all off-chip interrupt sources and forwards interrupt request signals to AP MCU.

4.4.2 Features

EINTC supports up to 229 external interrupt signals and performs the following processes for the interrupt signals from external sources:

- Polarity inversion
- Edge/level trigger selection
- De-bounce with a configurable 32 kHz clock (optional)

According to the register configuration, the external interrupt source will be forwarded to the AP MCU built-in interrupt controller with different IRQ signals, eint_irq or eint_direct_irq. EINTC generates wake-up events to SPM controller.

Eint[39:0] supports de-bounce feature. The de-bounce time is shown in Table 4-5.

Table 4-5 De-bounce Time in MT2712

De-bounce Time
0.5 ms
1 ms
16 ms
32 ms
64 ms
128 ms
256 ms
512 ms

4.4.3 Block Diagram

Figure 4-8 shows the block diagram of external interrupt controller in MT2712.

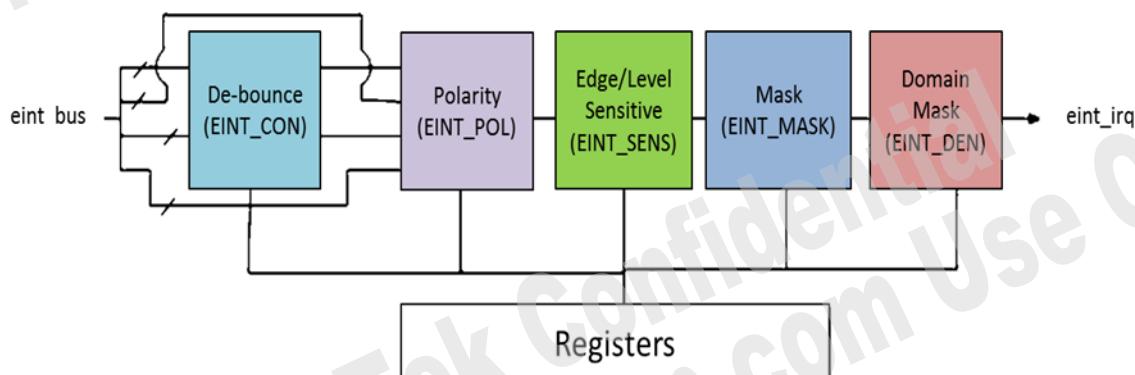


Figure 4-8 Block Diagram of External Interrupt Controller in MT2712

Normally, the external interrupt source goes through the de-bounce unit which is driven by a 32 kHz clock and triggers the corresponding CPU with eint_irq. Therefore the minimum latency from eint_bus to eint_irq will be 30.52 μ s. Since the latency introduced by the de-bounce module may be too long for some applications, EINTC provides an alternative path which bypasses the de-bounce module and directly triggers the interrupt signals, eint_direct_irq[3:0] to AP MCU.

Table 4-6 shows the signals' connections to the interrupt controllers of different CPUs.

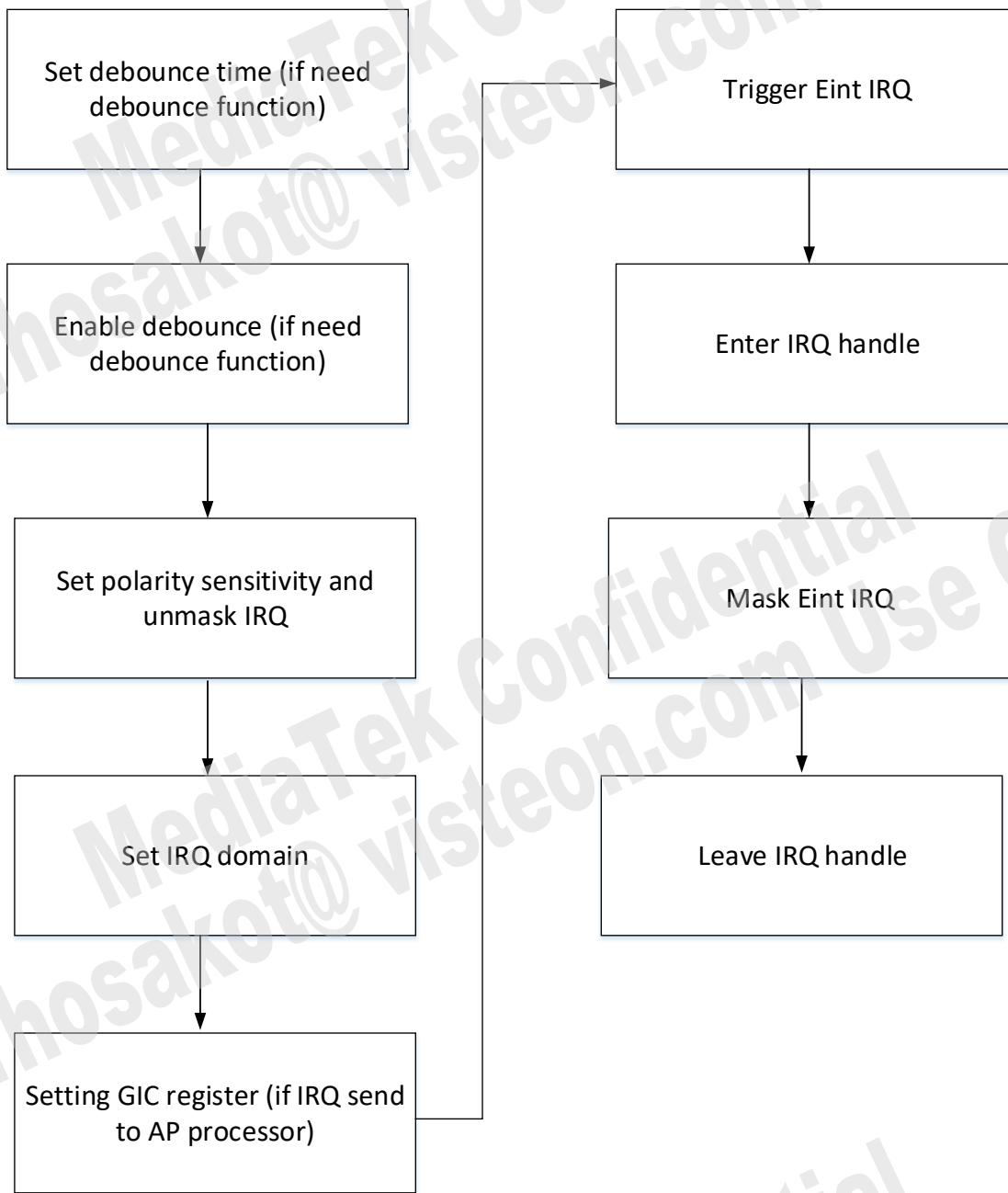
Table 4-6 External Interrupt Request Signal Connection

IRQ Name	AP MCU IRQ
eint_irq	IRQ[185]
eint_event_b	IRQ[187]
eint_direct_irq[0]	IRQ[189]
eint_direct_irq[1]	IRQ[190]
eint_direct_irq[2]	IRQ[191]
eint_direct_irq[3]	IRQ[192]

4.4.4 Register Definition

For register details, please refer to Chapter 2.4 of “MT2712 IVI Application Processor Registers”.

4.4.5 Programming Guide



4.5 Infrastructure System Configuration

4.5.1 Introduction

Infrastructure System Configuration (INFRACFG) module provides reset, clock and miscellaneous control signals in the infrastructure system.

4.5.2 Features

INFRACFG provides the following control signals to the functional blocks inside the infrastructure system:

- Software reset signals
- Clock gating control signals
- Dynamic Clock Management (DCM) control signals
- Top AXI bus fabric control signals
- DCM function

4.5.2.1 Dynamic Clock Management Details

The DCM function is used to slow down the clock frequency for power saving when the system is in idle state automatically.

Figure 4-9 shows a sample clock waveform when DCM is activated. In this example, the clock frequency in DCM mode is set to a quarter of the original clock frequency. The ratio of clock frequency slow-down is controlled by the INFRA_DCMFSEL register.

After the bus idle signal is low, it will take several cycles of latency to make the slow-down clock return to the normal frequency. The cycle number varies with the runtime status of the clock gating logic and will somehow cause minor performance impact. In order to minimize the impact when the system is in heavy load status, the INFRA_DCMDBC register controls the cycle count once the bus idle signal is asserted. Setting the de-bounce cycle to be longer and enabling the function will reduce the probability of the system entering the DCM mode.

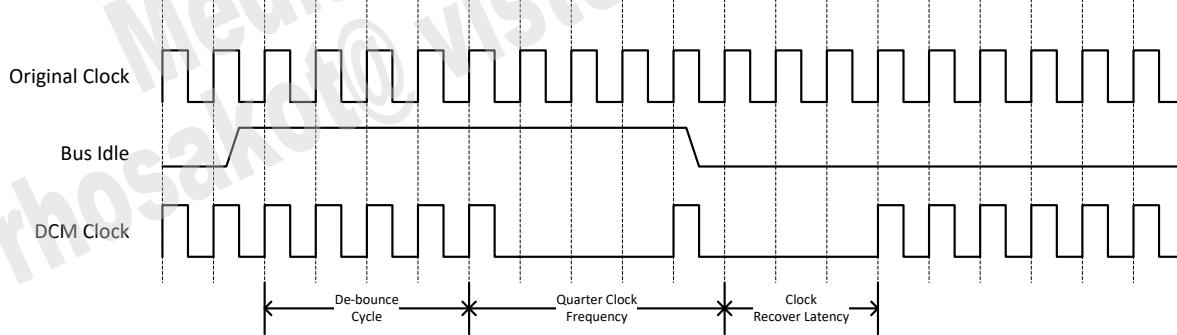


Figure 4-9 DCM in Action

4.5.2.2 AXI Fabric Control

The AXI fabric control registers help to prevent the system bus from hanging up caused by improper access while some parts of the system are in the power-down state. See Figure 4-10 for the AXI fabric interconnection. In AXI fabric, the MCUSYS, GPU, MMSYS and PERISYS AXI Masters are connected to DRAM Scheduler (EMI) for high bandwidth DRAMC access. The internal RAM/ROM and registers are accessed through Top AXI fabric. The

sleep protectors are added to bus ports and bus components. The corresponding control registers can be found in Register Definition.

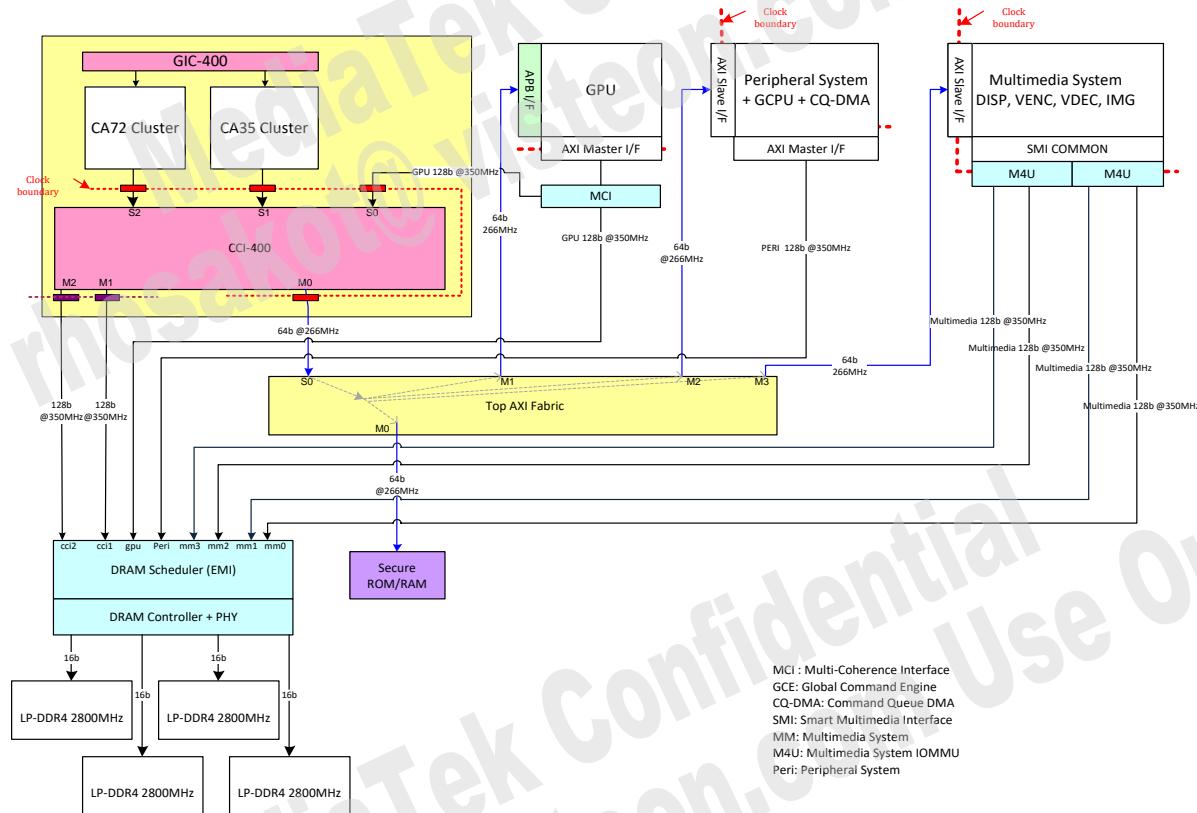


Figure 4-10 Infra Bus Fabric

4.5.3 Register Definition

For register details, please refer to Chapter 2.5 of “MT2712 IVI Application Processor Registers”.

4.6 On-chip Memory Controller

4.6.1 Introduction

The On-chip Memory Controller provides 96 KB boot Read-only Memory (ROM) and 192 KB Static Random-Access Memory (SRAM) resources. Table 4-7 shows the memory map of boot ROM and on-chip SRAM.

Table 4-7 On-chip Memory Controller Memory Map

Bank	Start Address	End Address	Size	Device
0	0x0000_0000	0x0001_7FFF	96 KB	Boot ROM
	0x0010_0000	0x0012_FFFF	192 KB	On-chip SRAM
	0x0800_0000	0x0800_000F	16 B	Chip ID/HW, SW version

4.6.2 Block Diagram

The On-chip Memory Controller consists of a SRAM controller, a ROM controller, an AXI-FPC bus bridge, a bus interface unit, a setting register via APB bus bridge and a chip ID unit (see Figure 4-11).

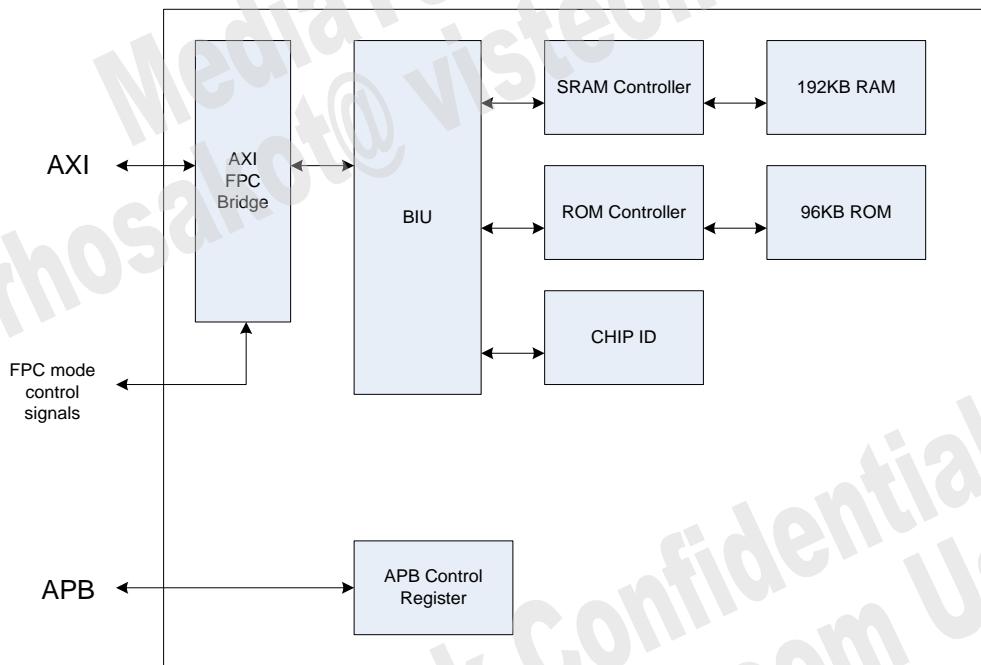


Figure 4-11 On-chip Memory Controller Block Diagram

4.6.3 Boot ROM Power-Down Mode

Boot ROM power-down mode is used in the following scenarios:

- After system boot, boot ROM will be powered down and prevented from any probe of ROM content
- In Multi-core-deep-idle (MCDI), it is the bootstrap for suspending/resuming CPU

4.6.3.1 Boot ROM FPC Mode

Boot ROM FPC mode is mainly used in Function Pattern mode. When the chip is trapped in FPC mode, the AXI-FPC bridge will block all the transactions to ROM address by returning a far jump instruction with jump address specified in SRAMROM_FPC_BOOT_ADDR. The default value of SRAMROM_FPC_BOOT_ADDR is 0x00000000. The AXI-FPC bridge will automatically unblock the transaction when the FPC program is downloaded to SRAM memory address space.

4.6.4 Register Definition

For register details, please refer to Chapter 2.6 of "MT2712 IVI Application Processor Registers".

4.7 Application Processor DMA (AP_DMA)

4.7.1 Introduction

The purpose of AP_DMA is to perform data transfer between memory and peripherals.

4.7.2 Features

AP_DMA has the following features.

- Support up to 18 channels of simultaneous data transfers
- Comply with the system bus (AXI)
- A data FIFO of 128 bits is embedded in DMA channel
- Support peripherals and channels
 - UART x 6 (TX/RX channels are separated, so there are 12 channels in total)
 - I2C x 6
 - Using signal and hand-shaking signal from a Peripheral is a method to trigger the DMA transmission
- Source/Destination configuration
 - Only one side (either source or destination) can be programmable and the other side could be selected as the specified peripheral
- Burst size/Burst length
 - 8 Bytes/1 Beat
- Support trust-zone
 - The corresponding channel is treated as a secure channel, if SEC_EN(AP_DMA_I2Cx_SEC_EN , AP_DMA_UARTx_TX_SEC_EN or AP_DMA_UARTx_RX_SEC_EN)[0] = 1
 - If the channel is set as a secure one, it can issue secure requests, and its configuration registers can be accessed via secure masters only
- Interrupt notification
 - FIFO data are over/under a certain threshold
- Scheduling scheme
 - Round-robin. If many channels are triggered simultaneously, the triggering priority would depend on channel number. The channel with a smaller number has a higher priority. (e.g. channel 0 --> channel 1 --> channel 2 ...)

4.7.3 Block Diagram

The system level block diagram of AP_DMA in MT2712 is shown in Figure 4-12.

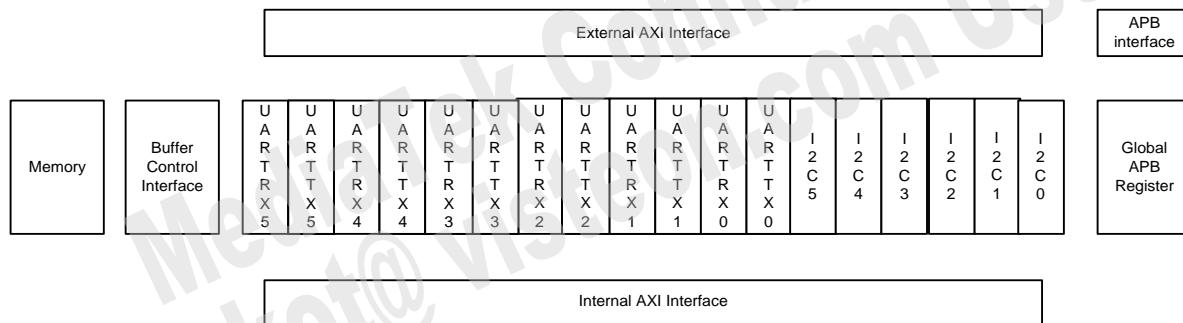


Figure 4-12 AP_DMA System Level Block Diagram

There are 18 channels in DMA. The external AXI interface is connected to the peripheral AXI bus fabric to provide external memory access ability. The internal AXI interface is also connected to the peripheral AXI bus fabric and re-directed to the related peripherals, e.g. I2C and UART. A memory block is used as a buffer which makes the transfer on the AXI bus interface more efficient. An APB interface is used to program registers for global registers and local registers existing in every individual DMA channel.

4.7.4 Register Definition

For register details, please refer to Chapter 2.7 of “MT2712 IVI Application Processor Registers”.

4.7.5 Programming Guide

4.7.5.1 I2C DMA Programming Guide

1. Configure DMA registers.

`AP_P_DMA_I2C_*_CON` (dir)
`AP_P_DMA_I2C_*_TX_MEM_ADDR`, `AP_P_DMA_I2C_*_TX_LEN` (Tx)
`AP_P_DMA_I2C_*_RX_MEM_ADDR`, `AP_P_DMA_I2C_*_RX_LEN` (Rx)

2. Set interrupt enable = 1.

`AP_P_DMA_I2C_*_INT_EN`

3. Wait for interrupt.

4. Clear interrupt flag.

- `AP_P_DMA_I2C_*_INT_FLAG`

4.7.5.2 UART-TX DMA Programming Guide

1. Configure registers.

`AP_P_DMA_UART_*_TX_VFF_ADDR`
`AP_P_DMA_UART_*_TX_VFF_LEN`, `AP_P_DMA_UART_*_TX_VFF_THRE`
`AP_P_DMA_UART_*_TX_VFF_WPT`

2. Write data to EMI and update SW write_pointer.
`AP_P_DMA_UART_*_TX_VFF_WPT`
3. Clear interrupt (repeat step 3-6 till finished).
`AP_P_DMA_UART_*_TX_INT_FLAG`
4. Set interrupt enable = 1.
`AP_P_DMA_UART_*_TX_INT_EN`
5. Set enable = 1 (first time).
`AP_P_DMA_UART_*_TX_EN`
6. Wait for interrupt.
7. Set stop = 1 if finished.
`AP_P_DMA_UART_*_STOP`

4.7.5.3 UART-RX DMA Programming Guide

1. Configure registers.
`AP_P_DMA_UART_*_RX_VFF_ADDR`
`AP_P_DMA_UART_*_RX_VFF_LEN, AP_P_DMA_UART_*_RX_VFF_THRE`
`AP_P_DMA_UART_*_RX_VFF_RPT`
`AP_P_DMA_UART_*_RX_FLOW_CTRL_THRE`
2. Set interrupt enable = 1.
`AP_P_DMA_UART_*_RX_INT_EN`
3. Set enable = 1 (first time).
`AP_P_DMA_UART_*_RX_EN`
4. Wait for interrupt (repeat step 4-6 till finished).
5. Read data from EMI; update SW read_pointer.
`AP_P_DMA_UART_*_RX_VFF_RPT`
6. Clear interrupt flag.
`AP_P_DMA_UART_*_RX_INT_FLAG`
7. Set stop = 1 if finished.
`AP_P_DMA_UART_*_RX_STOP`

4.8 Command Queue DMA (CQ_DMA)

4.8.1 Introduction

CQ_DMA is a general DMA which transfers data between DRAM and SRAM.

4.8.2 Features

There is only one general DMA engine in CQ_DMA.

- Comply with the system bus (AXI)
 - Support for unaligned data transfer (Byte alignment)
- A data FIFO of 16 x 8 Bytes is embedded in each DMA channel
- Source/Destination configuration
 - Both source and destination are programmable
- Support burst size
 - 8 Bytes
- Support burst length
 - 1~7 Beats

4.8.3 Block Diagram

Figure 4-13 shows the system level block diagram of CQ_DMA in MT2712.

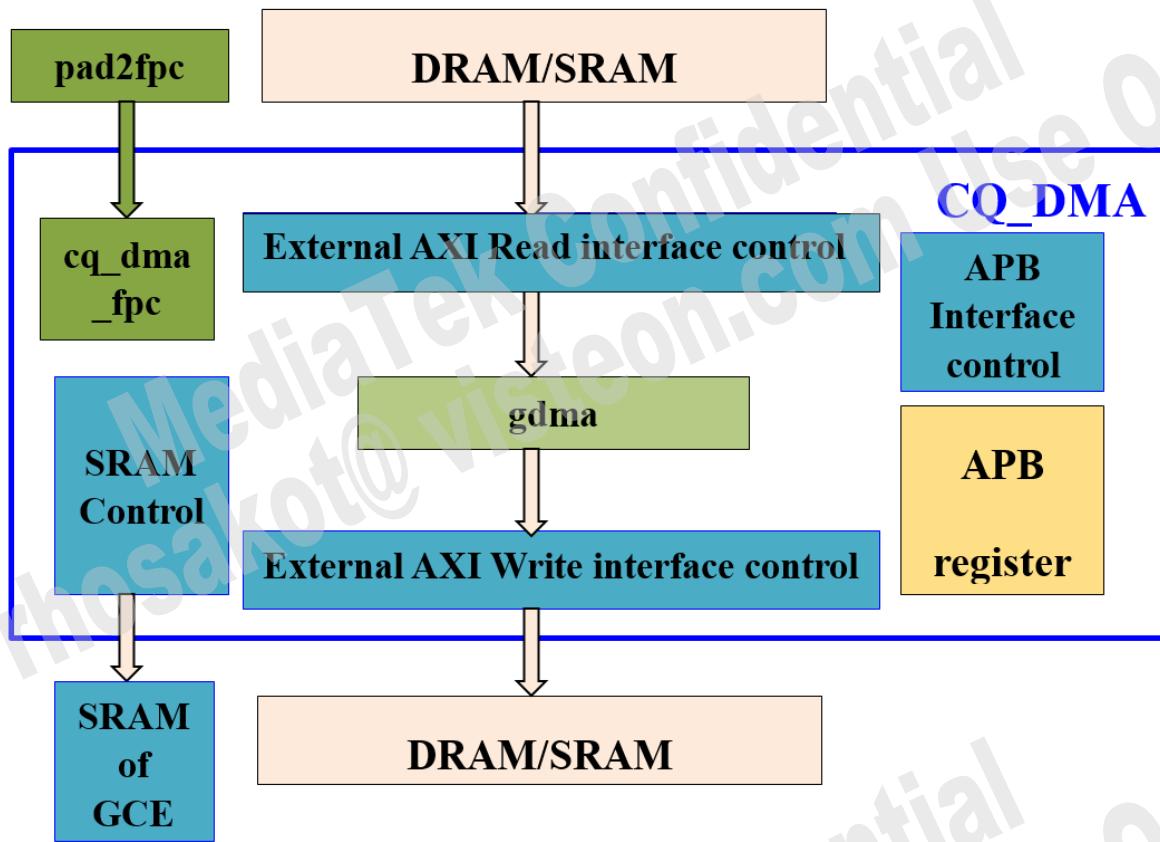


Figure 4-13 System Level Block Diagram of CQ_DMA

4.8.4 Register Definition

For register details, please refer to Chapter 2.8 of “MT2712 IVI Application Processor Registers”.

4.8.5 Programming Guide

1. Enable interrupt
 *`CQ_DMA_G_DMA_INT_EN` = 1
2. Configure DMA source address & destination address
 *`CQ_DMA_G_DMA_SRC_ADDR`
 *`CQ_DMA_G_DMA_DST_ADDR`
3. Configure DMA length
 *`CQ_DMA_G_DMA_LEN`
4. Enable DMA to trigger DMA engine
 *`CQ_DMA_G_DMA_EN` = 1
5. Wait for Interrupt & Clear
 *`CQ_DMA_G_DMA_INT_FLAG` = 0

4.9 CM4SYS

4.9.1 Introduction

CM4SYS is a tiny processor sub-system with Cortex-M4 processor and peripherals. This sub-system is designed to handle specific tasks for MT2712, such as Sensor HUB, UltraFastRVC (Ultra-Fast Rear View Camera) and other future tasks.

4.9.2 Features

The CM4 is a 32-bit micro-processor core with the following features:

- CortexM4 with FPU, MPU, NVIC
- AHB-lite bus interface
- JTAG embedded ICE

The CM4SYS subsys integrates the following features:

- Built-in 128 KB TCM
- Share memory with AP
- Share interrupt with AP
- 2 sets of semaphore register with AP
- 16 × 32 bits share register with AP
- Watch dog
- Timer 32 bit x1

4.9.3 Block Diagram

Figure 4-14 shows the block diagram of CM4SYS. The blocks are as follows:

- CM4: A 32-bit micro controller
- MEM Pool: CM4 code memory pool
- CFGREG: Configuration registers
- Bus fabric: AHB async, AHB arbiter, AHB2APB bridge, APB async, APB arbiter...

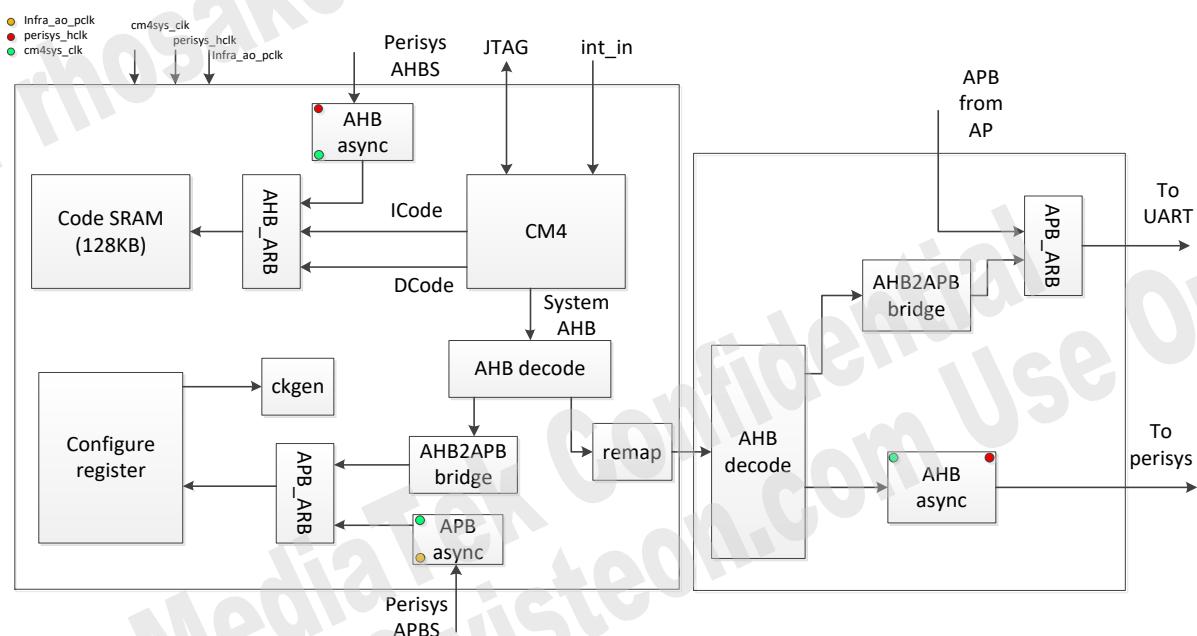


Figure 4-14 Block Diagram of CM4SYS

4.9.4 Programming Guide

4.9.4.1 CM4 Memory Mapping

CM4 memory mapping is shown as Figure 4-15.

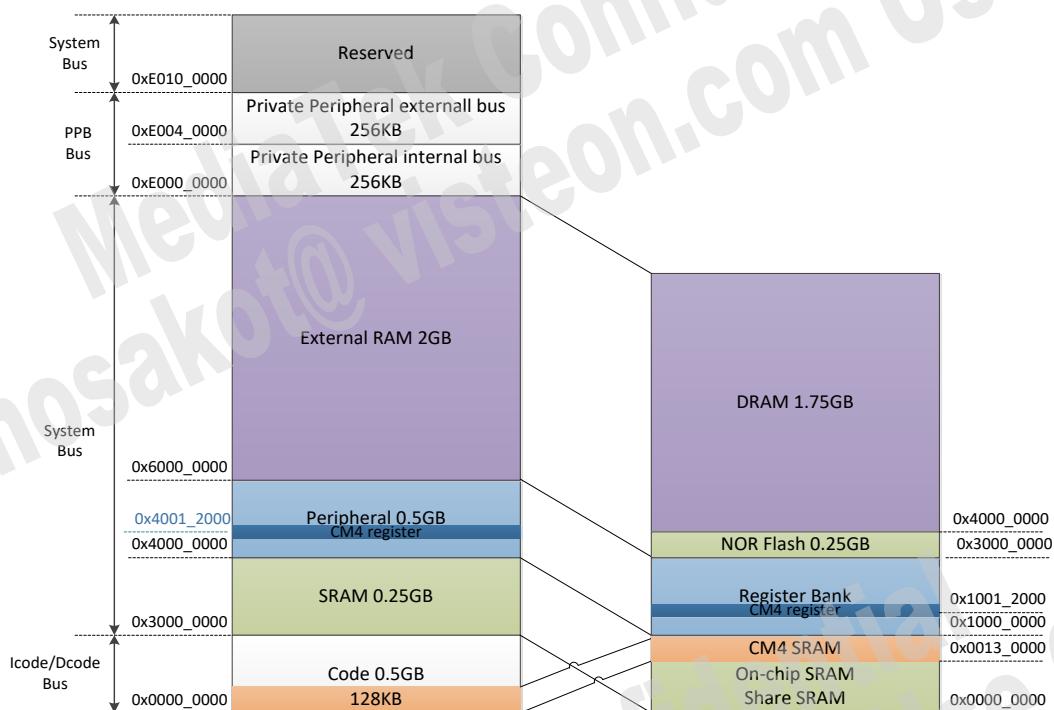


Figure 4-15 Memory Mapping of CM4SYS

4.9.4.2 HW Interrupt Table

Table 4-8 CM4 HW Interrupt

IRQ No.	Source(Module)	Note
218:0	Mapping to AP irq table IRQ_EXT ID[218:0]	-
219:220	reserved	-
221	eint[1]	-
222	scpsys	-
223	scp	-
224	apxgpt	-
225	Spi0	infra_ao domain
226	Spi1	infra_ao domain
227	uart	infra_ao domain
235:228	Reserved	-
236	cm4sys	Cm4 soft irq
237	cm4sys	Timer irq
238	cm4sys	Resv0 irq
239	cm4sys	Resv1 irq

4.9.4.3 CM4SYS Active Sequence

1. Switch clock source.
2. AP initializes CM4 code to CM4SYS MEM_Pool.
3. AP releases CM4 reset.
4. AP releases CM4 clock gate.

4.9.4.4 CM4SYS Idle Sequence

1. AP gates CM4 clock.
2. AP resets CM4.
3. AP releases clock source.

5 Peripherals

5.1 Memory Map

5.1.1 Introduction

This chapter introduces the memory map of MT2712.

5.1.2 Top Memory Map

Table 5-1 is the top level memory map of MT2712.

Table 5-1 Memory Map of MT2712

Bank	Start Address	End Address	Size	Devices
0	0x0000_0000	0x0001_7FFF	96 KB	Boot Section
	0x0001_8000	0x000B_FFFF	-	Reserved
	0x000C_0000	0x000F_FFFF	256 KB	Share SRAM
	0x0010_0000	0x0010_FFFF	64 KB	On-Chip SRAM
	0x0011_0000	0x0011_FFFF	64 KB	
	0x0012_0000	0x0012_FFFF	64 KB	
	0x0013_0000	0x0014_FFFF	128 KB	
	0x0020_0000	0x0023_FFFF	256 KB	SCPSYS
	0x0024_0000	0x07FF_FFFF	-	Reserved
	0x0800_0000	0x0800_000C	16 B	Chip ID/HW, SW Version
	0x0800_0010	0x0FFF_FFFF	-	Reserved
1	0x1000_0000	0x10FF_FFFF	16 MB	Infrastructure, Mixmode & MCU System
	0x1100_0000	0x11FF_FFFF	16 MB	Peripheral System
	0x1200_0000	0x12FF_FFFF	16 MB	Reserved
	0x1300_0000	0x13FF_FFFF	16 MB	MFG System
	0x1400_0000	0x14FF_FFFF	16 MB	MM System
	0x1500_0000	0x15FF_FFFF	16 MB	IMAGE System
	0x1600_0000	0x16FF_FFFF	16 MB	VDEC System
	0x1700_0000	0x17FF_FFFF	16 MB	Reserved
	0x1800_0000	0x18FF_FFFF	16 MB	VENC System
	0x1900_0000	0x19FF_FFFF	16 MB	JPGDEC
	0x1A00_0000	0x1FFF_FFFF	112 MB	Reserved
2	0x2000_0000	0x2FFF_FFFF	256 MB	PCIe Interface

Bank	Start Address	End Address	Size	Devices
3	0x3000_0000	0x3FFF_FFFF	256 MB	NOR Flash
4	0x4000_0000	0x4FFF_FFFF		
5	0x5000_0000	0x5FFF_FFFF		
6	0x6000_0000	0x6FFF_FFFF		
7	0x7000_0000	0x7FFF_FFFF		
8	0x8000_0000	0x8FFF_FFFF		
9	0x9000_0000	0x9FFF_FFFF		
A	0xA000_0000	0xAF00_FFFF		
B	0xB000_0000	0xBF00_FFFF		
C	0xC000_0000	0xCF00_FFFF		
D	0xD000_0000	0xDF00_FFFF		
E	0xE000_0000	0xEF00_FFFF		
F	0xF000_0000	0xFF00_FFFF		
10	0x1_0000_0000	0x1_0FFF_FFFF		
11	0x1_1000_0000	0x1_1FFF_FFFF		
12	0x1_2000_0000	0x1_2FFF_FFFF		
13	0x1_3000_0000	0x1_3FFF_FFFF		
14	0x1_4000_0000	0x1_4FFF_FFFF		
15	0x1_5000_0000	0x1_5FFF_FFFF		
16	0x1_6000_0000	0x1_6FFF_FFFF		
17	0x1_7000_0000	0x1_7FFF_FFFF		
18	0x1_8000_0000	0x1_8FFF_FFFF		
19	0x1_9000_0000	0x1_9FFF_FFFF		
1A	0x1_A000_0000	0x1_AFFF_FFFF		
1B	0x1_B000_0000	0x1_BFFF_FFFF		
1C	0x1_C000_0000	0x1_CFFF_FFFF		
1D	0x1_D000_0000	0x1_DFFF_FFFF		
1E	0x1_E000_0000	0x1_EFFF_FFFF		
1F	0x1_F000_0000	0x1_FFFF_FFFF		

5.1.3 Boot Section Memory Map

Table 5-2 Boot Section Memory Map of MT2712

Remap	Start Address	End Address	Size	Devices
0	0x0000_0000	0x0001_7FFF	96K B	Boot ROM

Remap	Start Address	End Address	Size	Devices
	0x0001_8000	0x000B_FFFF	672 KB	Reserved
1	0x0000_0000	0x0001_7FFF	96 KB	Reserved
1	0x0001_8000	0x000B_FFFF	672 KB	Reserved

5.1.4 Infrastructure System Memory Map

Table 5-3 Infrastructure System Memory Map of MT2712

Start Address	End Address	Size	Devices
0x1000_0000	0x1000_0FFF	4 KB	Top-level Clock Generator
0x1000_1000	0x1000_1FFF	4 KB	Infrasys Configuration Registers
0x1000_2000	0x1000_2FFF	4 KB	Reserved
0x1000_3000	0x1000_3FFF	4 KB	Perisys Configuration Registers
0x1000_4000	0x1000_4FFF	4 KB	Reserved
0x1000_5000	0x1000_5FFF	4 KB	GPIO Controller
0x1000_6000	0x1000_6FFF	4 KB	Top-level Sleep Manger
0x1000_7000	0x1000_7FFF	4 KB	Top-level Reset Generator
0x1000_8000	0x1000_8FFF	4 KB	GPT
0x1000_9000	0x1000_9FFF	4 KB	Reserved
0x1000_A000	0x1000_AFFF	4 KB	SEJ
0x1000_B000	0x1000_BFFF	4 KB	APMCU External Interrupt Controller
0x1000_C000	0x1000_CFFF	4 KB	sys_timer
0x1000_D000	0x1000_DFFF	4 KB	irrx
0x1000_E000	0x1000_EFFF	4 KB	device APC AO
0x1000_F000	0x1000_FFFF	4 KB	UART5
0x1001_0000	0x1001_0FFF	4 KB	Keypad Configuration Registers
0x1001_1000	0x1001_1FFF	4 KB	TOP RTC
0x1001_2000	0x1001_2FFF	4 KB	SPI4
0x1001_3000	0x1001_3FFF	4 KB	SPI1
0x1001_4000	0x1001_4FFF	4 KB	GPT2

Start Address	End Address	Size	Devices
0x1010_0000	0x1010_7FFF	32 KB	DRAMC CH0
0x1010_8000	0x1010_FFFF	32 KB	DRAMC CH1
0x1011_0000	0x1011_7FFF	32 KB	DRAMC CH2
0x1011_8000	0x1011_FFFF	32 KB	DRAMC CH3

Start Address	End Address	Size	Devices
0x1020_0000	0x1020_0FFF	4 KB	Reserved
0x1020_1000	0x1020_1FFF	4 KB	infrasys Control register
0x1020_2000	0x1020_2FFF	4 KB	BootROM/SysSRAM
0x1020_3000	0x1020_3FFF	4 KB	EMI Bus Interface
0x1020_4000	0x1020_4FFF	4 KB	System CIRQ
0x1020_5000	0x1020_5FFF	4 KB	MM IOMMU 1 configuration
0x1020_6000	0x1020_6FFF	4 KB	EFUSEC
0x1020_7000	0x1020_7FFF	4 KB	device_apc monitor module
0x1020_8000	0x1020_8FFF	4 KB	Bus Debug Tracker
0x1020_9000	0x1020_9FFF	4 KB	AP Mixed Control Register
0x1020_A000	0x1020_AFFF	4 KB	MM IOMMU 2 configuration
0x1020_B000	0x1020_BFFF	4 KB	ANA_MIPI_DSI3
0x1020_C000	0x1020_CFFF	4 KB	GPIO1 Controller
0x1020_D000	0x1020_DFFF	4 KB	Infrasys Mbist Control Register
0x1020_E000	0x1020_EFFF	4 KB	EMI MPU Control
0x1020_F000	0x1020_FFFF	4 KB	TRNG
0x1021_0000	0x1021_0FFF	4 KB	GCPU
0x1021_1000	0x1021_1FFF	4 KB	GCPU
0x1021_2000	0x1021_2FFF	4 KB	CQ-DMA
0x1021_3000	0x1021_3FFF	4 KB	gcpu_m4u
0x1021_4000	0x1021_4FFF	4 KB	ANA_MIPI_DSI2
0x1021_5000	0x1021_5FFF	4 KB	ANA_MIPI_DSI0
0x1021_6000	0x1021_6FFF	4 KB	ANA_MIPI_DSI1
0x1021_7000	0x1021_7FFF	4 KB	ANA_MIPI_CSIO
0x1021_8000	0x1021_8FFF	4 KB	ANA_MIPI_CS1
0x1021_9000	0x1021_9FFF	4 KB	Reserved
0x1022_0000	0x1022_3FFF	16 KB	MCUSYS Configure Register
Debug system			
0x1040_0000	0x10FF_FFFF	12 MB	CoreSight
MCU system			
0x1039_0000	0x1039_FFFF	64 KB	CCI-400 Configuration Register
0x1051_0000	0x1057_FFFF	64 KB	GIC400

5.1.5 Peripheral System Memory Map

Table 5-4 Peripheral System Map of MT2712

Start Address	End Address	Size	Module Description
0x1100_0000	0x1100_0FFF	4 KB	DMA

Start Address	End Address	Size	Module Description
0x1100_1000	0x1100_1FFF	4 KB	AUXADC
0x1100_2000	0x1100_2FFF	4 KB	UART0
0x1100_3000	0x1100_3FFF	4 KB	UART1
0x1100_4000	0x1100_4FFF	4 KB	UART2
0x1100_5000	0x1100_5FFF	4 KB	UART3
0x1100_6000	0x1100_6FFF	4 KB	PWM
0x1100_7000	0x1100_7FFF	4 KB	I2C0
0x1100_8000	0x1100_8FFF	4 KB	I2C1
0x1100_9000	0x1100_9FFF	4 KB	I2C2
0x1100_A000	0x1100_AFFF	4 KB	SPI0
0x1100_B000	0x1100_BFFF	4 KB	THERM_CTRL
0x1100_C000	0x1100_CFFF	4 KB	Reserved
0x1100_D000	0x1100_DFFF	4 KB	SPI_NOR
0x1100_E000	0x1100_EFFF	4 KB	NFI
0x1100_F000	0x1100_FFFF	4 KB	NFI_ECC
0x1101_0000	0x1101_0FFF	4 KB	I2C3
0x1101_1000	0x1101_1FFF	4 KB	I2C4
0x1101_2000	0x1101_2FFF	4 KB	Rev
0x1101_3000	0x1101_3FFF	4 KB	I2C5
0x1101_4000	0x1101_4FFF	4 KB	Reserved
0x1101_5000	0x1100_5FFF	4 KB	SPI2
0x1101_6000	0x1100_6FFF	4 KB	SPI3
0x1101_7000	0x1100_7FFF	4 KB	Reserved
0x1101_8000	0x1100_8FFF	4 KB	SPI5
0x1101_9000	0x1101_9FFF	4 KB	UART4
0x1101_A000	0x1100_AFFF	4 KB	Reserved
0x1101_B000	0x1101_EFFF	16 KB	GMAC
0x1120_0000	0x1120_FFFF	64 KB	Reserved
0x1121_0000	0x1121_FFFF	64 KB	Reserved
0x1122_0000	0x1122_FFFF	64 KB	Audio
0x1123_0000	0x1123_FFFF	64 KB	MSDC0
0x1124_0000	0x1124_FFFF	64 KB	MSDC1
0x1125_0000	0x1125_FFFF	64 KB	MSDC2
0x1126_0000	0x1126_FFFF	64 KB	MSDC3
0x1127_0000	0x1127_FFFF	64 KB	SSUSB_CSR_SLV
0x1128_0000	0x1128_FFFF	64 KB	SSUSB_SIF_SLV
0x1129_0000	0x1129_FFFF	64 KB	SSUSB_MI
0x112A_0000	0x112B_FFFF	128 KB	Audio
0x112C_0000	0x112C_FFFF	64 KB	SSUSB_CSR_SLV

Start Address	End Address	Size	Module Description
0x112D_0000	0x112D_FFFF	64 KB	SSUSB_SIF_SLV
0x112E_0000	0x112E_FFFF	64 KB	SSUSB_MI
0x112F_F000	0x112F_FFFF	4 KB	PCIe port1 control register space
0x1130_0000	0x116F_FFFF	4 MB	PCIe port1 MMIO space
0x1170_0000	0x1170_0FFF	4 KB	PCIe port0 control register space
0x2000_0000	0x2FFF_FFFF	256 MB	PCIe port0 MMIO space

5.1.6 Multimedia System Memory Map

Table 5-5 Audio System Memory Map

Start Address	End Address	Size	Devices
0x1122_0000	0x1122_FFFF	64 KB	Audio system configuration

Table 5-6 MFG System Memory Map

Start Address	End Address	Size	Devices
0x13FF_F000	0x13FF_FFFF	4 KB	MFG configuration

Table 5-7 MMSYS System Memory Map

Start Address	End Address	Size	Devices
0x1400_0000	0x1400_0FFF	4 KB	MMSYS_CONFIG
0x1400_1000	0x1400_1FFF	4 KB	MDP_RDMA0
0x1400_2000	0x1400_2FFF	4 KB	MDP_RDMA1
0x1400_3000	0x1400_3FFF	4 KB	MDP_RSZ0
0x1400_4000	0x1400_4FFF	4 KB	MDP_RSZ1
0x1400_5000	0x1400_5FFF	4 KB	MDP_RSZ2
0x1400_6000	0x1400_6FFF	4 KB	MDP_WDMA
0x1400_7000	0x1400_7FFF	4 KB	MDP_WROTO
0x1400_8000	0x1400_8FFF	4 KB	MDP_WROT1
0x1400_9000	0x1400_9FFF	4 KB	MDP_TDSHPO
0x1400_A000	0x1400_AFFF	4 KB	MDP_TDSHP1
0x1400_B000	0x1400_BFFF	4 KB	Reserved
0x1400_C000	0x1400_CFFF	4 KB	DISP_OVLO
0x1400_D000	0x1400_DFFF	4 KB	DISP_OVL1
0x1400_E000	0x1400_EFFF	4 KB	DISP_RDMA0
0x1400_F000	0x1400_FFFF	4 KB	DISP_RDMA1
0x1401_0000	0x1401_0FFF	4 KB	DISP_RDMA2
0x1401_1000	0x1401_1FFF	4 KB	DISP_WDMA0

Start Address	End Address	Size	Devices
0x1401_2000	0x1401_2FFF	4 KB	DISP_WDMA1
0x1401_3000	0x1401_3FFF	4 KB	DISP_COLOR0
0x1401_4000	0x1401_4FFF	4 KB	DISP_COLOR1
0x1401_5000	0x1401_5FFF	4 KB	DISP_AAL
0x1401_6000	0x1401_6FFF	4 KB	DISP_GAMMA
0x1401_7000	0x1401_7FFF	4 KB	Reserved
0x1401_8000	0x1401_8FFF	4 KB	DISP_SPLITO
0x1401_9000	0x1401_9FFF	4 KB	Reserved
0x1401_A000	0x1401_AFFF	4 KB	DISP_UFOE
0x1401_B000	0x1401_BFFF	4 KB	DSI0
0x1401_C000	0x1401_CFFF	4 KB	DSI1
0x1401_D000	0x1401_DFFF	4 KB	DPI
0x1401_E000	0x1401_EFFF	4 KB	DISP_PWM0
0x1401_F000	0x1401_FFFF	4 KB	DISP_PWM1
0x1402_0000	0x1402_0FFF	4 KB	MM_MUTEX
0x1402_1000	0x1402_1FFF	4 KB	SMI_LARBO
0x1402_2000	0x1402_2FFF	4 KB	SMI_COMMON
0x1402_3000	0x1402_3FFF	4 KB	DISP_OD
0x1402_4000	0x1402_4FFF	4 KB	DPI1
0x1402_5000	0x1402_5FFF	4 KB	Reserved
0x1402_6000	0x1402_6FFF	4 KB	LVDS
0x1402_7000	0x1402_7FFF	4 KB	SMI_LARB4
0x1402_8000	0x1402_8FFF	4 KB	MDP_RDMA2
0x1402_9000	0x1402_9FFF	4 KB	DISP_COLOR2
0x1402_A000	0x1402_AFFF	4 KB	DISP_AAL1
0x1402_B000	0x1402_BFFF	4 KB	DISP_OD1
0x1402_C000	0x1402_CFFF	4 KB	DISP_OVL2
0x1402_D000	0x1402_DFFF	4 KB	DISP_WDMA2
0x1402_E000	0x1402_EFFF	4 KB	LVDS1
0x1402_F000	0x1402_FFFF	4 KB	MDP_TDSHP2
0x1403_0000	0x1403_0FFF	4 KB	SMI_LARB5
0x1403_1000	0x1403_1FFF	4 KB	SMI_COMMON1
0x1403_2000	0x1403_2FFF	4 KB	SMI_LARB7
0x1403_3000	0x1403_3FFF	4 KB	MDP_RDMA3
0x1403_4000	0x1403_4FFF	4 KB	MDP_WROT2
0x1403_5000	0x1403_5FFF	4 KB	DSI2
0x1403_6000	0x1403_6FFF	4 KB	DSI3
0x1403_7000	0x1403_7FFF	4 KB	Reserved
0x1403_8000	0x1403_8FFF	4 KB	DISP_MONITOR0

Start Address	End Address	Size	Devices
0x1403_9000	0x1403_9FFF	4 KB	DISP_MONITOR1
0x1403_A000	0x1403_AFFF	4 KB	DISP_MONITOR2
0x1403_B000	0x1403_BFFF	4 KB	DISP_MONITOR3
0x1403_C000	0x1403_CFFF	4 KB	DISP_PWM2

Table 5-8 Image System Memory Map

Start Address	End Address	Size	Devices
0x1500_0000	0x1500_0FFF	4 KB	IMGSYS_CONFIG
0x1500_1000	0x1500_1FFF	4 KB	SMI_LARB2
0x1500_2000	0x1500_2FFF	4 KB	SENINF_TOP0
0x1500_3000	0x1500_3FFF	4 KB	SENINF_TOP1
0x1500_4000	0x1500_4FFF	4 KB	CAMSV_TOP0
0x1500_5000	0x1500_5FFF	4 KB	CAMSV_TOP1
0x1500_6000	0x1500_6FFF	4 KB	CAMSV_TOP2
0x1500_7000	0x1500_7FFF	4 KB	CAMSV_TOP3
0x1500_8000	0x1500_8FFF	4 KB	CAMSV_TOP4
0x1500_9000	0x1500_9FFF	4 KB	CAMSV_TOP5
0x1500_A000	0x1500_AFFF	4 KB	Reserved
0x1500_B000	0x1500_BFFF	4 KB	Reserved
0x1500_C000	0x1500_CFFF	4 KB	Reserved
0x1500_D000	0x1500_DFFF	4 KB	Reserved
0x1500_E000	0x1500_EFFF	4 KB	Reserved
0x1500_F000	0x1500_FFFF	4 KB	Reserved
0x1501_0000	0x1501_0FFF	4 KB	BDP_DISPSYS_CONFIG
0x1501_1000	0x1501_1FFF	4 KB	BDP_DISPFORMAT
0x1501_2000	0x1501_2FFF	4 KB	BDP_VDO
0x1501_3000	0x1501_3FFF	4 KB	BDP_NR
0x1501_4000	0x1501_4FFF	4 KB	BDP_NR2
0x1501_5000	0x1501_5FFF	4 KB	BDP_TVD
0x1501_6000	0x1501_6FFF	4 KB	BDP_WR_CHANNEL_DI
0x1501_7000	0x1501_7FFF	4 KB	BDP_WR_CHANNEL_VDI
0x1501_8000	0x1501_8FFF	4 KB	BDP_LARB
0x1501_9000	0x1501_9FFF	4 KB	BDP_LARB_RT
0x1501_A000	0x1501_AFFF	4 KB	BDP_DRAM2AXI_BRIDGE

Table 5-9 Video Decoding System Memory Map

Start Address	End Address	Size	Devices
0x1600_0000	0x 1600_FFFF	64 KB	System configuration
0x 1601_0000	0x 1601_FFFF	64 KB	SMI_LARB1
0x 1602_0000	0x 1602_FFFF	64 KB	Video decoder
0x 1603_0000	0x 1603_FFFF	64 KB	Image Resizer
0x 1604_0000	0x 1604_FFFF	64 KB	Video decoder MBIST control

Table 5-10 Video Encoding System Memory Map

Start Address	End Address	Size	Devices
0x 1800_0000	0x 1800_0FFF	4 KB	System configuration
0x 1800_1000	0x 1800_1FFF	4 KB	SMI_LARB3
0x 1800_2000	0x 1800_2FFF	4 KB	SMI_LARB6
0x 1800_3000	0x 1800_3FFF	4 KB	smi_sub_common
0x 1800_4000	0x 1800_4FFF	4 KB	Video encoder

Table 5-11 JPEGDEC System Memory Map

Start Address	End Address	Size	Devices
0x 1900_0000	0x 1900_0FFF	4 KB	System configuration
0x 1900_1000	0x 1900_1FFF	4 KB	JPG DEC
0x 1900_2000	0x 1900_2FFF	4 KB	JPG DEC

5.1.7 Memory Map for 4GB DRAM Support

MT2712 supports up to 4 GB DRAM memory space by enabling 4 GB mode. The memory allocation in legacy mode is shown in Figure 5-1. The areas 0x00000000 to 0x3FFFFFF are for HW registers and internal SRAM. From 0x40000000 to 0xFFFFFFFF, there is 3 GB of DRAM space.

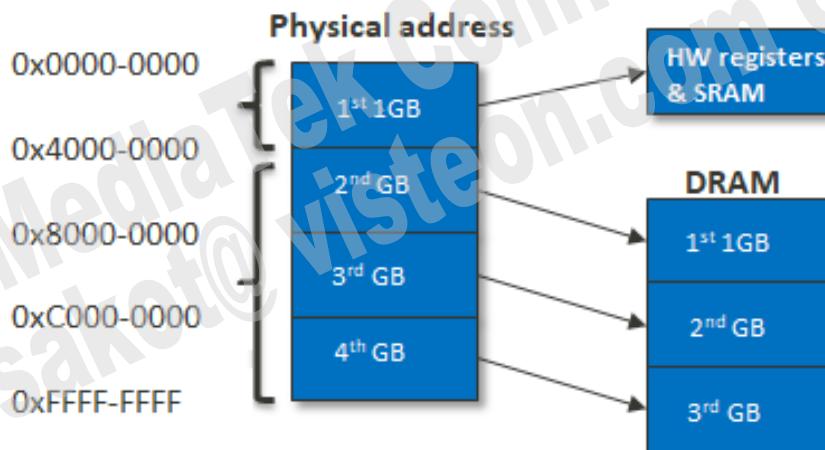


Figure 5-1 DRAM Physical Address Mapping in Legacy Mode

- As shown in Figure 5-2, 33-bit address is used in 4 GB mode for over 4 GB memory space allocation.
- The 1st 1 GB area is the same as that in legacy mode for HW registers and internal SRAM.
- Full 4 GB DRAM memory space is allocated to 0x100000000 to 0x13FFFFFF. The areas 0x040000000 to 0x0FFFFFFF are reserved in 4 GB mode. However, 0x40000000 to 0xFFFFFFF also map to 2nd GB to 4th GB DRAM space accordingly.

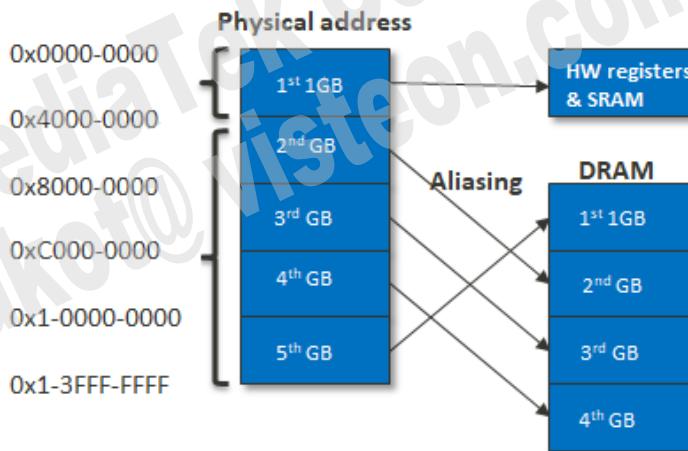


Figure 5-2 DRAM Physical Address Mapping in 4GB Mode

5.2 GPIO Controller

5.2.1 Introduction

MT2712 offers 210 general-purpose I/O (GPIO) pins. By setting up the control registers, the MCU software can control the GPIO direction and output value, and read the input values on these pins. The GPIOs are multiplexed with other functions to reduce the pin count.

5.2.2 Features

GPIO supports multi-function selection, which is controlled by GPIO mode registers.

- Normal multi-function

5.2.3 Block Diagram

Figure 5-3 shows the block diagram of GPIO controller in MT2712.

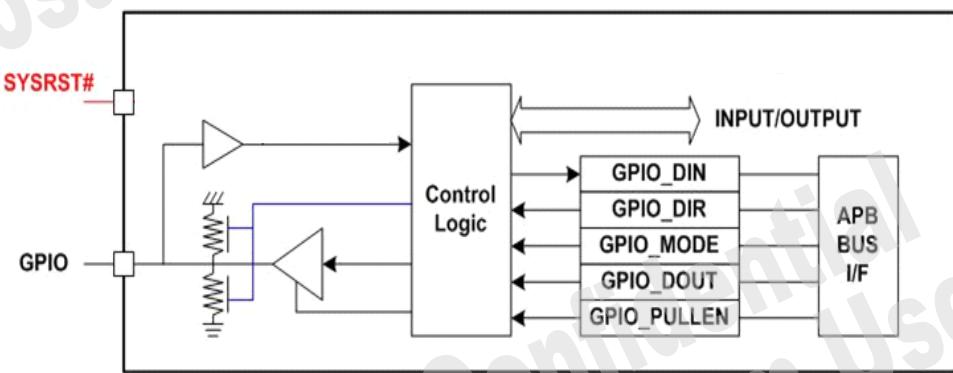
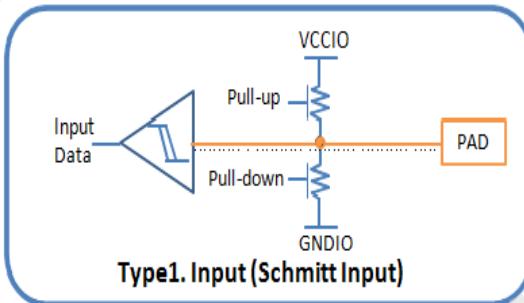


Figure 5-3 MT2712 GPIO Controller

Each GPIO controls the auxiliary mode by programming GPIO_MODE command register. Besides, the dedicated register bits can be set to 1 or 0 by writing the bits of GPIO_MODE_SETx or GPIO_MODE_CLRx to 1. GPIO_DIR, GPIO_DOUT and GPIO_PULLEN are also programmable by the same method as that for GPIO_MODE.

Figure 5-4 shows the IO types.



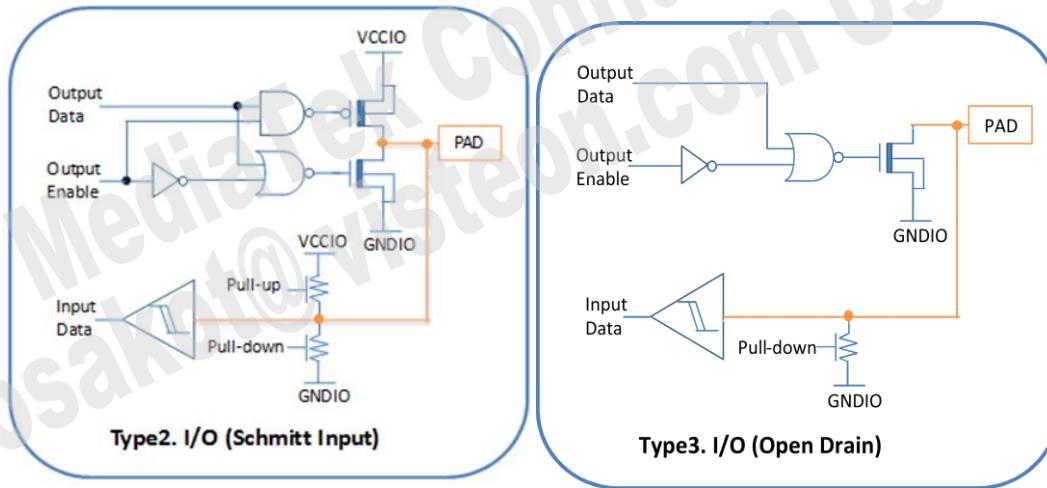


Figure 5-4 IO Types

5.2.4 GPIO AC Timing

5.2.4.1 I/O DC Parameters

Permanent device damage may occur if the absolute maximum ratings are exceeded. Functional operations under these conditions or at any other condition beyond those indicated in the operational sections are not implied. Exposure to the absolute maximum rating conditions for extended periods may affect the device's reliability.

This section includes the DC parameters of the following I/O types:

GPIO/I2C

MSDC3

RGMII

NOR

Schmitt input IO

5.2.4.1.1 General Purpose I/O (GPIO) DC Parameters

Table 5-12 shows DC parameters for only 1.8 V GPIO.

Table 5-12 1.8V GPIO Pads DC Parameters

Parameter	Description	Min.	Typ.	Max.	Unit	Note
VCC18IO	Supply voltage of IO power	1.7	1.8	1.9	V	-
V _{IH}	Input logic high voltage	0.65*VCC18IO	-	VCC18IO+0.3	V	1
V _{IL}	Input logic low voltage	-0.3	-	0.35*VCC18IO	V	1
V _{OH} (DC)	DC Output logic high voltage	0.75*VCC18IO	-	-	V	-
V _{OL} (DC)	DC Output logic low voltage	-	-	0.25*VCC18IO	V	-
R _{Pu}	Input pull-up resistance	40	75	190	KΩ	Control PU=1, PD=0

Parameter	Description	Min.	Typ.	Max.	Unit	Note
R _{pd}	Input pull-down resistance	40	75	190	kΩ	Control PU=0, PD=1
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC18IO)	-5	-	5	μA	2
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	2

Note:

1. Please specify the different types of input logic level for different types of pins.
2. Leakage sensitive applications (RTC, Crystal Pads) sensitive to this specification. Co-simulation is needed.

Table 5-13 shows DC parameters for only 1.8 V I2C GPIO.

Table 5-13 1.8V I2C GPIO Pads DC Parameters

Parameter	Description	Min.	Typ.	Max.	Unit	Note
VCC18IO	Supply voltage of IO power	1.7	1.8	1.9	V	-
V _{IH}	Input logic high voltage	0.65*VCC18IO	-	5.5	V	1
V _{IL}	Input logic low voltage	-0.3	-	0.35*VCC18IO	V	1
V _{OLO} (DC)	DC Output logic low voltage	-	-	0.2*VCC18IO	V	-
I _{OL} (DC)	DC Output logic low current	3	-	-	mA	-
R _{pd}	Input pull-down resistance	40	75	350	kΩ	Control PD = 1
Repull-up	External Pull-up resistance	-	4.7	-	kΩ	-
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC18IO)	-5	-	5	μA	2
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	2

Note:

1. Please specify different types of input logic level for different types of pins.
2. Leakage sensitive applications (RTC, crystal pads) sensitive to this specification. Co-simulation is needed.

Table 5-14 shows DC parameters for 18OD33V GPIO pads.

Table 5-14 18OD33V GPIO Pads DC Parameters

DC Operating Conditions of 3.3V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	2.97	3.3	3.63	V	-
V _{OH} (DC)	DC Output logic high voltage	VCC33IO-0.4	-	VCC33IO+0.3	V	
V _{OL} (DC)	DC Output logic low voltage	-0.3	-	0.4	V	
V _{IH}	Input logic high voltage	2.0	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.8	V	-
R _{pu}	Input pull-up resistance	40	75	190	kΩ	Control PU=1, PD=0
R _{pd}	Input pull-down resistance	40	75	190	kΩ	Control PD=1, PU=0

DC Operating Conditions of 3.3V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
I _{IN}	Input leakage current (any input $0V < V_{IN} < VCC33IO$)	-5	-	5	μA	2
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	2
DC Operating Conditions of 1.8V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	1.7	1.8	1.9	V	1
V _{OH} (DC)	DC Output logic high voltage	VCC33IO-0.2	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC Output logic low voltage	-0.3	-	0.2	V	-
V _{IH}	Input logic high voltage	1.27	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.58	V	-
R _{pu}	Input pull-up resistance	10	50	100	KΩ	Control PU=1, PD=0
R _{pd}	Input pull-down resistance	10	50	100	KΩ	Control PD=1, PU=0
I _{IN}	Input leakage current (any input $0V < V_{IN} < VCC33IO$)	-5	-	5	μA	2
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	2

Note:

- 1.8 V corner is 1.7~1.9 V. The voltage < 1.7 V (1.62 V, for example) is not acceptable in the circuit.
- Leakage sensitive applications (RTC, crystal pads) sensitive to this specification. Co-simulation is needed.

5.2.4.1.2 MSDC I/O DC Parameters

Table 5-15 shows the MSDC 1.8V IO DC timing.

Table 5-15 MSDC 1.8V IO DC Timing

Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC18IO	Supply voltage of IO power	1.7	1.8	1.9	V	-
V _{IH}	Input logic high voltage	1.3	-	2.0	V	1
V _{IL}	Input logic low voltage	-0.3	-	0.58	V	1
R _{pu}	Input pull-up resistance	-	-	-	KΩ	See details in register map
R _{pd}	Input pull-down resistance	-	-	-	KΩ	See details in register map
V _{OH} (DC)	DC output logic high voltage	1.4	-	-	V	-
V _{OL} (DC)	DC output logic low voltage	-	-	0.45	V	-
I _{IN}	Input leakage current (any input $0V < V_{IN} < VCC18IO$)	-2	-	2	μA	2
I _{OZ}	Tri-state output leakage current	-2	-	2	μA	2

Note:

1. Please specify different types of input logic level for different types of pins.
2. Leakage sensitive applications (RTC, Crystal Pads) sensitive to this specification. Co-simulation is needed.

Table 5-16 shows the MSDC 180D33V IO DC timing.

Table 5-16 MSDC 180D33V IO DC Timing

DC Operating Conditions of 3.3V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	2.97	3.3	3.63	V	-
V _{OH} (DC)	DC Output logic high voltage	0.75*VCC33IO	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC Output logic low voltage	-0.3	-	0.125*VCC33IO	V	-
V _{IH}	Input logic high voltage	0.625*VCC33I O	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.25*VCC33IO	V	-
R _{pu}	Input pull-up resistance	-	-	-	KΩ	See details in register map
R _{pd}	Input pull-down resistance	-	-	-	KΩ	See details in register map
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC33IO)	-5	-	5	μA	1
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	1
DC Operating Conditions of 1.8V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	1.7	1.8	1.9	V	-
V _{OH} (DC)	DC output logic high voltage	1.4	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC output logic low voltage	-0.3	-	0.45	V	-
V _{IH}	Input logic high voltage	1.27	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.58	V	-
R _{pu}	Input pull-up resistance	-	-	-	KΩ	See details in register map
R _{pd}	Input pull-down resistance	-	-	-	KΩ	See details in register map
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC33IO)	-5	-	5	μA	1
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	1

Note: Leakage sensitive applications (RTC, Crystal Pads) sensitive to this specification. Co-simulation is needed.

5.2.4.1.3 RGMII I/O DC Parameters

Table 5-17 is shown below.

Table 5-17 RGMII I/O Timing

DC Operating Conditions of 3.3V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	2.97	3.3	3.63	V	-
V _{OH} (DC)	DC output logic high voltage	VCC33IO-0.4	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC output logic low voltage	-0.3	-	0.4	V	-
V _{IH}	Input logic high voltage	2.0	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.8	V	-
R _{pu}	Input pull-up resistance	40	75	190	KΩ	Control PU=1, PD=0
R _{pd}	Input pull-down resistance	40	75	190	KΩ	Control PD=1, PU=0
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC33IO)	-5	-	5	μA	2
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	2
DC Operating Conditions of 1.8V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	1.7	1.8	1.9	V	1
V _{OH} (DC)	DC output logic high voltage	VCC33IO-0.2	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC output logic low voltage	-0.3	-	0.2	V	-
V _{IH}	Input logic high voltage	1.27	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.58	V	-
R _{pu}	Input pull-up resistance	10	50	100	KΩ	Control PU=1, PD=0
R _{pd}	Input pull-down resistance	10	50	100	KΩ	Control PD=1, PU=0
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC33IO)	-5	-	5	μA	2
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	2

Note:

- 1.8 V corner is 1.7~1.9 V. Voltage <1.7 V (1.62 V for example) is not acceptable in the circuit.
- Leakage sensitive applications (RTC, Crystal Pads) sensitive to this specification. Co-simulation is needed.

5.2.4.1.4 NOR I/O DC Parameters

Table 5-18 is shown below.

Table 5-18 NOR Flash I/O DC Timing

DC Operating Conditions of 3.3V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of IO power	2.97	3.3	3.63	V	-

DC Operating Conditions of 3.3V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
V _{OH} (DC)	DC output logic high voltage	VCC33IO-0.4V	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC output logic low voltage	-0.3	-	0.4	V	-
V _{IH}	Input logic high voltage	2.0	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.8	V	-
R _{pu}	Input pull-up resistance	40	75	190	KΩ	Control PU=1, PD=0
R _{pd}	Input pull-down resistance	40	75	190	KΩ	Control PD=1, PU=0
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC33IO))	-5	-	5	μA	2
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	2
DC Operating Conditions of 1.8V Applications						
Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC33IO	Supply voltage of SIM IO power	1.7	1.8	1.9	V	
V _{OH} (DC)	DC output logic high voltage	VCC33IO-0.2	-	VCC33IO+0.3	V	-
V _{OL} (DC)	DC output logic low voltage	-0.3	-	0.2	V	-
V _{IH}	Input logic high voltage	1.27	-	VCC33IO+0.3	V	-
V _{IL}	Input logic low voltage	-0.3	-	0.58	V	-
R _{pu}	Input pull-up resistance	10	50	100	KΩ	Control PU=1, PD=0
R _{pd}	Input pull-down resistance	10	50	100	KΩ	Control PD=1, PU=0
I _{IN}	Input leakage current (any input 0V < V _{IN} < VCC33IO))	-5	-	5	μA	2
I _{OZ}	Tri-state output leakage current	-5	-	5	μA	2

Note:

- 1.8V corner is 1.7~1.9V. Voltage <1.7V (1.62V, for example) is not acceptable in the circuit.
- Leakage sensitive applications (RTC, Crystal Pads) sensitive to this specification. Co-simulation is needed.

5.2.4.1.5 Schmitt Input IO

Table 5-19 shows the Schmitt input IO DC parameters.

Table 5-19 Schmitt Input I/O DC Timing

Parameter	Descriptions	Min.	Typ.	Max.	Unit	Note
VCC18IO	Supply voltage of IO power	1.7	1.8	1.9	V	-
V _{IH}	Input logic high voltage	0.65*VCC18IO	-	VCC18IO+0.3	V	1
V _{IL}	Input logic low voltage	-0.3	-	0.35*VCC18IO	V	1
Parameter	Descriptions	VDDIO	-40°C~125°C		Unit	Note
			Min.	Max.		
VT+		1.62	0.52*VCC18IO	0.65*VCC18IO	V	-

	Positive-going input the threshold voltage	1.8	0.53*VCC18IO	0.65*VCC18IO	V	-
		1.98	0.53*VCC18IO	0.65*VCC18IO	V	-
VT-	Negative-going input the threshold voltage	1.62	0.35*VCC18IO	0.46*VCC18IO	V	-
		1.8	0.35*VCC18IO	0.47*VCC18IO	V	-
		1.98	0.35*VCC18IO	0.48*VCC18IO	V	-
ΔVT	Hysteresis $VT+ - VT-$	1.62	0.15*VCC18IO	0.3*VCC18IO	V	-
		1.8	0.15*VCC18IO	0.3*VCC18IO	V	-
		1.98	0.15*VCC18IO	0.3*VCC18IO	V	-

Note: Please specify different types of input logic level for different types of pins.

5.2.4.2 General Purpose I/O AC Parameters

The I/O AC simulation parameters for GPIO in slow and fast modes are presented in the Table 5-20 & Table 5-21 respectively. Note that the fast or slow I/O behavior is determined by the appropriate control bits in the GPIO drive strength control registers.

Table 5-20 AC Parameters 1.8 V GPIO Mode

IO Typ.	Parameter	Sym.	Test Condition	Min.	Typ.	Max.	Unit
1.8V GPIO	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.40/0.42	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	0.49/0.53	ns
	Output pad transition Times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	0.68/0.75	ns
	Output pad transition times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	1.25/1.41	ns
18OD33V GPIO	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.81/0.63	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	1.10/0.92	ns
	Output pad transition times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	1.57/1.25	ns
	Output pad transition times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	3.16/2.53	ns
1.8V MSDC3	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.40/0.40	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	0.60/0.63	ns
	Output pad transition times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	0.84/0.91	ns
	Output pad transition times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	2.40/2.64	ns

IO Typ.	Parameter	Sym.	Test Condition	Min.	Typ.	Max.	Unit
18OD33V MSDC3	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.59/0.53	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	0.98/0.89	ns
	Output pad transition times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	1.68/1.50	ns
	Output pad transition times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	6.81/6.12	ns
RGMII	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.69/0.62	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	0.91/0.86	ns
	Output pad transition times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	1.31/1.24	ns
	Output pad transition Times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	2.64/2.52	ns
NOR	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.58/0.42	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	0.90/0.68	ns
	Output pad transition times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	1.11/0.84	ns
	Output pad transition times, rise/fall	tr/tf	Low rive, Cload=10pF	-	-	4.40/3.48	ns

Table 5-21 AC Parameters 3.3V GPIO Mode

IO Typ.	Parameter	Sym.	Test Condition	Min.	Typ.	Max.	Unit
18OD33V GPIO	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.91/0.87	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	1.25/0.92	ns
	Output pad transition times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	1.69/1.22	ns
	Output pad transition times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	3.31/2.44	ns
18OD33V MSDC3	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	1.11/0.91	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	1.32/1.26	ns
	Output pad transition times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	1.63/1.35	ns
	Output pad transition times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	6.32/5.46	ns

IO Typ.	Parameter	Sym.	Test Condition	Min.	Typ.	Max.	Unit
RGMII	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.70/0.62	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	0.91/0.81	ns
	Output pad transition times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	1.29/1.19	ns
	Output pad transition times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	2.54/2.38	ns
NOR	Output pad transition times, rise/fall	tr/tf	Max drive, Cload=10pF	-	-	0.55/0.42	ns
	Output pad transition times, rise/fall	tr/tf	High drive, Cload=10pF	-	-	0.84/0.62	ns
	Output Pad Transition Times, rise/fall	tr/tf	Mid drive, Cload=10pF	-	-	1.02/0.77	ns
	Output pad transition times, rise/fall	tr/tf	Low drive, Cload=10pF	-	-	3.88/3.05	ns

5.2.5 Register Definition

For register details, please refer to Chapter 3.1 of “MT2712 IVI Application Processor Registers”.

5.2.6 Multiple Functions

Table 5-22 provides the multiple functions of each GPIO, showing the mapping table of function name, mode number, pull control and other adjustment attributes.

CU: Controllable pull-up

CD: Controllable pull-down

OD: OpenDrain

Adj: Adjustment

5V Tol: 5 V Tolerance

IO.Dir: IO Direction

I: Input

O: Output

IO: Inout

IES: RX input buffer enable

SMT: RX input buffer schmit trigger hysteresis control enable

RDSEL: input level shifter duty adjustment

TDSEL: Output level shifter duty adjustment

PULL: Pullup and pulldown enable

CFG: Configurable Used for Wake up Source

-: Unsupported mode

Table 5-22 GPIO Control

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
EINT0	0	GPIO0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	DSIA_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	DSIC_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_D3	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	PURE_HW_PROTECT	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
EINT1	0	GPIO1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	IR_IN	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	DSIB_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	DSID_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_D4	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
EINT2	0	GPIO2	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	IR_IN	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	LCM_RST1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_D5	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
EINT3	0	GPIO3	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	IR_IN	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	LCM_RST0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_D6	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
PWM0	0	GPIO4	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	PWM0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DISP0_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	DISP1_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_CLK	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
PWM1	0	GPIO5	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	PWM1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DISP1_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
PWM2	3	DISPO_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_VSYNC	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO6	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	PWM2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DISPO_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
PWM3	3	DISP1_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	DISP2_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_HSYNC	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO7	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	PWM3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DISP1_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
PWM4	3	DISPO_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	LCM_RST2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_D0	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO8	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	PWM4	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DISPO_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
PWM5	3	DISP1_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	DSIA_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_D1	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO9	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	PWM5	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DISP1_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
PWM6	3	DISPO_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	DSIB_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_D2	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO10	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	PWM6	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	DISPO_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
PWM7	0	GPIO11	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	PWM7	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	DISP1_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DISPO_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	LCM_RST1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
IDDIG_P0	0	GPIO12	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	IDDIG_A	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DIN_D7	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
DRV_VBUS_P0	0	GPIO13	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	DRV_VBUS_A	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
IDDIG_P1	0	GPIO14	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	IDDIG_B	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
DRV_VBUS_P1	0	GPIO15	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	DRV_VBUS_B	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
DRV_VBUS_P2	0	GPIO16	IO	-	Adj	-	-	CU/CD	CFG	1.8V
	1	DRV_VBUS_C	O	-	Adj	-	-	CU/CD	CFG	1.8V
	2	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	3	-	-	-	-	Adj	-	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
DRV_VBUS_P3	4	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	5	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	6	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	7	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	0	GPIO17	IO	-	Adj	-	-	CU/CD	CFG	1.8V
	1	DRV_VBUS_D	O	-	Adj	-	-	CU/CD	CFG	1.8V
	2	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	3	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
KPROW0	4	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	5	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	6	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	7	-	-	-	Adj	-	-	CU/CD	CFG	1.8V
	0	GPIO18	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KROW0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPCOL0	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO19	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KCOL0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPROW1	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO20	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KROW1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPCOL1	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
KPROW2	0	GPIO22	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KROW2	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	DISP1_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPCOL2	0	GPIO23	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KCOL2	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	DISPO_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
CMMCLK	0	GPIO24	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	CMMCLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
CM2MCLK	0	GPIO25	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	CM2MCLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
PCM_TX	0	GPIO26	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	PCM1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	MRG_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DAI_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	MRG_RX	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	DAI_RX	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	PCM1_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
PCM_CLK	0	GPIO27	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	PCM1_CLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	MRG_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DAI_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
PCM_RX	0	GPIO28	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	PCM1_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	MRG_RX	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DAI_RX	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	MRG_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	DAI_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	PCM1_DB	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
PCM_SYNC	0	GPIO29	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	PCM1_SYNC	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	MRG_SYNC	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DAI_SYNC	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
NCEB0	0	GPIO30	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	NCEB0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
NCEB1	0	GPIO31	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	NCEB1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
NF_DQS	0	GPIO32	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	NF_DQS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
NWEB	0	GPIO33	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	NWEB	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
NREB	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO34	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
NCLE	1	NREB	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO35	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
NALE	1	NCLE	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO36	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
MSDCOE_CLK	1	NALE	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO37	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DAT7	1	MSDC0_CLK	O	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	0	GPIO38	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DAT7	1	MSDC0_DAT7	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_ND7	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DAT6	0	GPIO39	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_DAT6	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_ND6	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DAT5	0	GPIO40	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_DAT5	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_ND5	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DATA4	0	GPIO41	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_DAT4	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_ND4	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DAT3	0	GPIO42	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_DAT3	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_ND3	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DAT2	0	GPIO43	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_DAT2	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_ND2	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DAT1	0	GPIO44	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_DAT2	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_ND1	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DAT0	0	GPIO45	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_DATO	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_NDO	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_CMD	0	GPIO46	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_CMD	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	NAND_NRNB	I	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_DSL	0	GPIO47	IO	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	1	MSDC0_DSL	I	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	No	1.8V
MSDCOE_RSTB	0	GPIO48	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	MSDC0_RSTB	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
MSDC1_CLK	0	GPIO63	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC1_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UDI_TCK	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC1_DAT3	0	GPIO64	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
MSDC1_DAT1	1	MSDC1_DAT3	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UDI_TDI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC1_DAT2	0	GPIO65	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC1_DAT1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UDI_TMS	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC1_PSW	0	GPIO66	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC1_DAT2	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UDI_TDO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC1_DATO	0	GPIO67	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UDI_NTRST	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC1_CMD	0	GPIO68	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC1_DAT0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
MSDC1_INS	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO70	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	--	--	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC2_CLK	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO89	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC2_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC2_DAT3	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO90	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC2_DAT3	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC2_DAT2	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO91	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC2_DAT2	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC2_DAT1	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO92	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC2_DAT1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC2_DATO	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO93	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC2_DATO	1	MSDC2_DATO	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
MSDC2_INS	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO94	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC2_CMD	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO95	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	MSDC2_CMD	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC2_PSW	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO96	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
MSDC3_DAT3	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO49	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	MSDC3_DAT3	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
MSDC3_DAT2	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
MSDC3_DAT1	0	GPIO51	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	MSDC3_DAT1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
MSDC3_DAT0	0	GPIO52	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	MSDC3_DAT0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
MSDC3_CMD	0	GPIO53	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	MSDC3_CMD	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
MSDC3_INS	0	GPIO54	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	MSDC3_INS	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
MSDC3_DSL	0	GPIO55	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	MSDC3_DSL	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
MSDC3_CLK	0	GPIO56	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	MSDC3_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
NOR_CS	0	GPIO57	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	NOR_CS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
NOR_CK	0	GPIO58	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	NOR_CK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
NOR_IO0	0	GPIO59	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	NOR_IO0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
NOR_IO1	0	GPIO60	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	NOR_IO1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
NOR_IO2	0	GPIO61	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	NOR_IO2	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
NOR_IO3	0	GPIO62	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	NOR_IO3	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
GBE_TXD3	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO71	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_TXD2	1	GBE_TXD3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO72	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_TXD1	1	GBE_TXD2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO73	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_TXD0	1	GBE_TXD1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO74	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_TXC	1	GBE_TXD0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO75	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_TXC	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_TXEN	0	GPIO76	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_TXEN	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_TXER	0	GPIO77	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_TXER	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_RXD3	0	GPIO78	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_RXD3	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_RXD2	0	GPIO79	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_RXD2	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_RXD1	0	GPIO80	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_RXD1	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_RXD0	0	GPIO81	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_RXD0	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
GBE_RXDV	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO82	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_RXDV	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_RXER	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO83	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_RXER	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_RXC	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO84	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_RXC	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_MDC	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO85	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_MDC	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_MDIO	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO86	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_MDIO	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
GBE_CBL	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO87	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
GBE_INTR	1	GBE_CBL	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
URXD0	0	GPIO88	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	GBE_INTR	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	GBE CRS	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
URXD1	0	GPIO117	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	URXD0	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	UTXD0	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
URXD2	0	GPIO118	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	URXD1	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	UTXD1	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
URXD3	0	GPIO119	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	URXD2	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	UTXD2	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
URXD4	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO97	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	URXD4	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UTXD4	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	MRG_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_CLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2S_IQ2_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
URXD5	6	I2SO1_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO101	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	URXD5	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UTXD5	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2SO3_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_LRCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
UTXD0	6	I2SO0_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO120	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	UTXD0	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	URXD0	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
UTXD1	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO121	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	UTXD1	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	URXD1	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
UTXD2	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO122	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	UTXD2	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	URXD2	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
UTXD3	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO124	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
UTXD3	1	UTXD3	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	URXD3	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
UTXD4	3	PURE_HW_PROTECT	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO99	IO	Adj	Adj	Adj	Adj	CD/CD	CFG	1.8V/3.3V
	1	UTXD4	O	Adj	Adj	Adj	Adj	CD/CD	CFG	1.8V/3.3V
	2	URXD4	I	Adj	Adj	Adj	Adj	CD/CD	CFG	1.8V/3.3V
UTXD5	3	MRG_SYNC	O	Adj	Adj	Adj	Adj	CD/CD	CFG	1.8V/3.3V
	4	PCM1_SYNC	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2S_IQ0_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2SO1_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO103	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	UTXD5	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	URXD5	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
URTS3	3	I2SO3_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2SO0_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO125	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V
	1	URTS3	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V
	2	UCTS3	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V
URTS4	3	WATCH_DOG	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V
	0	GPIO98	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	URTS4	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UCTS4	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
URTS5	3	MRG_RX	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2S_IQ1_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2SO1_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO102	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	URTS5	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	UCTS5	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
UCTS3	0	GPIO126	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	UCTS3	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	URTS3	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	SRCLKENAO	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
UCTS4	0	GPIO100	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	UCTS4	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	URTS4	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	MRG_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2S_IQ0_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2SO1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
UCTS5	0	GPIO104	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	UCTS5	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	URTS5	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2SO0_D01	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	IR_IN	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2SO0_D00	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2C_SDA0	0	GPIO105	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SDA0	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SDA1	0	GPIO106	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SDA1	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SDA2	0	GPIO107	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SDA2	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SDA3	0	GPIO108	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SDA3	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SDA4	0	GPIO109	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SDA4	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SDA5	0	GPIO110	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SDA5	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SCL0	0	GPIO111	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SCL0	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SCL1	0	GPIO112	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SCL1	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SCL2	0	GPIO113	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SCL2	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
I2C_SCL3	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO114	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SCL4	1	SCL3	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO115	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
I2C_SCL5	1	SCL4	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO116	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
SPI2_CSN	1	SCL5	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO127	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
SPI2_MO	1	SPI_CS_2_	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	SPI_CS_1_	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	0	GPIO128	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
SPI2_MI	0	GPIO129	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SPI_MI_2_	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	SPI_SI_1_	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
SPI2_CK	0	GPIO130	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SPI_CK_2_	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	SPI_CK_1_	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
SPI3_CSN	0	GPIO131	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SPI_CS_3_	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
SPI3_MO	0	GPIO132	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SPI_MO_3_	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
SPI3_MI	0	GPIO133	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SPI_MI_3_	I	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
SPI3_CK	0	GPIO134	IO	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	1	SPI_CK_3_	O	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	2	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	3	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	4	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	5	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	6	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
	7	-	-	Adj	Adj	Adj	Adj	CD/OD	CFG	1.8V/5V Tol
KPROW3	0	GPIO135	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KROW3	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	DSIC_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPROW4	0	GPIO136	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KROW4	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	DSID_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPCOL3	0	GPIO137	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KCOL3	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	DISP2_PWM	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPCOL4	0	GPIO138	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KCOL4	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	LCM_RST2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPCOL5	0	GPIO139	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KCOL5	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DSIA_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	PURE_HW_PROTECT	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPCOL6	0	GPIO140	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
KPROW5	1	KCOL6	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	WATCH_DOG	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	LCM_RST1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
KPROW6	0	GPIO141	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KROW5	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	LCM_RST0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	PURE_HW_PROTECT	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
JTDO_ICE	0	GPIO142	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	KROW6	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	SRCLKENA0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DSIB_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
JTCK_ICE	0	GPIO143	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	JTDO_ICE	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DFD_TDO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
JTDI_ICE	0	GPIO144	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	JTCK_ICE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DFD_TCK	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
JTMS_ICE	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO146	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	JTMS_ICE	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DFD_TMS	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
JTRSTB_ICE	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO147	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	JTRST_B_ICE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DFD_NTRST	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
GPIO148	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO148	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	JTRSTB_CM4	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DFD_NTRST	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
GPIO149	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO149	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	JTCK_CM4	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DFD_TCK	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
GPIO150	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO150	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	JTMS_CM4	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DFD_TMS	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
GPIO151	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO151	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
1	JTDI_CM4	I	Adj	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
GPIO152	3	DFD_TDI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO152	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	1	JTDO_CM4	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	3	DFD_TDO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
SPIO_CSN	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V
	0	GPIO153	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CS_0_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	SRCLKENA0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	UTXD0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2SO0_DO1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPIO_MI	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ2_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO154	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_MI_0_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	SRCLKENA0	O-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	URXD0	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2SO0_D00	O-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPIO_CK	5	I2SO1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_DATA	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ1_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO155	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CK_0_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	SC_APBIAS_OFF	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	UTXD1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2SO0_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPIO_MO	5	I2SO1_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_BCK	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ0_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO156	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CK_0_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	SC_APBIAS_OFF	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	URXD1	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2SO0_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO1_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_LRCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ0_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
SPI5_CSN	0	GPIO157	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CS_5_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	LCM_RST0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	UTXD2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2SO0_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO1_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_MCLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI5_MI	0	GPIO158	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_MI_5_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DSIA_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	URXD2	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI5_MO	0	GPIO159	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_MO_5_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DSIB_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	UTXD3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI5_CK	0	GPIO160	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CK_5_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	LCM_RST1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	URXD3	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI1_CSN	0	GPIO161	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CS_1_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	SPI_CS_4_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2S_IQ2_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO2_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2SO0_DO1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI1_SI	0	GPIO162	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_SI_1_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	SPI_MI_4_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2S_IQ1_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	5	I2SO2_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_DATA	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2SO0_D00	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI1_CK	0	GPIO163	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CK_1_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	SPI_CK_4_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2S_IQ0_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO2_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2SO0_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI1_SO	0	GPIO164	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_SO_1_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	SPI_MO_4_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2S_IQ0_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO2_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_LRCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2SO0_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI4_CSN	0	GPIO165	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CS_4_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	LCM_RST0	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	SPI_CS_1_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	UTXD4	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_MCLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2SO0_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI4_MI	0	GPIO166	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_MI_4_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DSIA_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	SPI_SI_1_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	URXD4	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO1_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI4_MO	0	GPIO167	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_MO_4_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	DSIB_TE	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	SPI_SO_1_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	UTXD5	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO1_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
SPI4_CK	0	GPIO168	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	SPI_CK_4_	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
I2S10_DATA	2	LCM_RST1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	SPI_CK_1_	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	URXD5	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	I2SO1_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S10_LRCK	0	GPIO169	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S10_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S11_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S12_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S10_MCLK	0	GPIO170	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S10_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S11_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S12_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_LRCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM00_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM01_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S10_BCK	0	GPIO171	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S10_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S11_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S12_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_MCLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM00_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM01_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S12_DATA	0	GPIO172	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S10_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S11_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S12_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM00_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM01_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
I2S12_MCLK	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO174	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S12_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S10_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S11_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMIN_MCLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	PCM1_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S12_BCK	7	I2S_IQ2_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO175	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S12_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S10_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S11_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_CLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMIN_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S12_LRCK	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO176	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S12_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S10_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S11_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_SYNC	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMIN_LRCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S11_DATA	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO177	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S11_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S10_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S12_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S11_BCK	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO178	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S11_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S10_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S12_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM00_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM01_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S11_LRCK	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO179	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S11_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S10_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S12_WS	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
I2SI1_MCLK	4	TDMIN_LRCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM00_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM01_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2SO1_DATA0	0	GPIO180	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S11_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S10_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S12_MCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMIN_MCLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM00_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM01_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ2_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2SO1_BCK	0	GPIO181	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S01_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S00_D0O	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S02_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	DAI_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMIN_MCLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ2_SDIA	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2SO1_LRCK	0	GPIO182	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S01_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S00_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S02_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	DAI_SYNC	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMIN_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ2_BCK	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2SO1_MCLK	0	GPIO183	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S01_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S00_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S02_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	DAI_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMIN_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDM00_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ2_WS	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
AUD_EXT_CK2	0	GPIO185	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
AUD_EXT_CK1	1	AUD_EXT_CK2	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	AUD_EXT_CK1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2SO1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2SI2_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	MRG_RX	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	PCM1_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ0_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2SO2_BCK	0	GPIO186	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	AUD_EXT_CK1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	AUD_EXT_CK2	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2SO0_DO1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2SI1_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	MRG_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	PCM1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ0_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2SO2_LRCK	0	GPIO187	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2SO2_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2SO0_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2SO1_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_CLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	MRG_SYNC	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDMO1_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ0_BCK	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2SO2_MCLK	0	GPIO188	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2SO2_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2SO0_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2SO1_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_SYNC	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	MRG_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDMO1_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ0_WS	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2SO2_DATA0	0	GPIO189	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2SO2_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2SO0_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2SO1_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	PCM1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	MRG_RX	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	TDMO1_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ0_SDQA	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
I2S00_DATA1	6	PCM1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ0_SDIA	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S00_MCLK	0	GPIO191	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S00_DO1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S10_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S11_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	I2S12_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	DAI_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2S_IQ0_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ1_SDQB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S00_DATA0	0	GPIO192	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S00_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S01_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S02_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM01_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2S_IQ0_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ1_SDQA	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S00_LRCK	0	GPIO193	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S00_D00	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S01_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S02_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ1_SDIA	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
I2S00_BCK	0	GPIO194	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S00_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S01_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S02_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM01_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ1_WS	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
TDM01_MCLK	0	GPIO195	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	I2S00_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	I2S01_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	I2S02_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM01_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ1_BCK	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
TDMO1_LRCK	3	TDMIN_MCLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2S00_D01	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S_IQ1_SDIB	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO197	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	TDMO1_LRCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	TDMO0_LRCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
TDMO1_BCK	3	TDMIN_LRCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMO0_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMO1_DATA3	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2S03_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	TDMO1_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO198	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	TDMO1_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	TDMO0_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
TDMO1_DATA	3	TDMIN_BCK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMO0_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMO1_DATA2	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2S03_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	TDMO1_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO199	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	TDMO1_DATA	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	TDMO0_DATA	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
TDMO0_MCLK	3	TDMIN_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMO0_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMO1_DATA1	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	I2S03_WS	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO200	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	TDMO0_MCLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	TDMO1_MCLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
TDMO0_LRCK	3	PCM1_DI	I	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDMO0_MCLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDMO1_MCLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	MRG_TX	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2S02_MCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	0	GPIO201	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	TDMO0_LRCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	TDMO1_LRCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
TDM00_BCK	0	GPIO202	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	TDM00_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	TDM01_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	PCM1_CLK	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDM00_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM01_LRCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	MRG_SYNC	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2SO2_BCK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
TDM00_DATA	0	GPIO203	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	TDM00_DATA	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	TDM01_DATA	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	PCM1_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	TDM00_DATA	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	TDM01_DATA	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	MRG_CLK	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	I2SO2_DO	O	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
PERSTB_P0	0	GPIO204	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	PERST_B_P0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
CLKREQN_P0	0	GPIO205	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	CLKREQ_N_P0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
WAKEEN_P0	0	GPIO206	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	WAKE_EN_P0	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
PERSTB_P1	0	GPIO207	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	PERST_B_P1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Pin Name	Mode	Func. Name	IO.Dir	IES	SMT	RDSEL	TDSEL	PULL	Suspend Wake up	Voltage Domain
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
CLKREQN_P1	0	GPIO208	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	CLKREQ_N_P1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
WAKEEN_P1	0	GPIO209	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	1	WAKE_EN_P1	IO	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	2	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	3	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	4	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	5	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	6	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V
	7	-	-	Adj	Adj	Adj	Adj	CU/CD	CFG	1.8V/3.3V

Table 5-23 shows the reset status and driving strength of all IOs.

I: Input

OL: Output low

OH: Output high

OZ: Output high-z

PU: Pull up

PD: Pull down

Table 5-23 MT2712 Pins Reset Status & Driving

Pin Name	Reset	1.8V Mode Output Driving(mA)				3.3V Mode Output Driving(mA)				Connection For Unused Pins	IO Type
SYSTEM											
RESET_	I	-	PU	-	-	-	-	-	-	Floating	IO type 1
GPIO											
EINT0	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)
EINT1	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)
EINT2	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)
EINT3	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)
PWM											
PWM0	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)
PWM1	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)
PWM2	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)
PWM3	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)

Pin Name	Reset			1.8V Mode Output Driving(mA)				3.3V Mode Output Driving(mA)				Connection For Unused Pins	IO Type	
PWM4	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2	
PWM5	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2	
PWM6	I	0	PD	4(Def.)	8	12	16	-	-	-	-	Floating	IO type 2	
PWM7	I	0	PD	4(Def.)	8	12	16	-	-	-	-	Floating	IO type 2	
USB														
IDDIG_P0	I	1	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2	
IDDIG_P1	I	1	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2	
DRV_VBUS_P0	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2	
DRV_VBUS_P1	I	0	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2	
KEYPAD														
KPROW0	OL	1	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPROW1	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPROW2	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPROW3	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPROW4	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPROW5	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPROW6	OH	2	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPCOL0	I	1	PU	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPCOL1	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPCOL2	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPCOL3	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPCOL4	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPCOL5	I	0	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
KPCOL6	OL	2	PD	2(Def.)	4	6	8	-	-	-	-	Floating	IO type 2	
PCM														
PCM_TX	I	0	PD	4(Def.)	8	12	16	-	-	-	-	Floating	IO type 2	
PCM_CLK	I	0	PD	4(Def.)	8	12	16	-	-	-	-	Floating	IO type 2	
PCM_RX	I	0	PD	4(Def.)	8	12	16	-	-	-	-	Floating	IO type 2	
PCM_SYNC	I	0	PD	4(Def.)	8	12	16	-	-	-	-	Floating	IO type 2	
CAM														
CMMCLK	O:CLK	1	PD	4(Def.)	8	12	16	-	-	-	-	Floating	IO type 2	
CM2MCLK	O:CLK	1	PD	4(Def.)	8	12	16	-	-	-	-	Floating	IO type 2	
NAND														
NCEB0	OH	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
NCEB1	OH	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
NF_DQS	OL	1	PD	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
NWEB	OH	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
NREB	OH	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
NCLE	OL	1	PD	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
NALE	OL	1	PD	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDC0														
MSDCOE_CLK	OL	1	PD	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_DAT7	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_DAT6	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_DAT5	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_DATA4	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_DATA3	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_DATA2	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_DATA1	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_DATO	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2
MSDCOE_CMD	I	1	PU	2	4(Def.)	6 8	10 12	14 16	-	-	-	-	Floating	IO type 2

Pin Name	Reset		1.8V Mode Output Driving(mA)						3.3V Mode Output Driving(mA)						Connection For Unused Pins	IO Type	
			2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)			
MSDCOE_DSL	I	1	PD	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDCOE_RSTB	OH	1	PU	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDC1																	
MSDC1_CLK	OL	1	PD	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC1_DAT3	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC1_DAT2	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC1_DAT1	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC1_DATO	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC1_PSW	I	0	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC1_CMD	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC1_INS	I	0	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC2																	
MSDC2_CLK	OL	1	PD	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC2_DAT3	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC2_DAT2	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC2_DAT1	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC2_DATO	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC2_INS	I	0	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC2_CMD	I	1	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC2_PSW	I	0	PU	2	4(Def.)	6	8	10	12	14	16	2	5(Def.)	7	10(Forb.)	Floating	IO type 2
MSDC3																	
MSDC3_DAT3	I	1	PU	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDC3_DAT2	I	1	PU	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDC3_DAT1	I	1	PU	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDC3_DATO	I	1	PU	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDC3_CMD	I	1	PU	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDC3_INS	I	1	PD	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDC3_DSL	I	1	PD	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
MSDC3_CLK	OL	1	PD	2	4(Def.)	6	8	10	12	14	16	-	-	-	-	Floating	IO type 2
NOR																	
NOR_CS	OL	1	PU	2(Def.)	4	6	8	10	12	14	16	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
NOR_CK	OL	1	PD	2(Def.)	4	6	8	10	12	14	16	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
NOR_IO0	I	1	PD	2(Def.)	4	6	8	10	12	14	16	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
NOR_IO1	I	1	PD	2(Def.)	4	6	8	10	12	14	16	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
NOR_IO2	I	1	PU	2(Def.)	4	6	8	10	12	14	16	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
NOR_IO3	I	1	PU	2(Def.)	4	6	8	10	12	14	16	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
ETHERNET																	
GBE_TXD3	OL	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_TXD2	OL	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_TXD1	OL	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_TXD0	OL	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_TXC	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_TXEN	OL	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_TXER	OL	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_RXD3	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_RXD2	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_RXD1	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_RXD0	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_RXDV	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_RXER	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2
GBE_RXC	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	-	-	-	-	Floating	IO type 2

Pin Name	Reset			1.8V Mode Output Driving(mA)			3.3V Mode Output Driving(mA)			Connection For Unused Pins	IO Type			
	OL	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)		
GBE_MDC	OL	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
GBE_MDIO	I	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
GBE_COL	I	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
GBE_INTR	I	I	1	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
UART														
URXD0	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
URXD1	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
URXD2	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
URXD3	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
URXD4	I	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
URXD5	I	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
UTXD0	OZ	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
UTXD1	OZ	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
UTXD2	OZ	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
UTXD3	OZ	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
UTXD4	OH	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
UTXD5	OH	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
URTS3	OL	I	3	-	-	-	-	-	-	-	-	External pullup	IO type 3	
URTS4	OH	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
URTS5	OH	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
UCTS3	OZ	I	3	-	-	-	-	-	-	-	-	External pullup	IO type 3	
UCTS4	I	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
UCTS5	I	I	1	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2C0														
I2C_SDA0	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C_SCL0	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C1														
I2C_SDA1	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C_SCL1	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C2														
I2C_SDA2	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C_SCL2	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C3														
I2C_SDA3	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C_SCL3	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C4														
I2C_SDA4	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C_SCL4	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C5														
I2C_SDA5	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
I2C_SCL5	I	I	1	-	-	-	-	-	-	-	-	External pullup	IO type 3	
SPI0														
SPI0_CS0	I	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
SPI0_MI	I	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
SPI0_CK	I	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
SPI0_MO	I	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
SPI1														
SPI1_CS0	I	I	0	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
SPI1_SI	I	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
SPI1_CK	I	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2

Pin Name	Reset			1.8V Mode Output Driving(mA)			3.3V Mode Output Driving(mA)			Connection For Unused Pins	IO Type
SPI1_SO	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
SPI2											
SPI2_CSN	I	0	PD	-	-	-	-	-	-	-	Floating
SPI2_MO	I	0	PD	-	-	-	-	-	-	-	Floating
SPI2_MI	I	0	PD	-	-	-	-	-	-	-	Floating
SPI2_CK	I	0	PD	-	-	-	-	-	-	-	Floating
SPI3											
SPI3_CSN	I	0	PD	-	-	-	-	-	-	-	Floating
SPI3_MO	I	0	PD	-	-	-	-	-	-	-	Floating
SPI3_MI	I	0	PD	-	-	-	-	-	-	-	Floating
SPI3_CK	I	0	PD	-	-	-	-	-	-	-	Floating
SPI4											
SPI4_CSN	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
SPI4_MI	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
SPI4_MO	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
SPI4_CK	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
SPI5											
SPI5_CSN	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
SPI5_MI	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
SPI5_MO	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
SPI5_CK	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
JTAG_ICE											
JTDO_ICE	OL	1	PU	4(Def.)	8	12	16	-	-	-	Floating
JTCK_ICE	I	1	PU	4(Def.)	8	12	16	-	-	-	Floating
JTDI_ICE	I	1	PU	4(Def.)	8	12	16	-	-	-	Floating
JTMS_ICE	I	1	PU	4(Def.)	8	12	16	-	-	-	Floating
JTRSTB_ICE	I	1	PD	4(Def.)	8	12	16	-	-	-	Floating
GPIO											
GPIO148	I	1	PD	4(Def.)	8	12	16	-	-	-	Floating
GPIO149	I	1	PU	4(Def.)	8	12	16	-	-	-	Floating
GPIO150	I	1	PU	4(Def.)	8	12	16	-	-	-	Floating
GPIO151	I	1	PU	4(Def.)	8	12	16	-	-	-	Floating
GPIO152	OL	1	PU	4(Def.)	8	12	16	-	-	-	Floating
I2S10											
I2S10_DATA	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S10_LRCK	I	0	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S10_MCLK	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S10_BCK	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S11											
I2S11_DATA	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S11_BCK	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S11_LRCK	I	0	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S11_MCLK	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S12											
I2S12_DATA	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S12_MCLK	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S12_BCK	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2S12_LRCK	I	0	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2SO0											
I2SO0_DATA1	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)
I2SO0_DATA0	I	0	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)

Pin Name	Reset			1.8V Mode Output Driving(mA)			3.3V Mode Output Driving(mA)			Connection For Unused Pins	IO Type		
	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)		
I2SO0_MCLK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO0_LRCK	I	O	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO0_BCK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO1													
I2SO1_DATA0	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO1_BCK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO1_LRCK	I	O	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO1_MCLK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO2													
I2SO2_BCK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO2_LRCK	I	O	PU	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO2_MCLK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
I2SO2_DATA0	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
AUDIO													
AUD_EXT_CK2	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
AUD_EXT_CK1	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
TDM00													
TDM00_MCLK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
TDM00_LRCK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
TDM00_BCK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
TDM00_DATA	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
TDM01													
TDM01_MCLK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
TDM01_LRCK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
TDM01_BCK	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
TDM01_DATA	I	O	PD	2(Def.)	4	6	8	4	8(Def.)	12(Forb.)	16(Forb.)	Floating	IO type 2
PCIE													
PERSTB_P0	I	O	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
CLKREQN_P0	I	O	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
WAKEEN_P0	I	O	PU	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
PERSTB_P1	I	O	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
CLKREQN_P1	I	O	PD	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2
WAKEEN_P1	I	O	PU	2(Def.)	4	6	8	4(Def.)	8	12(Forb.)	16(Forb.)	Floating	IO type 2

Table 5-24 shows the corresponding relation between external interrupts and pin names.

Table 5-24 MT2712 Interrupt Source

Name	Eint Num.	Note
EINT0	eint[6]	Need Enable Debounce Feature
EINT1	eint[7]	Need Enable Debounce Feature
EINT2	eint[8]	Need Enable Debounce Feature
EINT3	eint[9]	Need Enable Debounce Feature

Name	Eint Num.	Note
PWM0	eint[10]	Need Enable Debounce Feature
PWM1	eint[11]	Need Enable Debounce Feature
PWM2	eint[12]	Need Enable Debounce Feature
PWM3	eint[13]	Need Enable Debounce Feature
PWM4	eint[14]	Need Enable Debounce Feature
PWM5	eint[15]	Need Enable Debounce Feature
PWM6	eint[16]	Need Enable Debounce Feature
PWM7	eint[17]	Need Enable Debounce Feature
IDDIG_P0	eint[42]	-
DRV_VBUS_P0	eint[43]	-
IDDIG_P1	eint[44]	-
DRV_VBUS_P1	eint[45]	-
DRV_VBUS_P2	eint[46]	-
DRV_VBUS_P3	eint[47]	-
KPROW0	eint[18]	Need Enable Debounce Feature
KPCOLO	eint[19]	Need Enable Debounce Feature
KPROW1	eint[48]	-
KPCOL1	eint[49]	-
KPROW2	eint[50]	-
KPCOL2	eint[51]	-
CMMCLK	eint[52]	-
CM2MCLK	eint[53]	-
PCM_TX	eint[54]	-
PCM_CLK	eint[55]	-
PCM_RX	eint[56]	-
PCM_SYNC	eint[57]	-
NCEB0	eint[58]	-
NCEB1	eint[59]	-
NF_DQS	eint[60]	-
NWEB	eint[61]	-
NREB	eint[62]	-
NCLE	eint[63]	-
NALE	eint[64]	-
MSDCOE_RSTB	eint[142]	-
MSDC3_DAT3	eint[65]	-
MSDC3_DAT2	eint[66]	-
MSDC3_DAT1	eint[67]	-
MSDC3_DAT0	eint[68]	-
MSDC3_CMD	eint[69]	-
MSDC3_INS	eint[20]	Need Enable Debounce Feature
MSDC3_DSL	eint[70]	-
MSDC3_CLK	eint[71]	-
NOR_CS	eint[72]	-
NOR_CK	eint[73]	-
NOR_IO0	eint[74]	-
NOR_IO1	eint[75]	-
NOR_IO2	eint[76]	-
NOR_IO3	eint[77]	-
MSDC1_CLK	eint[78]	-
MSDC1_DAT3	eint[79]	-
MSDC1_DAT1	eint[80]	-
MSDC1_DAT2	eint[81]	-

Name	Eint Num.	Note
MSDC1_PSW	eint[82]	-
MSDC1_DAT0	eint[83]	-
MSDC1_CMD	eint[84]	-
MSDC1_INS	eint[85]	-
GBE_TXD3	eint[86]	-
GBE_TXD2	eint[87]	-
GBE_TXD1	eint[88]	-
GBE_RXD0	eint[89]	-
GBE_TXC	eint[90]	-
GBE_TXEN	eint[91]	-
GBE_TXER	eint[92]	-
GBE_RXD3	eint[93]	-
GBE_RXD2	eint[94]	-
GBE_RXD1	eint[95]	-
GBE_RXD0	eint[96]	-
GBE_RXDV	eint[97]	-
GBE_RXER	eint[98]	-
GBE_RXC	eint[99]	-
GBE_MDC	eint[100]	-
GBE_MDIO	eint[101]	-
GBE_COL	eint[102]	-
GBE_INTR	eint[21]	Need Enable Debounce Feature
MSDC2_CLK	eint[103]	-
MSDC2_DAT3	eint[104]	-
MSDC2_DAT2	eint[105]	-
MSDC2_DAT1	eint[106]	-
MSDC2_DAT0	eint[107]	-
MSDC2_INS	eint[108]	-
MSDC2_CMD	eint[109]	-
MSDC2_PSW	eint[110]	-
URXD4	eint[111]	-
URTS4	eint[112]	-
UTXD4	eint[113]	-
UCTS4	eint[114]	-
URXD5	eint[30]	Need Enable Debounce Feature
URTS5	eint[31]	Need Enable Debounce Feature
UTXD5	eint[32]	Need Enable Debounce Feature
UCTS5	eint[33]	Need Enable Debounce Feature
I2C_SDA0	eint[115]	-
I2C_SDA1	eint[116]	-
I2C_SDA2	eint[117]	-
I2C_SDA3	eint[118]	-
I2C_SDA4	eint[119]	-
I2C_SDA5	eint[34]	Need Enable Debounce Feature
I2C_SCL0	eint[120]	-
I2C_SCL1	eint[121]	-
I2C_SCL2	eint[122]	-
I2C_SCL3	eint[123]	-
I2C_SCL4	eint[124]	-
I2C_SCL5	eint[35]	Need Enable Debounce Feature
URXDO	eint[125]	-
URXD1	eint[126]	-

Name	Eint Num.	Note
URXD2	eint[127]	-
UTXDO	eint[128]	-
UTXD1	eint[129]	-
UTXD2	eint[130]	-
URXD3	eint[131]	-
UTXD3	eint[132]	-
URTS3	eint[133]	-
UCTS3	eint[134]	-
SPI2_CSN	eint[135]	-
SPI2_MO	eint[136]	-
SPI2_MI	eint[137]	-
SPI2_CK	eint[138]	-
SPI3_CSN	eint[139]	-
SPI3_MO	eint[143]	-
SPI3_MI	eint[144]	-
SPI3_CK	eint[145]	-
KPROW3	eint[146]	-
KPROW4	eint[36]	Need Enable Debounce Feature
KPCOL3	eint[147]	-
KPCOL4	eint[37]	Need Enable Debounce Feature
KPCOL5	eint[38]	Need Enable Debounce Feature
KPCOL6	eint[39]	Need Enable Debounce Feature
KPROW5	eint[40]	-
KPROW6	eint[41]	-
JTDO_ICE	eint[148]	-
JTCK_ICE	eint[149]	-
JTDI_ICE	eint[150]	-
JTMS_ICE	eint[151]	-
JTRSTB_ICE	eint[152]	-
GPIO148	eint[153]	-
GPIO149	eint[154]	-
GPIO150	eint[155]	-
GPIO151	eint[156]	-
GPIO152	eint[157]	-
SPI0_CSN	eint[158]	-
SPI0_MI	eint[159]	-
SPI0_CK	eint[160]	-
SPI0_MO	eint[161]	-
SPI5_CSN	eint[162]	-
SPI5_MI	eint[163]	-
SPI5_MO	eint[164]	-
SPI5_CK	eint[165]	-
SPI1_CSN	eint[166]	-
SPI1_SI	eint[167]	-
SPI1_CK	eint[168]	-
SPI1_SO	eint[169]	-
SPI4_CSN	eint[170]	-
SPI4_MI	eint[171]	-
SPI4_MO	eint[172]	-
SPI4_CK	eint[173]	-
I2SIO_DATA	eint[174]	-
I2SIO_LRCK	eint[175]	-

Name	Eint Num.	Note
I2S10_MCLK	eint[176]	-
I2S10_BCK	eint[177]	-
I2S12_DATA	eint[178]	-
I2S12_MCLK	eint[179]	-
I2S12_BCK	eint[180]	-
I2S12_LRCK	eint[181]	-
I2S11_DATA	eint[182]	-
I2S11_BCK	eint[183]	-
I2S11_LRCK	eint[184]	-
I2S11_MCLK	eint[185]	-
I2S01_DATA0	eint[186]	-
I2S01_BCK	eint[187]	-
I2S01_LRCK	eint[188]	-
I2S01_MCLK	eint[189]	-
AUD_EXT_CK2	eint[190]	-
AUD_EXT_CK1	eint[191]	-
I2S02_BCK	eint[192]	-
I2S02_LRCK	eint[193]	-
I2S02_MCLK	eint[194]	-
I2S02_DATA0	eint[195]	-
I2S00_DATA1	eint[196]	-
I2S00_MCLK	eint[197]	-
I2S00_DATA0	eint[198]	-
I2S00_LRCK	eint[199]	-
I2S00_BCK	eint[200]	-
TDM01_MCLK	eint[201]	-
TDM01_LRCK	eint[202]	-
TDM01_BCK	eint[203]	-
TDM01_DATA	eint[204]	-
TDM00_MCLK	eint[205]	-
TDM00_LRCK	eint[206]	-
TDM00_BCK	eint[207]	-
TDM00_DATA	eint[208]	-
PERSTB_P0	eint[209]	-
CLKREQN_P0	eint[210]	-
WAKEEN_P0	eint[211]	-
PERSTB_P1	eint[212]	-
CLKREQN_P1	eint[213]	-
WAKEEN_P1	eint[214]	-

5.2.7 Programming Guide

5.2.7.1 GPIO Mode Selection

- Set the mode register to select the mode (0-7)
- Mode1-mode7 are multi-function and decided by design
- Mode0 is GPIO function

5.2.7.2 GPIO Direction Selection

- Set the direction register to select the direction (0: input or 1: output)

5.2.7.3 Pull Selection

- General IO

Set pullen register to enable GPIO pull selection; then set pullsel register to select pull high or pull low

- Special IO

Set R1/R0 register to enable GPIO pull selection; then set pupd register to select pull high or pull low

5.2.7.4 Driving Strength Selection

- Set the driving register to select different driving strength

By setting the IO driving register, VCC33 IOs will have different driving strength in TX mode. As the default driving setting is used in Azalea High Temperature Operating Life (HTOL) qualification, the driving strength smaller than or same as the default driving strength is highly recommended. If there is any change to the default driving setting, please follow the guideline shown in Table 5-23.

5.3 Peripheral Configuration Controller

5.3.1 Introduction

The Peripheral Configuration (Pericfg) Controller is used to control the reset, clock and bus setting of peripheral subsystem. Each module inside the peripheral subsystem has its own software reset and clock gated control (power-down control). The hardware Dynamic Clock Management (DCM) of the peripheral subsystem is also controlled in the Pericfg Controller. Besides AP MCU, the modem MCU can also use this Pericfg Controller to control specific modules with clock gated control (power-down control).

5.3.2 Features

- Support software reset control of each module inside peripheral subsystem
- Support clock gated control of the modules inside peripheral subsystem by AP MCU
- Support DCM control of peripheral subsystem
- Support bus setting (bandwidth limit/way enable/...) of peripheral subsystem

5.3.3 Block Diagram

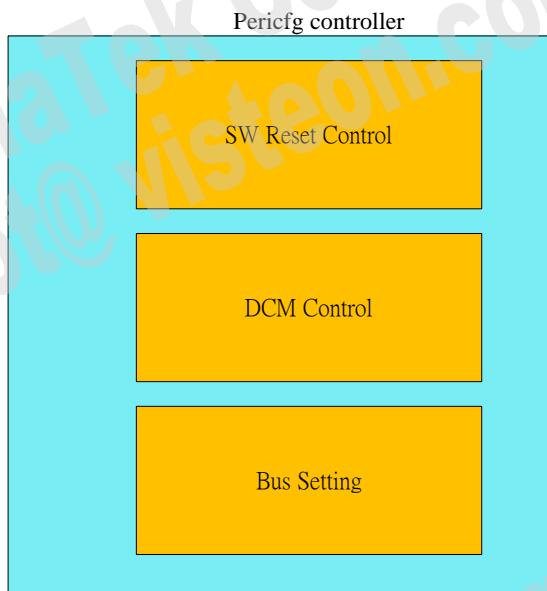


Figure 5-5 Pericfg Controller Block Diagram

5.3.4 Register Definition

For register details, please refer to Chapter 3.2 of “MT2712 IVI Application Processor Registers”.

5.4 Keypad Scanner

5.4.1 Introduction

The keypad supports: 7*7 single keys

5.4.2 Features

The 7*7 keypad can be divided into two parts:

1. The keypad interface that includes 7 columns and 7 rows.
2. The key detection block that provides key press, key release and de-bounce mechanisms.

5.4.3 Block Diagram

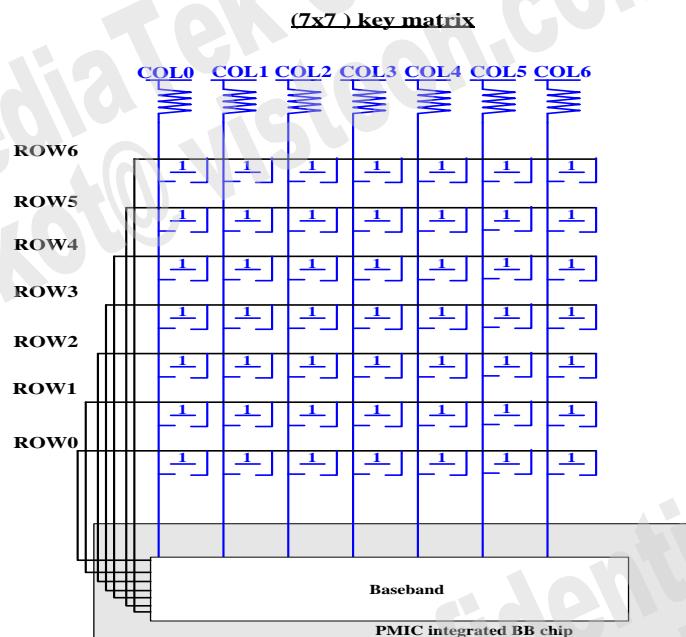


Figure 5-6 Keypad Block Diagram

Each time when the key is pressed or released, i.e. something different happens in the 7x7 matrix. The key detection block senses the change and recognizes if a key has been pressed or released. Whenever the key status changes and then becomes stable, a KEYPAD IRQ will be issued. The MCU can then read the pressed key(s) directly in the KP_MEM1, KP_MEM2, KP_MEM3, KP_MEM4 and KP_MEM5 registers. To ensure the key press information is not missed, the status register in keypad will not be read-cleared by the APB read command. The status register can only be changed by the key press detection FSM.

This keypad detects one or two keys pressed simultaneously with any combination. Figure 5-7 shows the one key pressed condition. Since the key pressed detection depends on the HIGH or LOW level of the external keypad interface, if the keys are pressed at the same time and there exists a key that is in the same column or the same row with other keys, the pressed key cannot be correctly decoded. For example, if there are three keys being pressed: key1 = (x1, y1), key2 = (x2, y2), and key3 = (x1, y2), both key3 and key4 = (x2, y1) will be detected, and therefore they cannot be distinguished correctly. Hence, the keypad detects only one or two keys pressed simultaneously in any combination. More than two keys pressed simultaneously in a specific pattern will retrieve the wrong information.

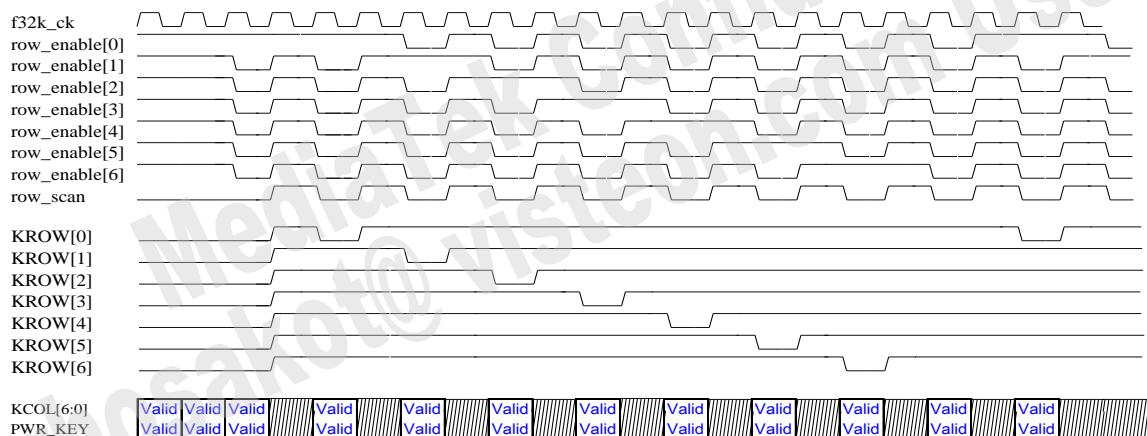


Figure 5-7 7x7 Keypad Scan Waveform

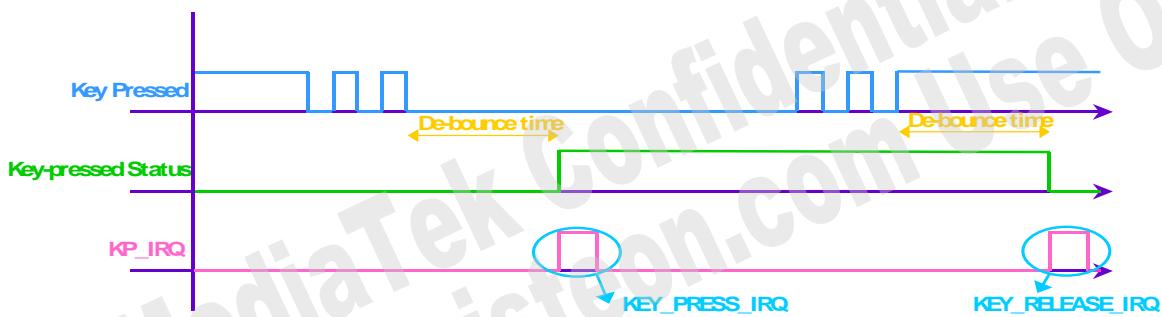


Figure 5-8 One Key Pressed with De-bounce Mechanism Denoted

5.4.4 Keypad AC Timing

Table 5-25 Keypad IO AC Timing

Parameter	Min.	Typ.	Max.	Unit	Note
De-bounce time	0.032	-	512	ms	Depend on de-bounce setting
VCC18IO	1.7	1.8	1.9	V	Supply voltage of IO power
V _{IH}	0.65*VCC18IO	-	VCC18IO+0.3	V	Input logic high voltage
V _{IL}	-0.3	-	0.35*VCC18IO	V	Input logic low voltage
V _{OH} (DC)	0.75*VCC18IO	-	-	V	DC output logic high voltage
V _{OL} (DC)	-	-	0.25*VCC18IO	V	DC output logic low voltage

5.4.5 Register Definition

For register details, please refer to Chapter 3.3 of "MT2712 IVI Application Processor Registers".

5.4.6 Programming Guide

This section describes the operating sequence, as shown in Table 5-26.

Table 5-26 Keypad Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set up keypad						
1	keypad base address + 0x018	KP_DEBOUNCE	DEBOUNCE	RW	14'h400	Set de-bounce time, De-bounce time = KP_DEBOUNCE/32 ms.
2	keypad base address + 0x024	KP_EN	KP_EN	RW	0'h1	Enable keypad
Wait for keypad to issue interrupt; and read keypad MEM register.						

5.5 UART

5.5.1 Introduction

The UARTs provide full duplex serial communication channels between MT2712 chip and external devices. The UART has M16C450 and M16550A operation modes, which are compatible with a range of standard software drivers. The extensions are designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.

In common with the M16550A, the UART supports word lengths from 5 to 8 bits, an optional parity bit and one or two stop bits, and this word length is fully programmable by a CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, along with separate transmission and received FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided. The UART also includes two DMA handshake lines which are used to indicate when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the several sources.

After hardware reset, the UART is in M16C450 mode. Its FIFOs can be enabled and the UART can enter M16550A mode. The UART adds further functionality beyond M16550A mode. Each of the extended functions can be selected individually under software control.

5.5.2 Features

- Provide 6 channels of UARTs
- UART0~UART2 are 2-pin (TX, RX) UART channel
- UART3~UART5 are 4-pin (TX, RX, CTS, RTS) UART channel
- Support both M16C450 and M16550A operation modes
- Compatible with standard software drivers
- Transfer system: Asynchronous

- Data length: 5 to 8 bits
- Hardware flow control: CTS/RTS-based automatic transmission and reception of control
- Software flow control: Use special character Xon/Xoff to do software flow control
- Baud rate is programmable from 300 bps to 3 Mbps
- Baud rate error: Less than 0.25 %
- Interrupt request: Receive interrupts/transmit interrupts
- Data transfer: DMA (Transmit/Receive) transfer is supported

5.5.3 Block Diagram

Each UART has a 32-byte transmit FIFO and a 32-byte receive FIFO.

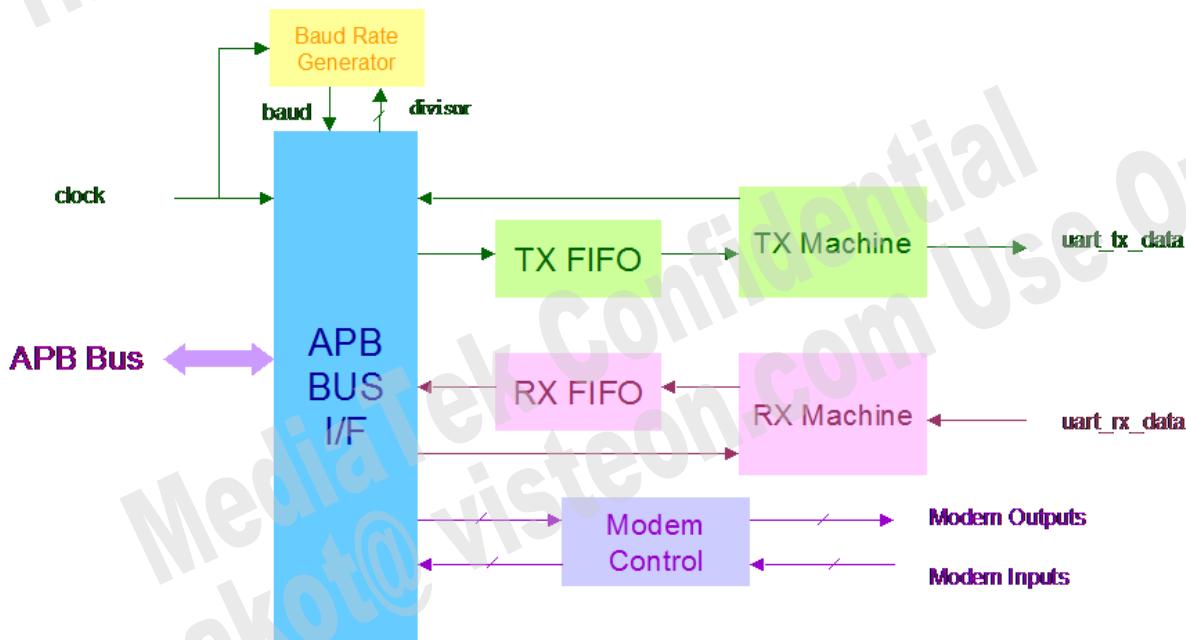


Figure 5-9 UART Block Diagram

5.5.4 UART AC Timing

5.5.4.1 UART (4-pin UART & 2-pin UART) AC Timing

This section describes the AC timing of UART module, which is applicable for both 4-pin UART and 2-pin UART.

5.5.4.2 UART Transmit Timing

Figure 5-10 shows the UART transmit timing, showing only one start bit, eight data bits and one stop bit. Table 5-27 describes the timing parameter (UA1) as shown in the figure.

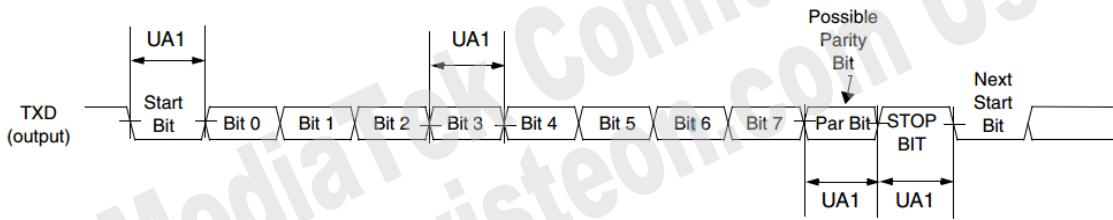


Figure 5-10 UART Transmit Timing Diagram

Table 5-27 UART Transmit Timing Parameters

ID	Parameter	Symbol	Min.	Max.	Units
UA1	Transmit Bit Time	t_{bit}	$1/F_{baud_rate}$	$1/F_{baud_rate}$	Second

Fbaud rate: Baud rate frequency.

The maximum baud rate of the 4-pin UART can support up to 4 Mbps. The maximum baud rate of 2-pin UART is 921.6 Kbps.

5.5.4.3 UART Receive Timing

Figure 5-11 presents the UART receive timing, showing only one start bit, eight data bits and one stop bit.

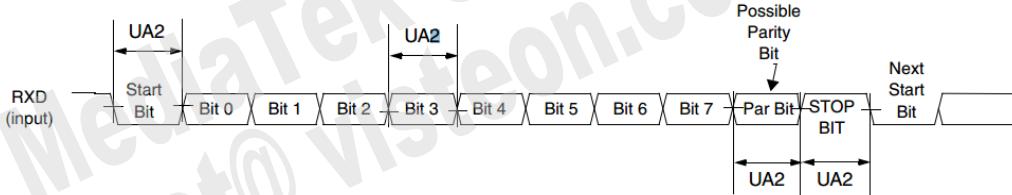


Figure 5-11 UART Receive Timing Diagram

Table 5-28 UART Receive Timing Parameters

ID	Parameter	Symbol	Min.	Max.	Units
UA2	Receive Bit Time	t_{bit}	$1/F_{baud_rate} - R_{ref_clk}$	$1/F_{baud_rate}$	Second

Fbaud rate: Baud rate frequency.

The maximum baud rate of the 4-pin UART can support up to 4 Mbps. The maximum baud rate of 2-pin UART is 921.6 Kbps.

Ref_clk: The period of UART reference clock is represented by ref clk (which is UART module clock bclk= 26 MHz).

The UART receiver can accumulate tolerance in one frame, as it must not exceed the criteria $(bclk/2F_{baud_rate}) - 2) / (10bclk/F_{baud_rate})$.

5.5.5 Communication Protocol

UART communication protocol is as follows:

- 1-bit start bit must be low.
- The data bit length is 5~8 bits.
- The parity check can be odd parity or even parity.
- The stop bit length is 1~2 bit(s). It must be high.

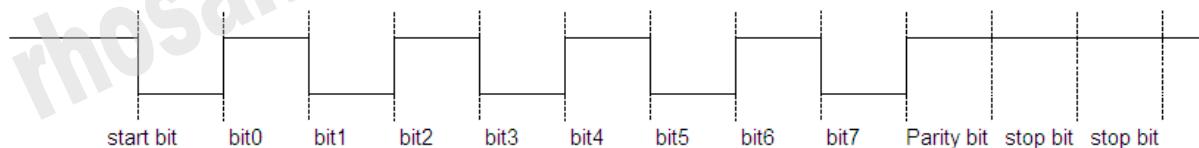


Figure 5-12 UART Communication Protocol

5.5.6 Theory of Operations

UART Enhancement Features

The UART provides more powerful enhancements than the industry-standard 16550:

Hardware flow control: Use two dedicated signals, clear to send (CTS) and request to send (RTS) signals, to indicate UART is ready to get data or send data. When CTS is low, UART can start to transmit data. As long as CTS is active, UART is not allowed to send data. RTS goes low means UART FIFO in received circuit is sufficient to receive data. UART is not allowed to receive data when RTS is high. This feature is very useful when the ISR latency is hard to predict and control in the embedded applications. The MCU is relieved of having to fetch the received data within a fixed amount of time.

Software flow control: Use special character Xon/Xoff to do software flow control. Special character Xon/Xoff is software programmable. When Xoff is received, UART transmission is halted. The transmission will not be resumed until Xon is received.

Note 1: In order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:5], FCR[5:4], ISR[5:4] and MCR[7:6] cannot be written. The Enhanced Mode bit ensures that the UART is backward compatible with software that has been written for 16C450 and 16550A devices.

Note 2: When the oversampling ratio between UART clock and baud rate is less than 8, it is necessary to enable guard time function in customer's UART TX device to make our UART RX work properly. Otherwise, frame error could happen and the received data could get corrupted.

UART Interrupt

UART generates several interrupts. The interrupt types are shown in Table 5-30.

Table 5-29 UART Interrupt Control Bits and Interrupt Factors

UARTn+0004h Interrupt Enable Register																UARTn_IER																	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0 <th>Bit</th> <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>	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CTSI	RTSI	XOFFI	X	EDSSI	ELSI	ETBEI	ERBFI <th>Type</th> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>R/W</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	Type									R/W							
Reset																	0																
IER[3:0] are modified when LCR[7] = 0.																																	
IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.																																	

IER[3:0] are modified when LCR[7] = 0.

IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.

UARTn+0008h Interrupt Identification Register																UARTn_IIR																	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0 <th>Bit</th> <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>	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE		ID4	ID3	ID2	ID1	ID0	NINT	Type									RO							
Reset											0	0	0	0	0	0	1																
ID4 and ID3 are presented only when EFR[4] = 1.																																	

Table 5-30 UART Interrupt Types

Interrupt Type	Interrupt Request Bit (UARTn_IER)	Interrupt Identification (UARTn_IIR)	Interrupt Factor	Notes
Received	ERBFI	IIR[5:0] = 000100b	RX Buffer contains data	-
	-	IIR[5:0] = 001100b	Timeout on character in RX FIFO	-
Transmit	ETBEI	IIR[5:0] = 000010b	TX Holding Register is empty or the contents of the TX FIFO have been reduced to its trigger level.	-
Communication Error	ELSI	IIR[5:0] = 000110b	When frame error, parity error, break interrupt or FIFO overrun happened, the interrupt will be generated.	For the detailed interrupt status, please refer to LSR[4:1]
Modem	EDSSI	IIR[5:0] = 000000b	Modem Status Change	For the detailed interrupt status, please refer to MSR[4:1]
Enhancement Feature	CTSI	IIR[5:0] = 100000b	Hardware Flow Control, When a rising edge is detected on the CTS modem control line	Available when Enhanced Feature is enabled (EFR[4] = 1)
	RTSI	IIR[5:0] = 100000b	Hardware Flow Control, When a rising edge is detected on the RTS modem control line	

Interrupt Type	Interrupt Request Bit (UARTn_IER)	Interrupt Identification (UARTn_IIR)	Interrupt Factor	Notes
	XOFF1	IIR[5:0] = 010000b	Software Flow Control, When an XOFF character is received	

Data Transmission

If the TX Holding Register (THR) is empty (FIFOs are disabled) or TX FIFO is reduced to its trigger level (FIFOs are enabled), then THRE bit of the LSR is “1” and the transmitted data can be written in the THR.

When THR is not empty (FIFOs are disabled) or TX FIFO is increased to its trigger level (FIFOs are enabled), then TX starts transmission automatically.

SW can write transmitted data into THR by asserting transmit interrupt (IIR[5:0] = 000010b) or polling the THRE bit status as “1” directly.

If FIFOs are enabled, the transmitted data can be written into the THR. The data will be transferred to TX FIFO directly.

Data Reception

If the RX Buffer is becoming full or a byte is being transferred into RX FIFO, the DR bit of the LSR is “1” and the received data can be read by RX Buffer Register (RBR).

SW can read the received data when receive interrupt (IIR[5:0] = 000100b) is asserted or UART is polling the DR bit status directly.

If FIFOs are enabled, the received data in the RX FIFO can be read by reading RBR.

Register Description

UART_BASE:

UART0 register base address is 0x11002000.

UART1 register base address is 0x11003000.

UART2 register base address is 0x11004000.

UART3 register base address is 0x11005000.

UART4 register base address is 0x11019000.

UART5 register base address is 0x1000f000.

Table 5-31 UART Register Map

Address	Name	Description
UART_BASE+0x0C	LCR	Line Control Register (LCR)
UART_BASE+0x24	HIGHSPEED	HIGH SPEED UART
UART_BASE+0x28	SAMPLE_COUNT	SAMPLE_COUNT
UART_BASE+0x2C	SAMPLE_POINT	SAMPLE_POINT

Address	Name	Description
UART_BASE+0x34	RATEFIX_AD	Rate Fix Address
UART_BASE+0x3C	GUARD	Guard Time Added Register
UART_BASE+0x40	ESCAPE_DAT	Escape Character register
UART_BASE+0x44	ESCAPE_EN	Escape Enable Register
UART_BASE+0x48	SLEEP_EN	Sleep Enable Register
UART_BASE+0x4C	VFIFO_EN	Virtual FIFO Enable Register
UART_BASE+0x50	RXTRI_AD	RX Trigger Address
UART_BASE+0x54	FRACTDIV_L	Fractional Divider LSB Address
UART_BASE+0x58	FRACTDIV_M	Fractional Divider MSB Address
UART_BASE+0x5C	FCR_RD	FIFO Control Register
UART_BASE+0x60	TX_ACTIVE_EN	TX Active Enable Address

		Condition: LCR[7] == 0		Condition: LCR[7] == 1	
Address	Name	Description		Name	Description
UART_BASE+0x00	THR RBR	TX Holding Register/ RX Buffer Register		DLL	Divisor Latch (LS)
UART_BASE+0x04	IER	Interrupt Enable Register		DLM	Divisor Latch (MS)

		Condition: LCR != 0xBF		Condition: LCR == 0xBF	
Address	Name	Description		Name	Description
UART_BASE+0x08	FCR IIR	FIFO Control Register/ Interrupt Identification Register		EFR	Enhanced Feature Register
UART_BASE+0x10	MCR	Modem Control Register		XON1	XON1
UART_BASE+0x14	LSR	Line Status Register		XON2	XON2
UART_BASE+0x18	MSR	Modem Status Register		XOFF1	XOFF1
UART_BASE+0x1C	SCR	Scratch Register		XOFF2	XOFF2

UARTn+0000h Divisor Latch (LS)																UARTn_DLL																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name									DLL[7:0]																							
Type									R/W																							
Reset									1																							

UARTn+0004h Divisor Latch (MS)																UARTn_DLM																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name									DLM[7:0]																							
Type									R/W																							
Reset									0																							

Note:

DLL & DLM can only be updated if LCR[7] is set ("1"). Note that division by 1 generates a BAUD signal that is constantly high. The table below shows the divisor that needs to generate a given baud rate from CLK inputs of 26 MHz and 30 MHz or 40 MHz.

e.g. clock source: 26 MHz, baud rate: 4800 bps

HIGHSPEED (0x24) = 0 case: $26 \text{ MHz} / 4800 / 16 \approx 339 = 0x153 \rightarrow \text{DLL: } 0x01, \text{ DLM: } 0x53$

HIGHSPEED (0x24) = 1 case: $26 \text{ MHz} / 4800 / 8 \approx 677 = 0x2A3 \rightarrow \text{DLL: } 0x02, \text{ DLM: } 0xA3$

HIGHSPEED (0x24) = 2 case: $26 \text{ MHz} / 4800 / 4 \approx 1354 = 0x54A \rightarrow \text{DLL: } 0x05, \text{ DLM: } 0x4A$

HIGHSPEED (0x24) = 3 case: $26 \text{ MHz} / 4800 / 256 + 1 \approx 22 = 0x16 \rightarrow \text{DLL: } 0x00, \text{ DLM: } 0x16$

UARTn+0024h HIGH SPEED UART																UARTn_HIGHSPEED	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SPEED [1:0]
Name																R/W	
Reset																0	

SPEED UART sample counter base

based on $16 * \text{baud_pulse}$, $\text{baud_rate} = \text{system clock frequency} / 16 / \{\text{DLM, DLL}\}$

based on $8 * \text{baud_pulse}$, $\text{baud_rate} = \text{system clock frequency} / 8 / \{\text{DLM, DLL}\}$

based on $4 * \text{baud_pulse}$, $\text{baud_rate} = \text{system clock frequency} / 4 / \{\text{DLM, DLL}\}$

based on sampe_count * baud_pulse, $\text{baud_rate} = \text{system clock frequency} / \text{sampe_count}$

UARTn+0028h SAMPLE_COUNT																UARTn_SAMPLE_COUNT	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SAMPLECOUNT [7:0]
Name																R/W	
Reset																0	

When HIGHSPEED = 3, the sample_count is the threshold value for UART sample counter (sample_num).

Sample Count = clock source/baud rate/{DLL, DLM} - 1

e.g. clock source: 26 MHz, baud rate: 4800 bps & DLL: 0x00, DLM: 0x16

High Speed (0x24) = 0 & 1 & 2 case: no need to set SAMPLE_COUNT

High Speed (0x24) = 3 case: $26 \text{ MHz} / 4800 / 0x16 - 1 \approx 245 \rightarrow \text{SAMPLE_COUNT} = 245$

UARTn+002Ch SAMPLE_POINT																UARTn_SAMPLE_POINT	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SAMPLEPOINT [7:0]
Name																R/W	
Reset																Ffh	

When HIGHSPEED = 3, UART will get the input data when sample_count = sample_num.

The SAMPLE_POINT is usually (SAMPLE_COUNT/2).

e.g. clock source: 26 MHz, baud rate: 4800 bps & DLL: 0x00, DLM: 0x16 & SAMPLE_COUNT = 245

sample point = $245 / 2 = 123$ (sample the central point to decrease the inaccuracy)

UARTn+0054h Fractional Divider LSB Address															UARTn_Fracdiv_L								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Name																FRACDIV_L							
Type																R/W							
Reset																0	0	0	0	0	0	0	0

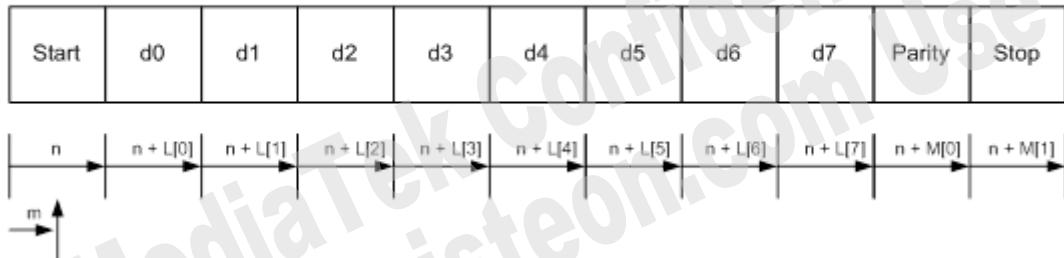
FRACDIV_L: Add sampling count (+1) from state data7 to state data0 in order to improve fractional divisor.

UARTn+0058h Fractional Divider MSB Address															UARTn_FRACDIV_M								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Name																Fracdiv_m							
Type																R/W							
Reset																0	0						

FRACDIV_M: Add sampling count in state stop and state parity in order to improve fractional divisor.

FRACDIV_L/FRACDIV_M: Add one sampling period to each symbol in order to increase the baud rate accuracy.

bit_extend register = FRACDIV_L[7:0]
FRACDIV_M[1:0]



Bit extend number = ROUND ((clock source/baud rate/{DLL, DLM} – (SAMPLE_COUNT + 1))*10)

e.g. clock source: 26 MHz, baud rate: 4800 bps & DLL: 0x00, DLM: 0x16 & SAMPLE_COUNT = 245

Bit extend number = ROUND ((26 MHz/4800/0x16 – (245+1))*10) = 2 → Need to compensate 2 bit (e.g.

FRACDIV_L = 0x44, FRACDIV_M = 0x00)

Refer to FRACDIV_L/FRACDIV_M table.

Table 5-32 Bit Extend Number Reference

Bit Extend Number	FRACDIV_L	FRACDIV_M
0	0x00	0x00
1	0x00	0x10
2	0x00	0x44
3	0x00	0x92
4	0x01	0x29
5	0x01	0xaa
6	0x01	0xb6
7	0x01	0xdb
8	0x01	0xef
9	0x01	0xff

Bit Extend Number	FRACTDIV_L	FRACTDIV_M
10	0x03	0xff

5.5.7 Register Definition

For register details, please refer to Chapter 3.4 of “MT2712 IVI Application Processor Registers”.

5.5.8 Programming Guide

1. Baud rate setting and UART initialization
2. Data transmission
3. Data reception
4. Baud rate setting and UART initialization

Table 5-33 shows the suggestion for UART baud rate setting from CLK inputs of 30 MHz.

Table 5-33 Suggestion for UART Baud Rate Setting

Baud Rate	HIGHSPEED	{DLM, DLL}	SAMPLE_COUNT	SAMPLE_POINT	FRACTDIV_M	FRACTDIV_L
3,000,000	3	0x00, 0x01	0x09	0x03	0x0	0x0
2,000,000	3	0x00, 0x01	0x0E	0x06	0x0	0x0
1,000,000	3	0x00, 0x01	0x1D	0x0D	0x0	0x0
500,000	3	0x00, 0x01	0x3B	0x1C	0x0	0x0
250,000	3	0x00, 0x01	0x77	0x3A	0x0	0x0
153,600	3	0x00, 0x01	0xC2	0x60	0x0	0x92
115,200	3	0x00, 0x02	0x81	0x3F	0x0	0x44
76,800	3	0x00, 0x02	0xC2	0x60	0x0	0x92
57,600	3	0x00, 0x03	0xAC	0x55	0x1	0xB6
38,400	3	0x00, 0x04	0xC2	0x60	0x0	0x92
28,800	3	0x00, 0x05	0xCF	0x66	0x0	0x92
19,200	3	0x00, 0x07	0xDE	0x6E	0x0	0x44
9,600	3	0x00, 0x0D	0xEF	0x76	0x1	0x92
7,200	3	0x00, 0x11	0xF4	0x79	0x1	0x01
4,800	3	0x00, 0x19	0xF9	0x7B	0x0	0x0

Use Register DLL, DLM, HIGHSPEED, SAMPLE_COUNT, SAMPLE_POINT, FRACTDIV_M and FRACTDIV_L to setup baud rate. After setting up the baud rate, UART can start transmission by filling the TX FIFO and receiving data from RX FIFO.

Here is an example for generating baud rate 115200 bps by CLK inputs of 30 MHz:

Table 5-34 UART Baud Rate Setting Example

Step	Description	Related Register Setting
1	Select UART sample counter base to SPEED 3.	HIGHSPEED = 0x3

Step	Description	Related Register Setting
2	Set sample counter	SAMPLE_COUNT = 0x81 SAMPLE_POINT = 0x3F FRACDIV_L = 0x44 FRACDIV_M = 0x1
3	Switch Register to divisor mode (Register MAP condition 2) to do divisor latch setting. UARTn_LCR[7] = 1	LCR = 0x80
4	Set divisor latch	DLL = 0x2 DLM = 0x0
5	Set guard time	GUARD
6	Switch Register to normal mode (Register MAP condition 1). UARTn_LCR[7] = 0	LCR = 0x00

Table 5-35 UART HW Initialization

Step	Description	Related Register Setting
1	Baud rate setting: please refer to Table 5-34	-
2	Enable enhanced feature (Register is accessible only when LCR = BF'h)	LCR = 0xBF EFR = 0x10 LCR = 0x00
3	Enable FIFO control	FCR
4	Word length (LCR[1:0]), parity (LCR[5:4]), STOP (LCR[2]) bit settings	LCR
5	Enable Interrupt	IER

The suggested ED SW programming sequence is shown below.

1. DRV_WriteReg32(UART_BASE+0x24, 0x00000003); //high speed uart
2. DRV_WriteReg32(UART_BASE+0x28, 0x00000081); //sample_count
3. DRV_WriteReg32(UART_BASE+0x2C, 0x0000003F); //sample_point
4. DRV_WriteReg32(UART_BASE+0x4C, 0x00000001); //Enable RX DMA
5. DRV_WriteReg32(UART_BASE+0x54, 0x00000044); //FRACDIV_L
6. DRV_WriteReg32(UART_BASE+0x58, 0x00000001); //FRACDIV_M
7. DRV_WriteReg32(UART_BASE+0x0C, 0x000000BF); //LCR==0xBF, change to Condition 2
8. DRV_WriteReg32(UART_BASE+0x00, 0x00000002); //DLL, LS
9. DRV_WriteReg32(UART_BASE+0x04, 0x00000000); //DLM, MS
10. DRV_WriteReg32(UART_BASE+0x08, 0x00000010); // Enable enhancement features
11. DRV_WriteReg32(UART_BASE+0x0C, 0x00000000); //LCR !=0xBF, change to Condition 1
12. DRV_WriteReg32(UART_BASE+0x08, 0x00000031); //FIFO trigger threshold and enable FIFO
13. DRV_WriteReg32(UART_BASE+0x0C, 0x00000003); //8 bits word length
14. DRV_WriteReg32(UART_BASE+0x04, 0x00000001); //Enable RX interrupt

Data Transmission

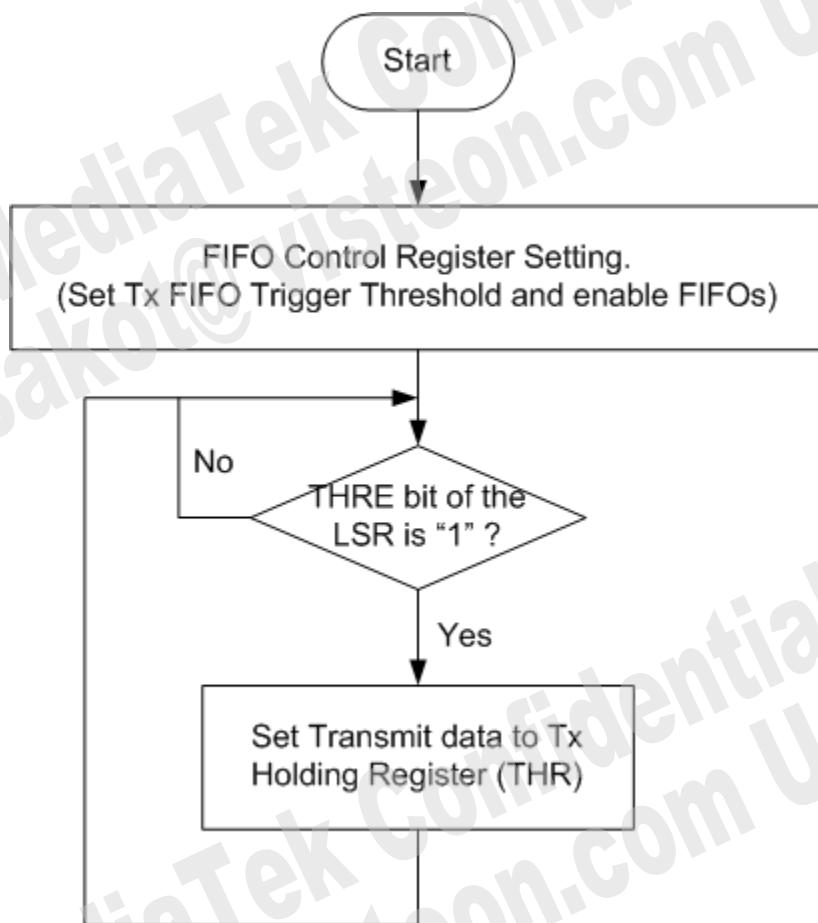


Figure 5-13 UART Data Transmission with Polling the THRE Bit Status

FIFO Register Setting:

- Enable FIFO: Set x'08[0] to 1'b1
- TX FIFO Trigger Threshold x'08[5:4]
 - 2'b00: 1 byte (down to trigger)
 - 2'b01: 4 bytes (down to trigger)
 - 2'b10: 8 bytes (down to trigger)
 - 2'b11: 14 bytes (down to trigger)

Data Reception

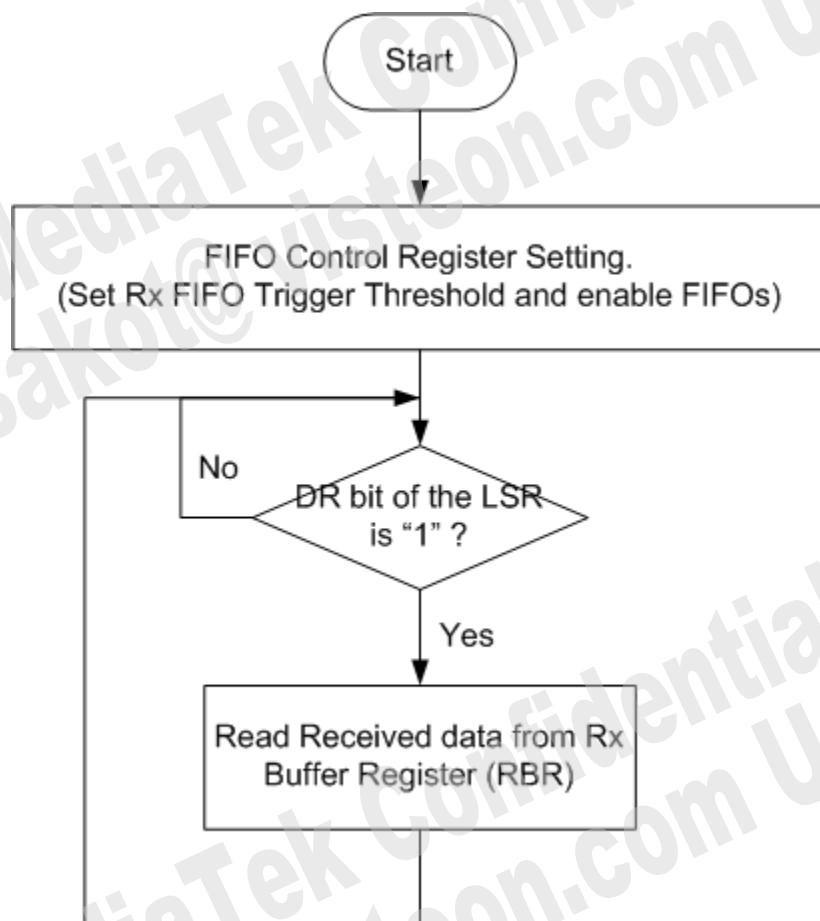


Figure 5-14 UART Data Reception with Polling the DR Bit Status

FIFO Register Setting:

- Enable FIFO: Set x'08[0] to 1'b1
- RX FIFO Trigger Threshold x'08[7:6]
 - 2'b00: 1 byte (up to trigger)
 - 2'b01: 6 bytes (up to trigger)
 - 2'b10: 12 bytes (up to trigger)
 - 2'b11: register x'50[4:0] (up to trigger)

5.6 USB Host Controller

5.6.1 Introduction

MT2712 has two USB host controllers that each one provides one USB3.1 Gen 1 host port and two USB2.0 host ports. The operation model of the host functional blocks conforms to eXtensible Host Controller Interface (xHCI) for Universal Serial Bus (USB) specification.

5.6.2 Features

- Hardware supports USB3.1 SS (Super-Speed) Gen1 with 5 Gb/s TX and 5 Gb/s RX bandwidth
- Hardware supports USB2.0 with LS (Low-speed) 1.5 Mbps/FS (Full-speed) 12 Mbps/HS (High-speed) 480 Mbps
- Embedded USB3.1 Gen 1 PHY with 32-bit/125 MHz PIPE interface
- Embedded USB2.0 PHY with 16-bit/30 MHz UTMI interface
- AHB interface for register access
- AXI interface for DMA access
- xHCI-based host controller
- LPM (Lower power management) on USB2.0 port
- U0/U1/U2/U3 state on USB3.1 Gen 1 port
- Dedicated DMA channel for USB3.1 data transfer
- Support all USB compliant data transfer types with control/bulk/interrupt/isochronous transfer and split transactions
- Compatible to connect to USB2.0/USB3.0 Hub
- Maximum of 15 devices
- Maximum of 64 endpoints configuration

5.6.3 Block Diagram

Figure 5-15 illustrates the architecture of SSUSB (Super-Speed Universal Serial Bus) host. It has one port configured as USB 3.1 Gen 1 host mode and two ports configured as USB2.0 host mode. Each PHY has its own MAC to handle protocol packets.

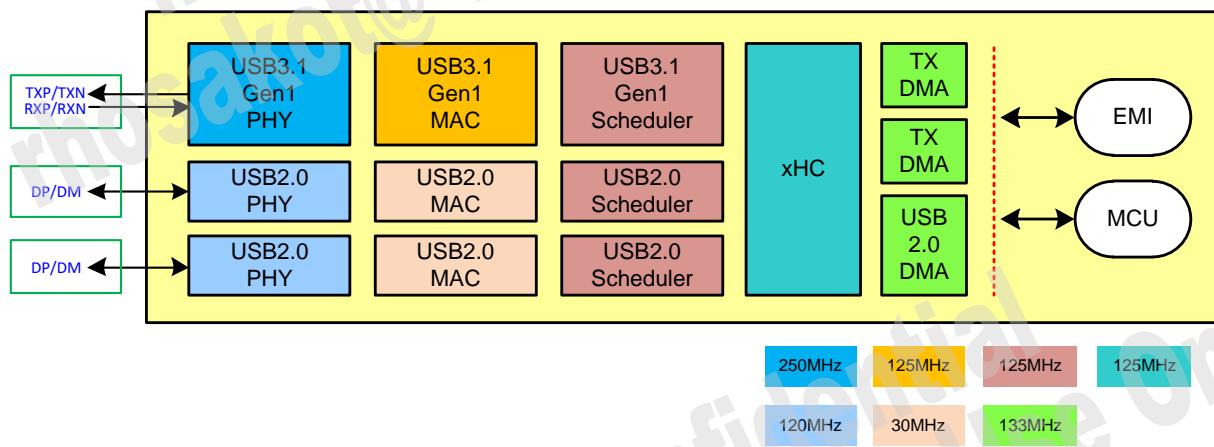


Figure 5-15 USB Host Architecture

All the resources of endpoint and device are handled by xHCI controller. Software could dynamically allocate resources for different ports and turn on/off each port separately. The dedicated DMA channel of USB3.1 Gen 1 port has high transmission rate up to 5 Gb/s TX and 5 Gb/s RX.

5.6.4 Register Definition

For register details, please refer to Chapter 3.5 of “MT2712 USB Registers”.

5.6.5 Programming Guide

This section provides the following operating sequence.

5.6.5.1 USB Host Initialization

After Power-On-Reset (POR), USB is powered down by default to save power.

Software has to execute the following steps before using USB.

Table 5-36 USB Host Initialization Flow

Step	Register Name	Local Address	R/W	Value	Description
USB Initialization					
1	SSUSB_IP_PW_CTRL0	SSUSB_IP_SW_RST [0]	W	1'b0	USB Software Reset When this bit is set, the whole USB is reset. Write “0” to release reset.
2	SSUSB_IP_PW_CTRL1	SSUSB_IP_HOST_PDN [0]	W	1'b0	This bit is USB power-down bit. Write “0” to disable USB power-down.
Enable USB3.1 Gen 1 Port					
3	SSUSB_U3_CTRL_OP	SSUSB_U3_PORT_DIS [0]	W	1'b0	USB3.0 port0 disable bit “0”: USB3.0 port0 is enabled.
4	SSUSB_U3_CTRL_OP	SSUSB_U3_PORT_PDN [1]	W	1'b0	USB3.0 port0 power-down bit “0”: USB3.0 port0 is powered on.
5	SSUSB_U3_CTRL_OP	SSUSB_U3_PORT_HOST_SEL [2]	W	1'b1	This bit is Host Mode selection of USB3.0 port0. “1”: This port is selected for Host Mode.
Enable USB2.0 Port0					
6	SSUSB_U2_CTRL_OP	SSUSB_U2_PORT_DIS [0]	W	1'b0	USB2.0 port0 disable bit “0”: USB2.0 port0 is enabled.
7	SSUSB_U2_CTRL_OP	SSUSB_U2_PORT_PDN [1]	W	1'b0	USB2.0 port0 power-down bit “0”: USB2.0 port0 is powered on.
8	SSUSB_U2_CTRL_OP	SSUSB_U2_PORT_HOST_SEL [2]	W	1'b1	This bit is Host Mode selection of USB2.0 port0. “1”: This port is selected for Host Mode.
Enable USB2.0 Port1					
9	SSUSB_U2_CTRL_1P	SSUSB_U2_PORT_DIS_1P [0]	W	1'b0	USB2.0 port0 disable bit “0”: USB2.0 port0 is enabled.
10	SSUSB_U2_CTRL_1P	SSUSB_U2_PORT_PDN_1P [1]	W	1'b0	USB2.0 port0 power-down bit

Step	Register Name	Local Address	R/W	Value	Description
					"0": USB2.0 port0 is powered on.
11	SSUSB_U2_CTRL_1P	SSUSB_U2_PORT_HOST_SEL_1P [2]	W	1'b1	This bit is Host Mode selection of USB2.0 port0 "1": This port is selected for Host Mode.
Check reference clock stability before software proceeds (i.e., Software proceeds after the following status bits are asserted)					
12	SSUSB_IP_PW_STS1	SSUSB_SYSPLL_STABLE [0]	R	-	When this bit is 1'b1, SYSPLL for USB is stable.
13	SSUSB_IP_PW_STS1	SSUSB_REF_RST_B_STS [8]	R	-	When this bit is 1'b1, it means that reset for reference clock(ref_ck) domain is inactive.
14	SSUSB_IP_PW_STS1	SSUSB_SYS125_RST_B_STS [10]	R	-	When this bit is 1'b1, it means that reset for sys125_ck domain is inactive.
15	SSUSB_IP_PW_STS1	SSUSB_XHCI_RST_B_STS [11]	R	-	When this bit is 1'b1, it means that reset for xhci_ck domain is inactive.
16	SSUSB_IP_PW_STS1	SSUSB_U3_MAC_RST_B_STS [16]	R	-	USB3.0 port0 PIPE clock status bit. When this bit is 1'b1, it means that reset for PIPE clock domain is inactive.

5.6.5.2 Difference between MTK xHC and Standard xHCI

There are some differences between MTK eXtensible Host Controller (xHC) and standard xHCI. These differences fall into two categories.

- Some Transfer Request Block (TRB) types defined in xHCI specification are not implemented in MTK xHC.
- Scheduling mechanism of synchronous endpoint is proposed to simplify hardware design.

5.6.5.2.1 TRB Types Not Implemented in MTK xHC

The following TRB types are not implemented in MTK xHC:

- Force Event Command TRB: (TRB Type = 'd18, Optional Normative)
- Negotiate Bandwidth Command TRB: (TRB Type = 'd19, Optional Normative)
- Set Latency Tolerance Value Command TRB: (TRB Type = 'd20, Optional Normative)
- Get Port Bandwidth Command TRB: (TRB Type = 'd21)
- Bandwidth Request Event TRB: (TRB Type = 'd35)
- Doorbell Event TRB: (TRB Type = 'd36)

5.6.5.2.2 Scheduling of Synchronous Endpoint

A proprietary Scheduling Algorithm is proposed to simplify the hardware design for Bandwidth Calculation and Scheduling on Synchronous Endpoint.

To implement this algorithm for MTK eXtensible Host Controller Driver (xHCD), the standard Linux xHCD driver has to be patched. The patch includes the following two steps:

1. Calculate if there is enough bandwidth reserved for the endpoint(s) to be added.
2. Decide a set of parameters which specify the scheduling for synchronous endpoint(s) to be added.

5.6.5.2.3 Bandwidth Calculation

Because bandwidth calculation is offloaded by xHCD, the MTK xHC can process the following xHCI Commands easier:

- Assume there is always enough bandwidth for endpoint(s) to be added by Configure Endpoint Command.
- Simply execute the Reset Device, Configure Endpoint (with DC = 1 of Drop Flag(s) = 1) and Disable Slot Command.

The xHCD will never place the Get Port Bandwidth Command TRB on command ring for the MTK xHC because all the bandwidth information is visible to xHCD directly.

5.6.5.2.4 Decide the Software Scheduling Parameters

To allow the xHC to schedule synchronous endpoints easily, the xHCD implements the proprietary scheduling algorithm before placing the Configure Endpoint TRB to Command TRB.

A set of parameters are defined for each device slot and the related synchronous endpoint. To bring such proprietary parameters to MTK xHC, some reserved fields of the endpoint context are now used to carry the values of these parameters.

Because there are still some reserved dual words (DWs) in the Endpoint Context defined in xHCI specification, these extra software scheduling parameters are brought to xHC for related endpoints through the commands originally defined in xHCI specification. The modified Endpoint Context is shown in Table 5-37 with the extra defined fields in the shade of orange.

Table 5-37 Endpoint Context

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																							
Max ESIT Payload Hi		Interval		LSA		MaxPStreams		Mult		RsvdZ		EP State																03-00H																										
Max Packet Size		Max Burst Size		HID		RZ		EP Type		CErr		RZ																07-04H																										
TR Dequeue Pointer Lo		RsvdZ		DCS																													08-08H																					
TR Dequeue Pointer Hi																													0F-0CH																									
Max ESIT Payload Lo		Average TRB Length																											13-10H																									
RsvdO		bBM		bCSCount		RsvdO		bPkts																										17-14H																				
RsvdO	bRepeat		RsvdO		bOffset																											1B-18H																						
RsvdO																													1F-1CH																									

The definition of extra defined fields in Endpoint Context is shown in Table 5-38.

Table 5-38 Extra Defined Fields for Endpoint Context

Field Name	Location	Definition
bPkts[5:0]	DW5[5:0]	Number of packets to be transferred in the scheduled micro frame(s).
bCSCount[2:0]	DW5[10:8]	<p>The number of CS (Complete Spilt) that host will trigger. This field is only for split transaction for FS/LS.</p> <p>For FS/LS Isochronous IN and Interrupt endpoints, this represents the pre-defined number of CS to do in a service interval.</p> <p>For FS/LS Isochronous OUT EPs, this represents the pre-defined number of SS (Start Spilt) to do in a service interval.</p>
bBM (bBurstMode)	DW5[11]	<p>Burst mode for scheduling 0: Normal burst mode. Distribute the bMaxBurst+1 packets for a single burst. 1: Distribute the $(bMaxBurst+1) \times (Mult+1)$ packets according to bPkts and bRepeat.</p>
bOffset[14:0]	DW6[14:0]	Which micro frame of the interval that transfer should be scheduled first within the interval.
bRepeat[14:0]	DW6[30:16]	The time gap between two micro frames that transfers are scheduled within an interval.

The relation between extra defined field parameters and micro frames is illustrated as Figure 5-16.

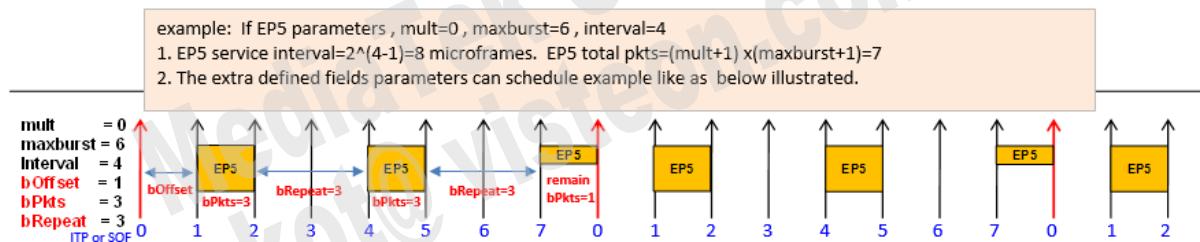


Figure 5-16 Relation between Extra Defined Parameters and Micro Frames

The relation between extra defined field parameters and micro frames for Split Transaction for FS/LS is illustrated as Figure 5-17.

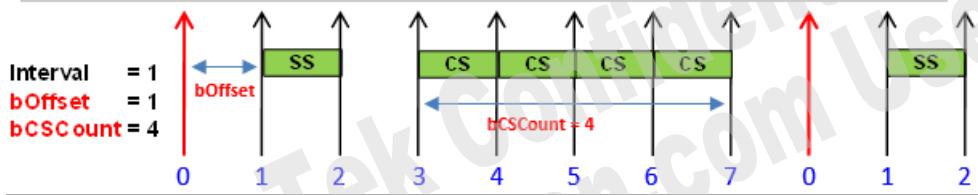
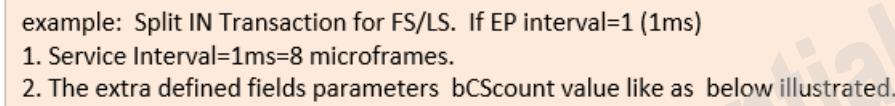


Figure 5-17 Relation between bCSCount Parameters and Micro Frames

The software flow of “`xhci_add_endpoint`” configuration by a standard xHCI driver is illustrated as the (a) part in Figure 5-18. To add extra defined field parameters to endpoint context, a sub-flow is patched. This patched sub-flow is marked by red dash line in (b) part in Figure 5-18.

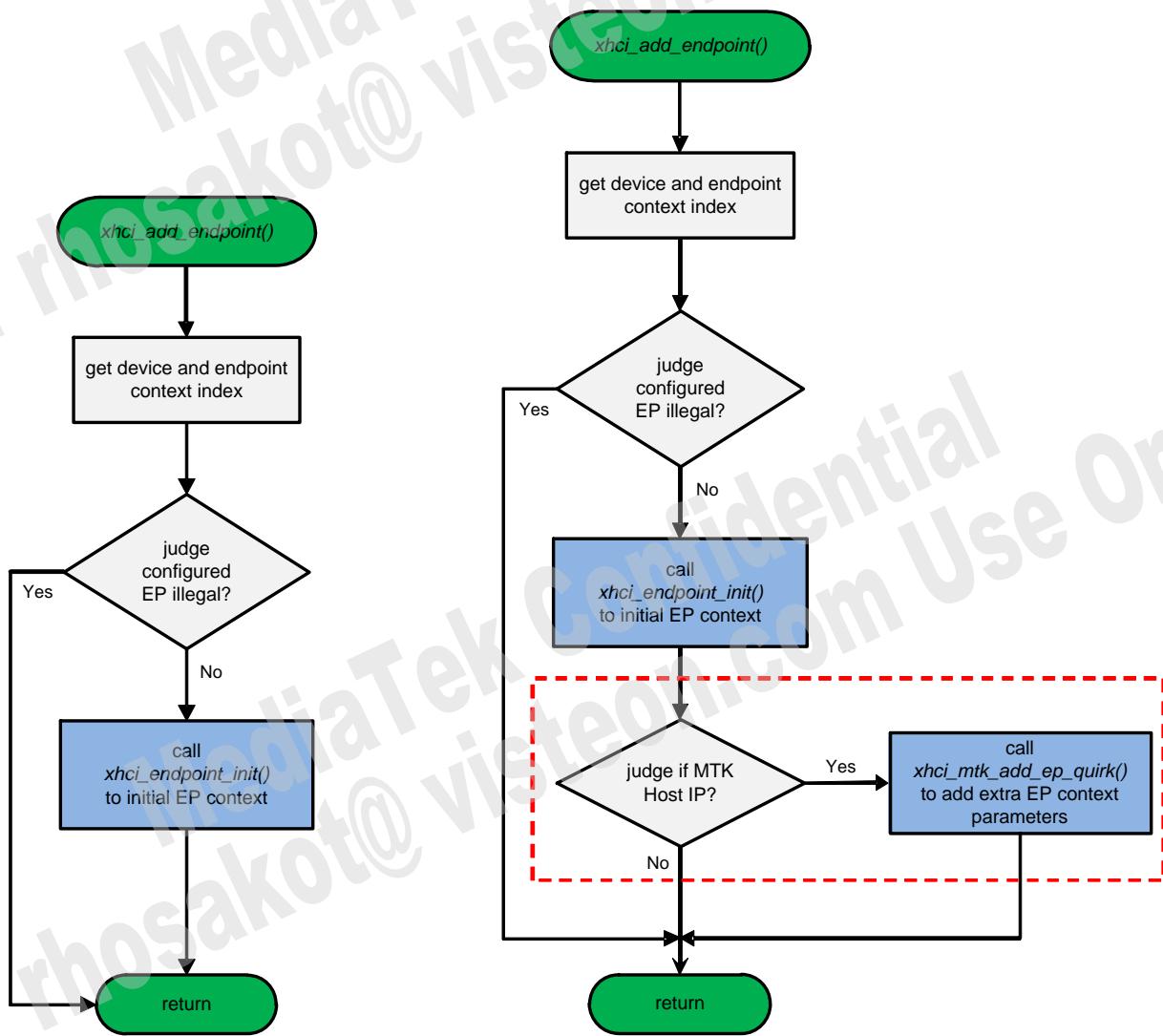


Figure 5-18 (a) Standard xHCI_add_endpoint () Flow (b) Patched xHCI_add_endpoint () Flow

The software flow of “`xhci_drop_endpoint`” configuration by a standard xHCI driver is illustrated as the (a) part in Figure 5-19. To drop software recorded extra defined field parameters, a sub-flow is patched. This patched sub-flow is marked by red dash line in (b) part in Figure 5-19.

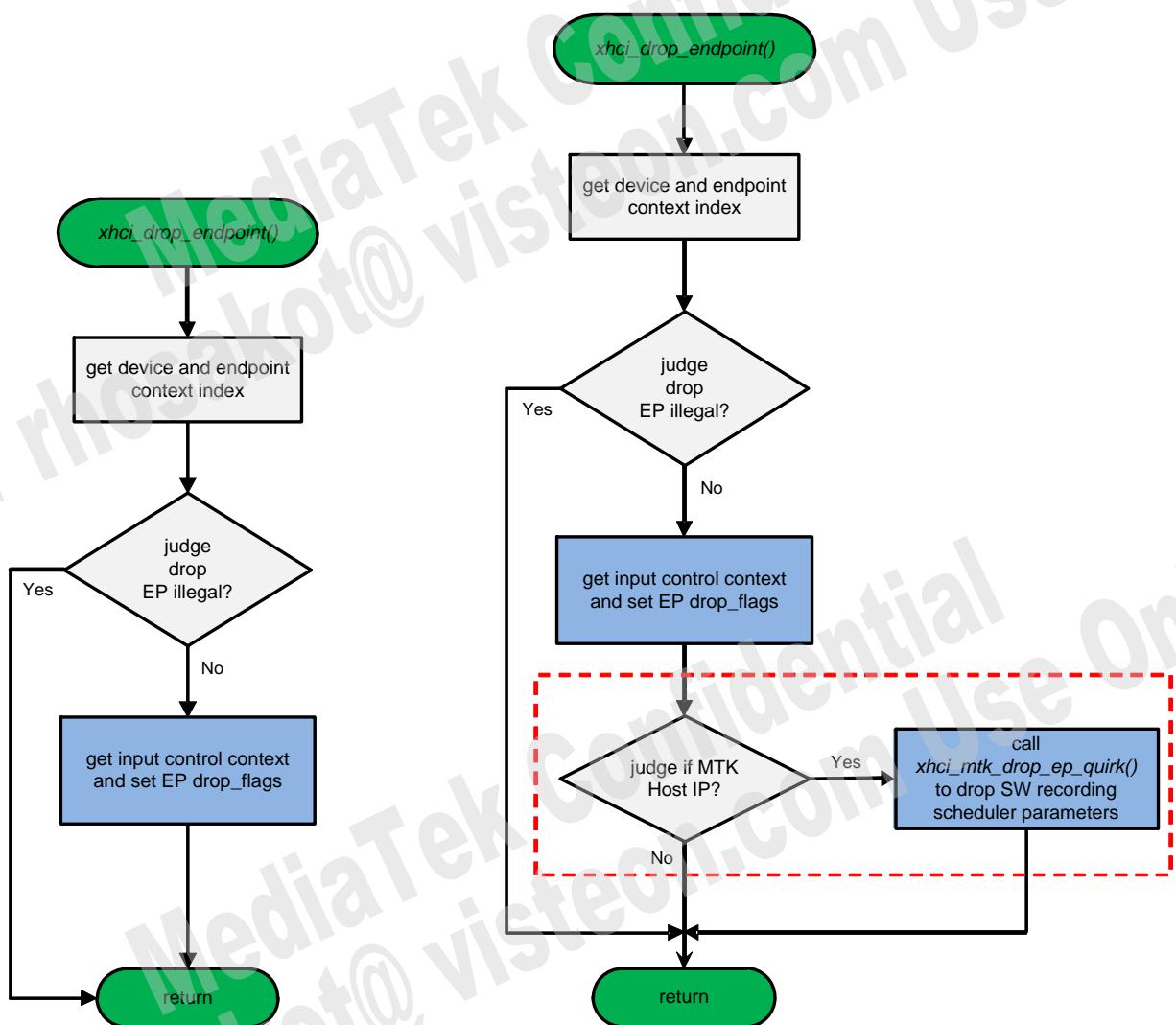


Figure 5-19 (a) Standard `xHCI_drop_endpoint()` Flow (b) Patched `xHCI_drop_endpoint()` Flow

5.7 USB Device Controller

5.7.1 Introduction

System can be completely turned off to improve the system power consumption without losing interrupts.

5.7.2 Features

- Hardware supported USB2.0 HS/FS
- Embedded USB2.0 PHY with 16-bit/30 MHz UTMI interface
- AHB interface for register access

- AXI interface for DMA access
- Embedded queue management function with scatter/gather DMA capability
- Proprietary application layer device controller with linked list queue and scatter/gather DMA
- LPM on USB2.0 port
- Hardware configurable up to 8 OUT Endpoints and 8 IN Endpoints
- Hardware configurable up to 4 packet slots for each endpoint separately
- Software configurable FIFO size allocation for each endpoint separately
- 8 KB of on-chip data RAM
- Software configurable transfer type to Bulk/Interrupt/Isochronous for each endpoint
- Software configurable Interrupt with the following interrupt status: VBUS On/Off, suspend/resume and USB Reset

5.7.3 Block Diagram

Figure 5-20 illustrates the architecture of SSUSB Device. It has a pair of MAC and PHY to handle protocol packets.



Figure 5-20 USB Device Architecture

5.7.4 Register Definition

For register details, please refer to Chapter 3.5 of “MT2712 IVI Application Processor Registers”.

5.7.5 Programming Guide

This section provides the following operating sequence:

5.7.5.1 USB Device Initialization

After Power-On-Reset, USB is powered down by default to save power.

Software has to execute the following steps before using USB.

Table 5-39 USB Device Initialization Flow

Step	Register Name	Local Address	R/W	Value	Description
Slew Rate Calibration for USB2.0 Port					
1	SSUSB_IP_PW_CTRL	SSUSB_IP_SW_RST [0]	W	1'b0	USB Software Reset When this bit is set, the whole USB is reset.

Step	Register Name	Local Address	R/W	Value	Description
					Write "0" to release reset.
USB Initialization					
2	SSUSB_IP_PW_CTRL_0	SSUSB_IP_SW_RST [0]	W	1'b0	USB Software Reset When this bit is set, the whole USB is reset. Write "0" to release reset.
3	SSUSB_IP_PW_CTRL_2	SSUSB_IP_DEV_PDN[0]	W	1'b0	SSUSB IP Dev Power Down When this bit is set, whole CKBG can be powered down. Write "0" to release reset.
Enable USB2.0 Port					
4	SSUSB_U2_CTRL_OP	SSUSB_U2_PORT_DIS [0]	W	1'b0	USB2.0 port0 disable bit "0": USB2.0 port0 is enabled.
5	SSUSB_U2_CTRL_OP	SSUSB_U2_PORT_PDN [1]	W	1'b0	USB2.0 port0 power-down bit "0": USB2.0 port0 is powered on.
6	SSUSB_U2_CTRL_OP	SSUSB_U2_PORT_HOST_SEL [2]	W	1'b0	This bit is Host Mode selection of USB2.0 port0. "0": This port is selected for Device Mode.
Check reference clock stability before software proceeds (i.e., Software proceeds after the following status bits are asserted)					
7	SSUSB_IP_PW_STS1	SSUSB_SYS125_RST_B_STS [10]	R	-	When this bit is 1'b1, it means that reset for sys125_ck domain is inactive.
8	SSUSB_IP_PW_STS2	SSUSB_U2_MAC_SYS_RST_B_STS [0]	R	-	When this bit is 1'b1, it means that reset for mac2_sys_ck domain is inactive.

Note on USB2.0 port power-down:

Registers in SSUSB_USB2 cannot be read correctly when SSUSB_U2_PORT_PDN or SSUSB_U2_PORT_DIS = 1

5.7.5.2 USB Endpoint Initialization

5.7.5.2.1 EPO Initialization

Endpoint 0 hardware should be configured properly before a successful USB enumeration.

Endpoint 0 is controlled by USB only and software fills in the correct value into EPO CSR. The EPO initialization is included in our USB initial flow. It is recommended that it should not be modified, except for the maximum packet size.

Table 5-40 Suggested EPO Programming Sequence

Step	Register Name	Local Address	R/W	Value	Description
1	EPOCSR	EPO_MAXPKTSZ0[9:0]	W	USER DEFINED	Setting maximum packet size e.g. 10'd64
2	EPIESR	EPOIESR[0]	W	1'b1	Endpoint 0 interrupt enable Set
3	EPIESR	SETUPPENDIESR[16]	W	1'b1	Endpoint 0 Setup End interrupt enable Set

5.7.5.2.2 EPn Initialization

Table 5-41 Suggested TX EPn Programming Sequence (e.g. EP1)

Step	Register Name	Local Address	R/W	Value	Description
1	TX1CSR0	TX_TXMAXPKTSZ[10:0]	W	USER DEFINED	Setting maximum packet size e.g. 11'd64
2	TX1CSR1	SS_TX_BURST[3:0]	W	USER DEFINED	Set burst size
3	TX1CSR1	TX_MULT[23:21]	W	USER DEFINED	Set TX_MULT size
4	TX1CSR1	TX_MAX_PKT[30:24]	W	USER DEFINED	Number of packets = (SS_TX_BURST+ 1) x (TX_MULT + 1) - 1 (Isochronous endpoint only)
5	TX1CSR2	TXFIFOADDR[12:0]	W	USER DEFINED	Start address of the selected TX endpoint FIFO
6	TX1CSR1	TX_SLOT[13:8]	W	USER DEFINED	Set slot number of hardware
7	TX1CSR2	TXFIFOSEGSIZE[19:16]	W	USER DEFINED	Set FIFO segment size of hardware layout. This register indicates the TX FIFO size of 2^n bytes
8	TX1CSR1	TXTYPE[5:4]	W	USER DEFINED	Select the required transfer type for the TX endpoint. 2'b00: Bulk 2'b01: Interrupt 2'b10: Isochronous
9	TX1CSR2	TXBINTERVAL[31:24]	W	USER DEFINED	Interval for servicing the endpoint for Isochronous/Interrupt data transfer
10	TX1CSR0	TX_DMAREQEN[29]	W	1'b1	Enable EP1 DMA request for the TX endpoint
11	QIESRO	TXQ_DONE_IESR[1]	W	1'b1	Enable EP1 TX QMU Done interrupt

Table 5-42 RX EPn Programming Sequence (e.g. EP2)

Step	Register Name	Local Address	R/W	Value	Description
1	RX2CSR0	RX_RXMAXPKTSZ[10:0]	W	USER DEFINED	Setting maximum packet size e.g. 11'd64
2	RX2CSR1	SS_RX_BURST[3:0]	W	USER DEFINED	Set burst size
3	RX2CSR1	RX_MULT[23:21]	W	USER DEFINED	Set RX_MULT size
4	RX2CSR1	RX_MAX_PKT[30:24]	W	USER DEFINED	Number of packets = (SS_RX_BURST+ 1) x (RX_MULT + 1) - 1 (Isochronous endpoint only)
5	RX2CSR2	RXFIFOADDR[12:0]	W	USER DEFINED	Start address of the selected RX endpoint FIFO
6	RX2CSR1	RX_SLOT[13:8]	W	USER DEFINED	Set slot number of hardware

Step	Register Name	Local Address	R/W	Value	Description
7	RX2CSR2	RXFIFOSEGSIZE[19:16]	W	USER DEFINED	Set FIFO segment size of hardware layout. This register indicates the RX FIFO size of 2^n bytes
8	RX2CSR1	RX_TYPE[5:4]	W	USER DEFINED	Select the required transfer type for the RX endpoint. 2'b00: Bulk 2'b01: Interrupt 2'b10: Isochronous
9	RX2CSR2	RXBINTERVAL[31:24]	W	USER DEFINED	Interval for servicing the endpoint for Isochronous/Interrupt data transfer
10	RX2CSR0	RX_DMAREQEN[29]	W	1'b1	Enable EP2 DMA request for the RX endpoint
11	QIESR0	TXQ_DONE_IESR [18]	W	1'b1	Enable EP2 QMU Done interrupt

5.7.6 EPO Programming Guide

EPO control is a state machine with 3 states: IDLE, TX Mode, and RX Mode.

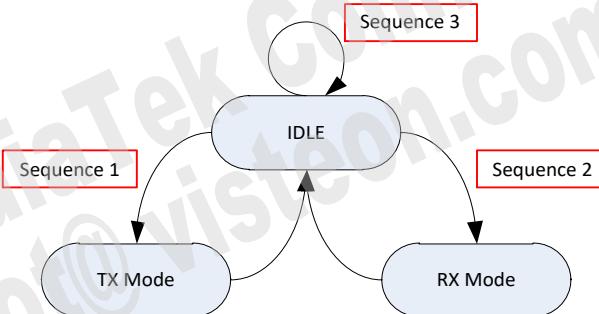


Figure 5-21 EPO State

5.7.6.1 EPO Programming Flow (IDLE)

1. After power-on or reset, EPO is in “IDLE” state.
2. After receiving a SETUP transaction
 - EPOCSR.SetupPktRdy is set.
 - EPOCSR.DPHTX is cleared.
 - An interrupt is generated to notify software.
3. Software unloads FIFO and decodes command.
4. Depending on command type, software can

- (Sequence 1/IN DATA) W1C(Write One Clear) *EPOCSR.SetupPktRdy* and set *EPOCSR.DPHTX*. EPO state goes into “TX Mode”.
- (Sequence 2/OUT DATA) W1C *EPOCSR.SetupPktRdy*. EPO goes into “RX Mode”.
- (Sequence 3/NO DATA) process command. Then, W1C *EPOCSR.SetupPktRdy* and set *EPOCSR.DataEnd* (or set *EPOCSR.SendStall*) at the same time. EPO state stays in “IDLE” state.

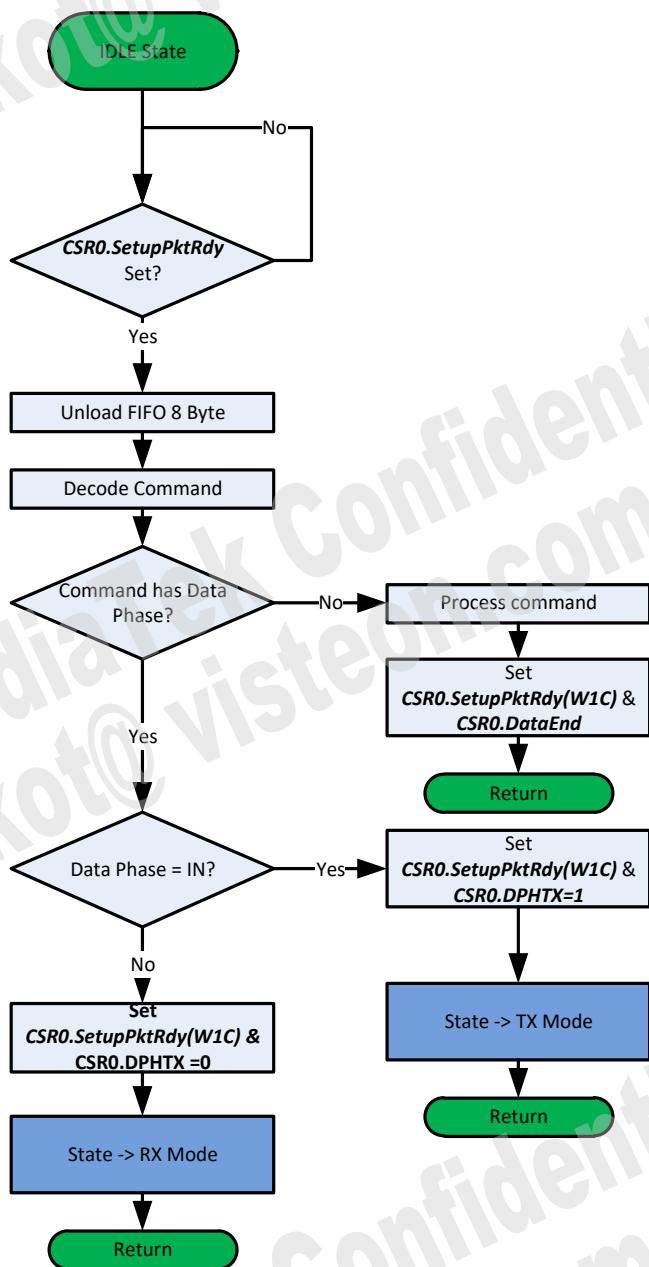


Figure 5-22 EPO IDLE Flow Chart

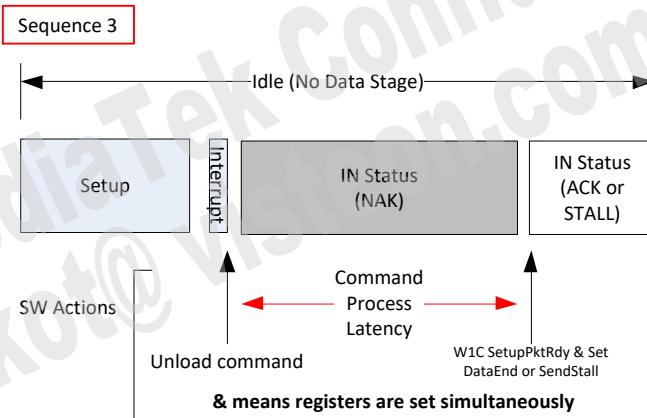


Figure 5-23 EPO IDLE

5.7.6.2 EPO Programming Flow (TX Mode)

Normal flow

- Software loads data packet ($\leq EPOCSR.MaxPktSz0$) to *FIFO0*, and sets *EPOCSR.TxPktRdy* to send it to host.
- If *EPOCSR.AutoSet* is set, software only needs to set *EPOCSR.TxPktRdy* for the last data packet that is a short packet.
- After sending data packet to host, *EPOCSR.FIFOFull* is cleared and an interrupt is generated to notify software.
- SW repeats the steps above until the required amount of data is sent, and set *EPOCSR.DATAEND* to leave DATA phase.

Error cases

- If a SETUP transaction is received in “TX Mode” (DATA phase), *EPOCSR.SetupEnd* is set and an interrupt is generated. Software aborts the current command and moves on to decode the new command.

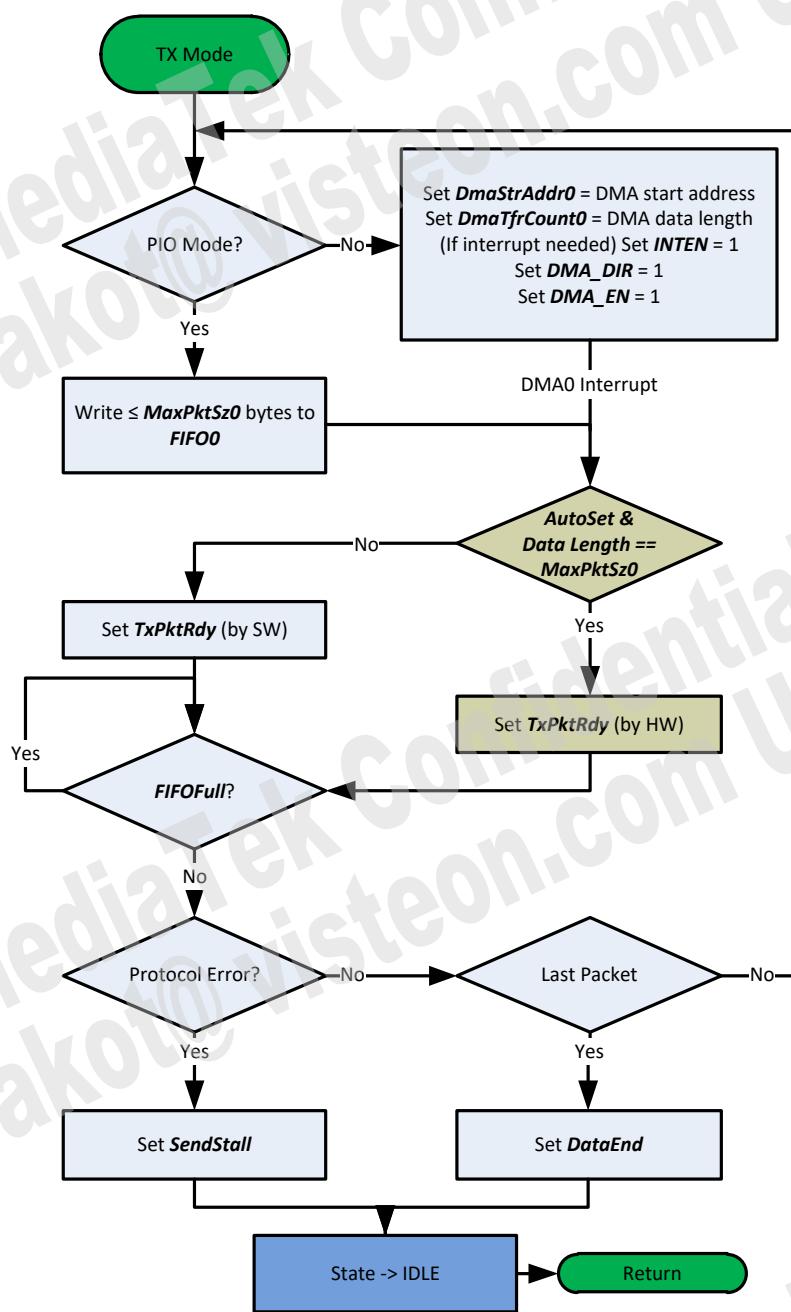


Figure 5-24 EPO TX Mode Flow Chart

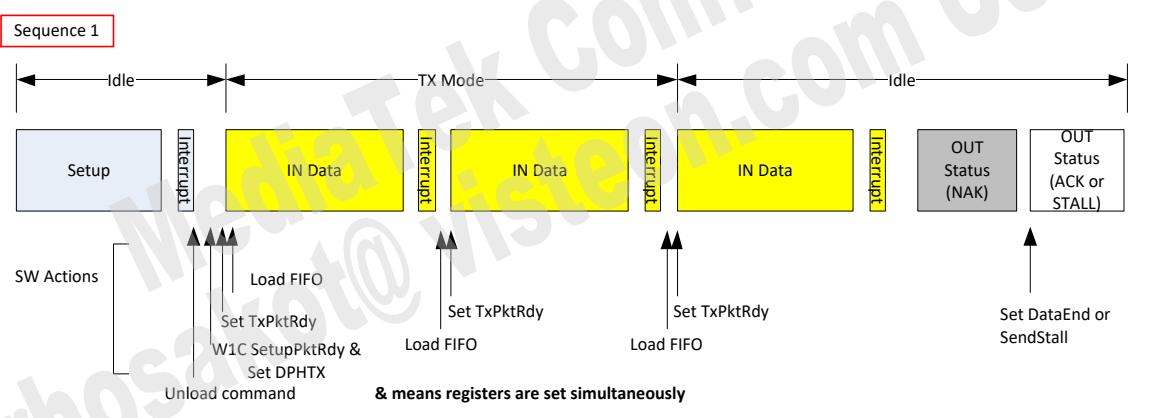


Figure 5-25 EPO TX Mode Flow Chart

5.7.6.3 EPO Programming Flow (RX Mode)

Normal flow

- After receiving data packet ($\leq EPOCSR.MaxPktSz0$), Hardware sets *EPOCSR.RxPktRdy* and generates an interrupt.
- Software unloads data packet ($\leq EPOCSR.MaxPktSz0$) from *FIFO0*, and W1C *EPOCSR.RxPktRdy*.
- If *EPOCSR.AutoClear* is set, *EPOCSR.RxPktRdy* is cleared automatically after data packet is unloaded, unless the data packet is of size 0. Software needs to write one to clear *EPOCSR.RxPktRdy* under this condition.
- Software repeats the steps until the required amount of data is received or a short packet is received, and set *EPOCSR.DATAEND* to leave DATA phase.

Error cases

- If a SETUP transaction is received in “RX Mode” (DATA phase), *EPOCSR.SetupEnd* is set and an interrupt is generated. Software aborts the current command and moves on to decode the new command.

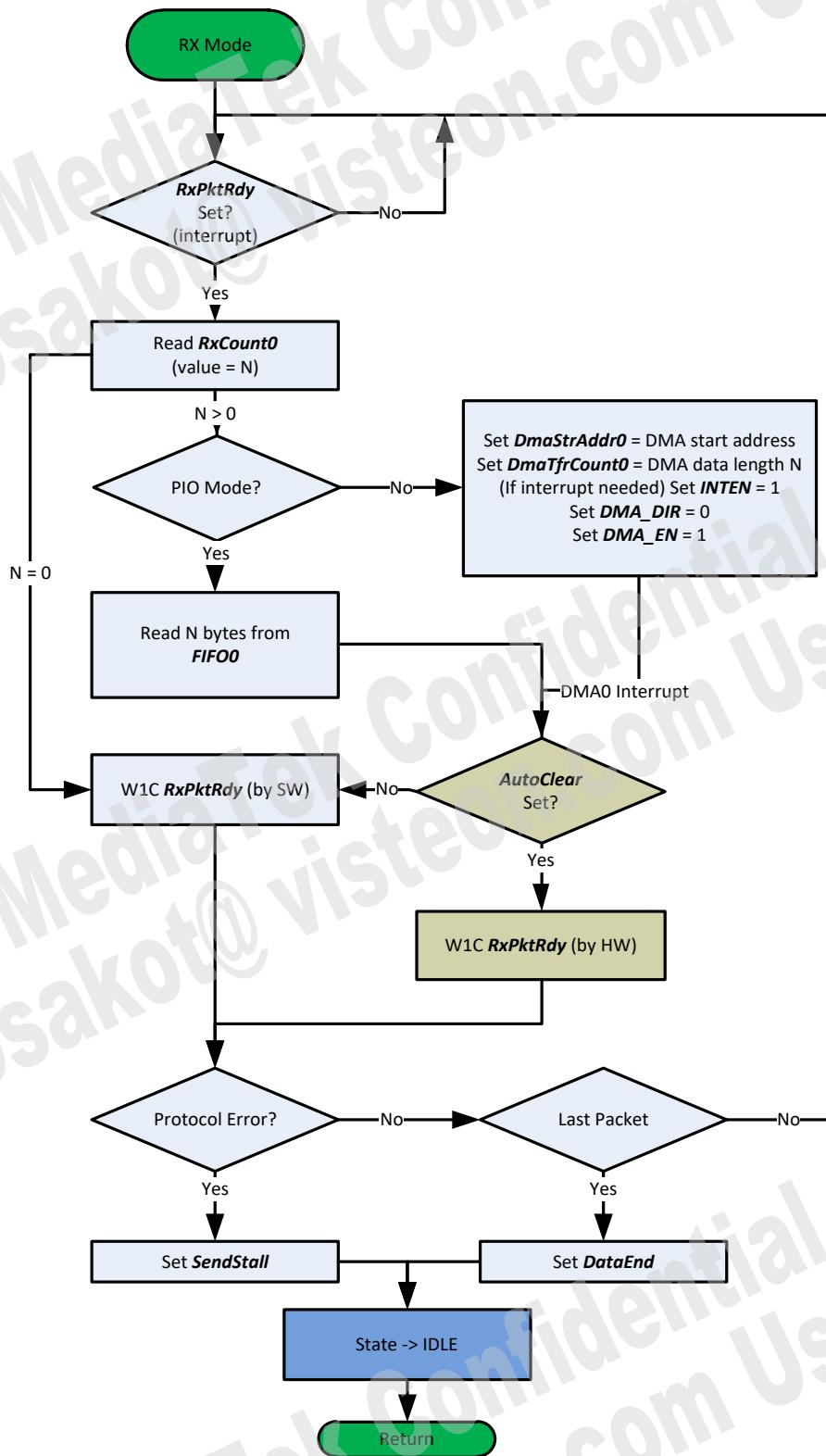


Figure 5-26 EPO RX Mode Flow Chart

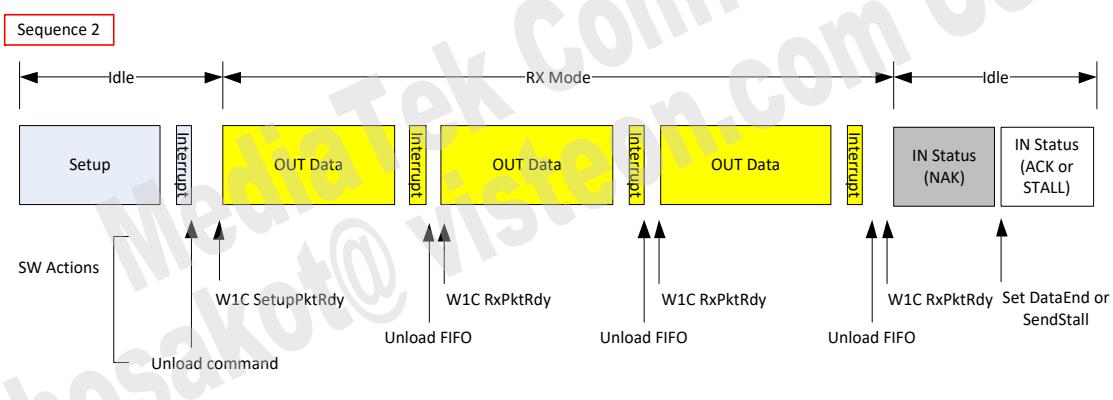


Figure 5-27 EPO IDLE for RX Mode

5.7.6.4 Notes

- EPO interrupt is generated when
 - EPOCSR.RxPktRdy* bit is set after data packet has been received and stored into *FIFO0*.
 - Data packet in *FIFO0* has been sent to host successfully.
 - EPOCSR.SentStall* bit is set after host receives STALL.
 - EPOCSR.SetupEnd* bit is set after receiving SETUP transaction in DATA/STATUS phase.
- EPOCSR.SetupEnd* indicates that the current control transfer is aborted.
 - Set when
 - IN transaction is received after setting *EPOCSR.DataEnd* to leave “TX Mode” (STATUS phase).
 - OUT transaction is received after setting *EPOCSR.DataEnd* to leave “RX Mode” (STATUS phase).
 - SETUP transaction is received in “TX Mode” or “RX Mode” (DATA phase).
 - Clear when software sets *EPOCSR.SetupEnd*.
 - Software can unload *FIFO* and decode the new command.
- EPOCSR.DataEnd* indicates that data phase has finished.
 - Software sets *EPOCSR.DataEnd* when
 - In “TX Mode”/“RX Mode”, the amount of data required by host is sent/received.
 - In “TX Mode”, smaller amount of data required by host is sent. Device has to notify host by sending short packet.
 - In “RX Mode”, a short packet is received.
 - Hardware clears *EPOCSR.DataEnd* when
 - (Normal) Status phase finished successfully.
 - (Error) *EPOCSR.SetupEnd* is set.

5.7.7 QMU Programming Guide

Queue Management Unit (QMU) is designed to unload software effort to serve DMA interrupts. By preparing General Purpose Descriptor (GPD) and Buffer Descriptor (BD). Software links data buffers and triggers QMU to send/receive data to host/from device at a time.

5.7.7.1 GPD/BD Introduction

See Figure 5-28 and Figure 5-29 for the relationship among GPD/BD/bus transfer.

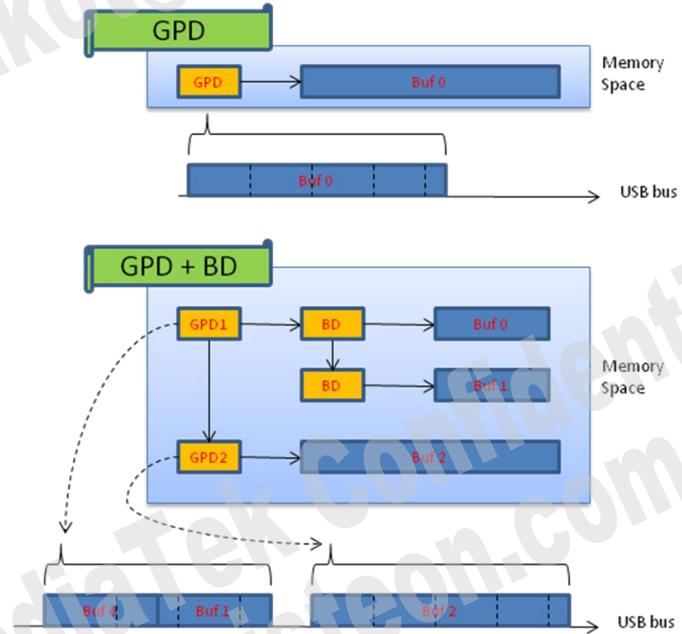


Figure 5-28 GPD/BD/Bus Transfer

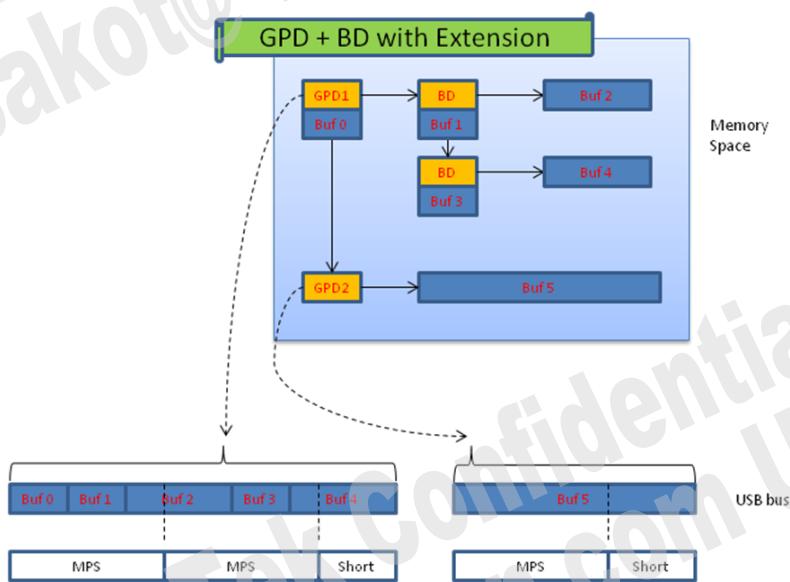


Figure 5-29 GPD/BD with Extension

BD and GPD are briefly described below. Detailed descriptor format is in the next section for full understanding.

BD

- Must contain pointer to a data buffer
- Can link to another BD
- Can have extension. Data is placed in memory immediately after this BD

GPD

- Must contain pointer to a data buffer if this GPD does not link to any BD
- Can link to several BDs
- Can have extension. Data is placed in memory immediately after this GPD
- Map to a transfer on USB bus. Data buffers are concatenated and transferred on USB bus, data packet by data packet. Each data packet is smaller than Maximum Packet Size (MPS)
- The data buffers are concatenated
 - GPD extension
 - GPD data buffer
 - For each BD in the chain
 - BD extension
 - BD data buffer

5.7.7.2 TX GPD/BD Format

The format of TX GPD/BD is shown in Figure 5-30. For the detailed description, please see Section 5.7.7.4 and Section 5.7.7.5 respectively.

Bit Location	B 31	B 24	B 16	B 8	B 0	offset
			GPD Check Sum	I O C		0x00
			Next GPD Pointer			0x04
			BDP=0: Data Buffer BDP=1: BD Pointer			0x08
	Z L P		GPD Extension Length	Data Buffer Length		0x0C
Bit Location	B 31	B 24	B 16	B 8	B 0	offset
			BD Check Sum		E O L	0x00
			Next BD Pointer			0x04
			Data Buffer			0x08
		BD Extension Length	Data Buffer Length			0x0C

Figure 5-30 TX GPD/BD Format

5.7.7.3 RX GPD/BD Format

The RX GPD/BD format is shown in Figure 5-31. For the detailed description, please see Section 5.7.7.4 and Section 5.7.7.5 respectively.

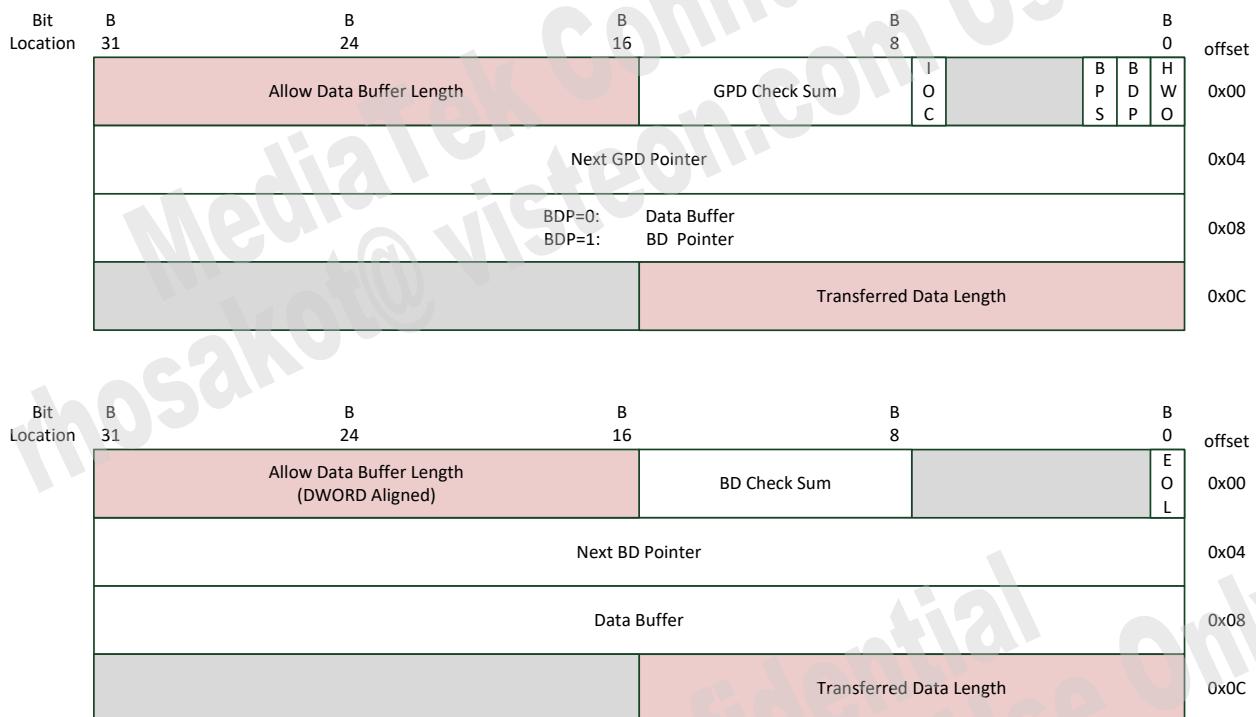


Figure 5-31 RX GPD/BD Format

5.7.7.4 GPD Field Description

- Hardware Own (HWO)
 - Used to indicate the current ownership of this GPD, the associated BD(s) and the associated data buffer(s)
 - 0, Software has the ownership
 - 1, Hardware has the ownership
- Buffer Descriptor Present (BDP)
 - 0, DWORD@0x8 points to a data buffer
 - 1, DWORD@0x8 points to a BD
- Bypass (BPS)
 - 0, Hardware does not skip this GPD if HWO = 1
 - 1, Hardware skips this GPD if HWO = 1
- Interrupt On Completion (IOC)
 - 0, Hardware does not issue interrupt when this GPD (including the associated BDs) is completed
 - 1, Hardware issues interrupt when this GPD (including the associated BDs) is completed

- GPD Checksum
 - This is used to validate the contents of this GPD
 - If TXQ_CS_EN/RXQ_CS_EN bit is set, an interrupt is issued when checksum validation fails
 - Q_CS16B_EN decides the way checksum value is calculated
 - 0, over the first 12 bytes of this GPD
 - 1, over the first 16 bytes of this GPD
- (RX ONLY) Allow Data Buffer Length
 - This value indicates the length of the assigned data buffer
- Next GPD Pointer
 - Value != 0, points to the next GPD
 - Value = 0, does not point to any GPD
- Data Buffer/BD Pointer
 - Refer to BDP description
- (TX ONLY) Data Buffer Length
 - This value indicates the length of the assigned data buffer
- (RX ONLY) Transferred Data Length
 - After receiving a transfer, the total length of data is written to this field
 - If the total length of data is over 64K, '0' is written to this field. Software has to sum up all BD's "Transferred Data Length" to get the total length of data
- (TX ONLY) GPD Extension Length
 - 0, does not use GPD extension feature
 - 1-255, specify GPD extension buffer size. GPD extension buffer is placed in memory immediately after this GPD
- Zero Length Packet (ZLP)

5.7.7.5 BD Field Description

- End of List (EOL)
 - 0, this is not the last BD in chain. The next BD is pointed by "Next BD Pointer"
 - 1, this is the last BD in chain
- BD Checksum
 - This is used to validate the contents of this BD
 - If TXQ_CS_EN/RXQ_CS_EN bit is set, an interrupt is issued when checksum validation fails

- Q_CS16B_EN decides the way for checksum value to be calculated
 - 0, over the first 12 bytes of this BD
 - 1, over the first 16 bytes of this BD
- (RX ONLY) Allow Data Buffer Length
 - This value indicates the length of the assigned data buffer

Next BD Pointer

- The pointer to the next BD
- Please refer to EOL description

Data Buffer

- Pointer to data buffer

(TX ONLY) Data Buffer Length

- This value indicates the length of the assigned data buffer

(RX ONLY) Transferred Data Length

- After receiving a transfer, the length of data transferred to the data buffer is written to this field

(TX ONLY) BD Extension Length

- 0, does not use BD extension feature
- 1-255, specify BD extension buffer size. BD extension buffer is placed in memory immediately after this BD

5.7.7.6 TX Programming Flow

This TXQ programming flow is shown in Figure 5-32. For the details about blue blocks, please refer to the following sub-sections.

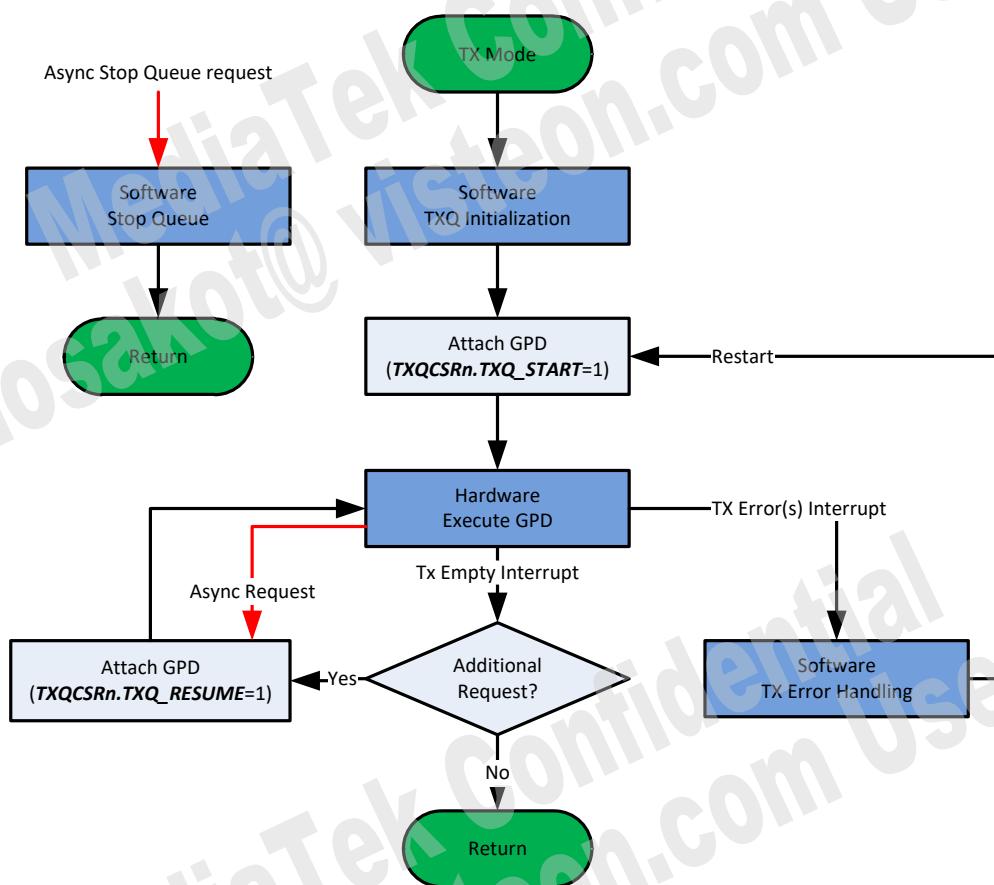


Figure 5-32 TXQ Programming Flow

5.7.7.6.1 Initialization Flow

Figure 5-33 shows which registers should be set before TXQ is used. Please refer to the register map for register setting details.

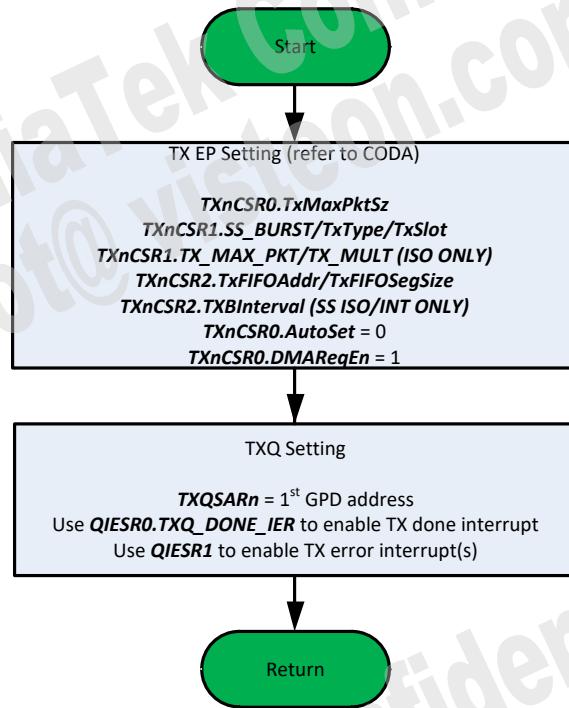


Figure 5-33 TXQ Initialization Flow

5.7.7.6.2 Stop Queue Flow

When TXQ is operating, some cases may occur, as a result, software has to stop the current transfer. There are two ways to notify the host. One is to STALL endpoint to notify the host explicitly; the other is sending short packet to terminate the current transfer to notify the host implicitly. Either way, the host has to handle the error. TXQ restart procedure at device side is application-specific.

5.7.7.6.3 STALL (Recommended)

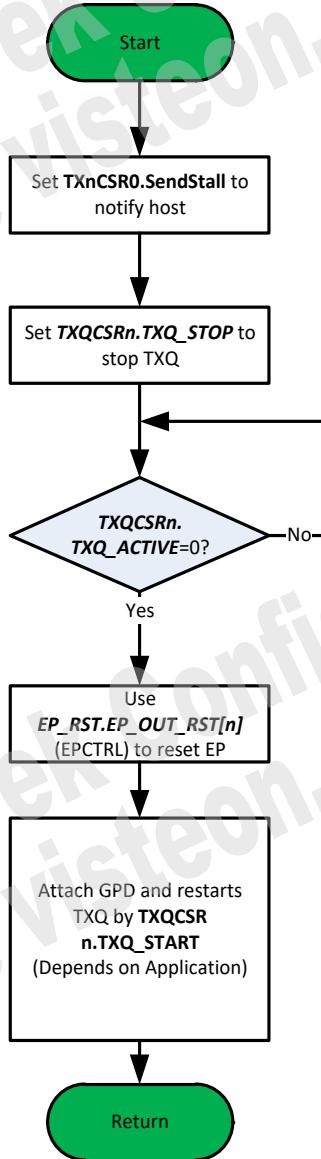


Figure 5-34 TXQ STALL Flow

5.7.7.6.4 Short Packet (!)

In this case, the host will receive a ZLP and terminate IN transfer. Since EPOUT_RST is not used, data fetched before TXQ is stopped and may be sent to the host after TXQ restarts.

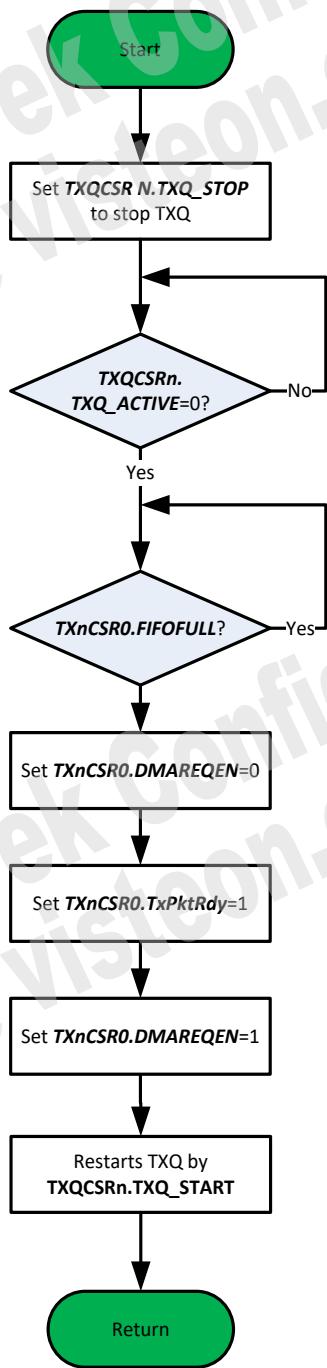


Figure 5-35 TXQ Short Packet (I) Flow

5.7.7.6.5 Short Packet (II)

In this case, device software has to make sure the content of GPD can request the hardware to send a short packet so that host can receive the short packet and terminate IN transfer. Since EPOUT_RST is not used, data fetched before TXQ is stopped and may be sent to host after TXQ restarts.

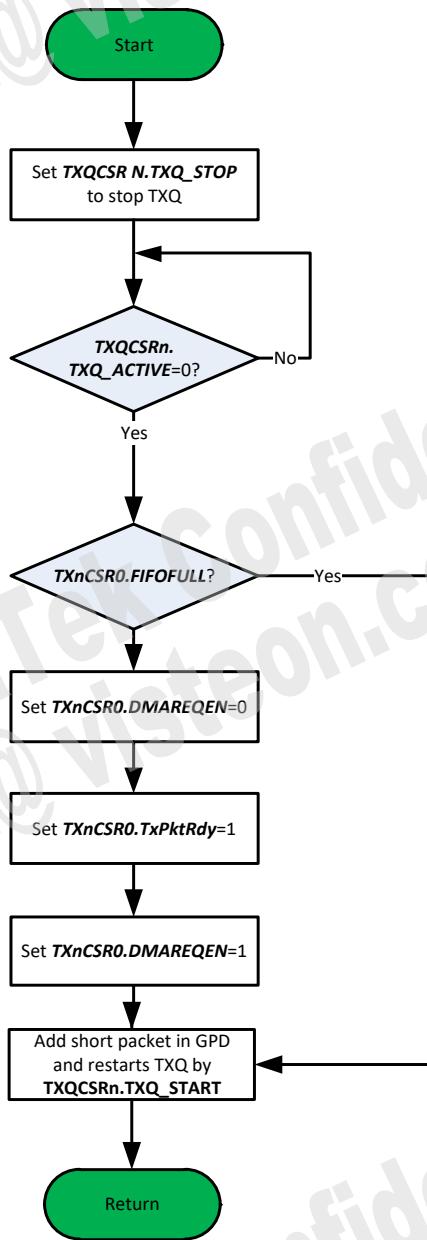


Figure 5-36 TXQ Short Packet (II) Flow

5.7.7.6.6 Error Handling Flow

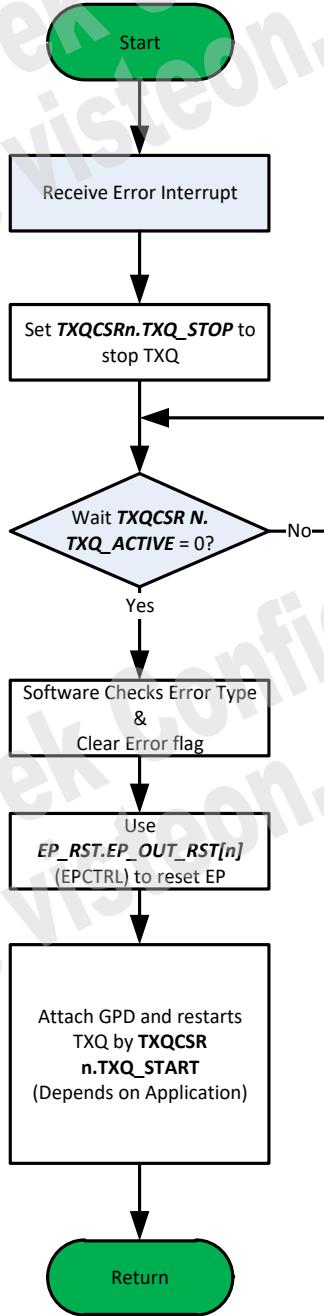


Figure 5-37 TXQ Error Handling Flow

5.7.7.6.7 GPD Execution Flow

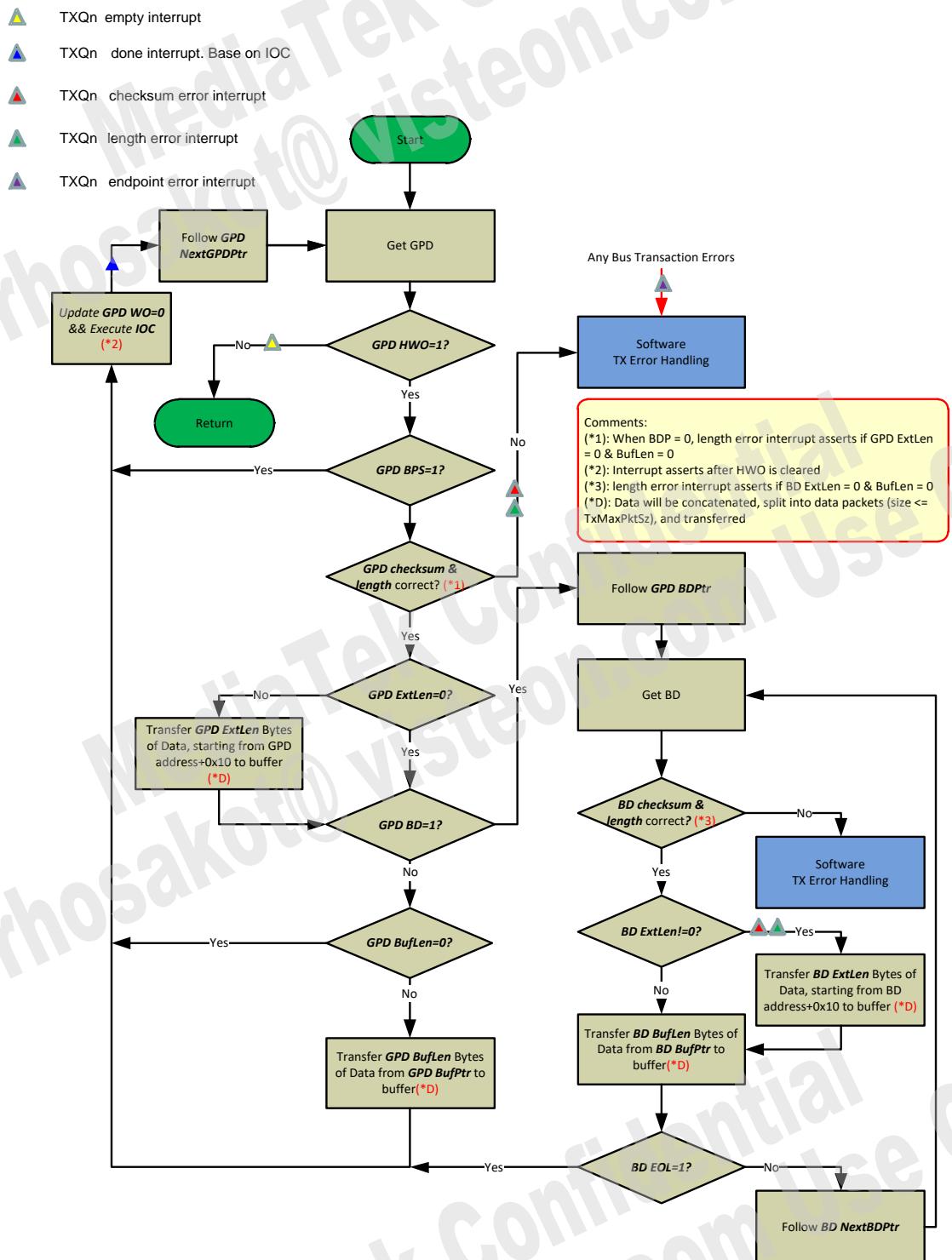


Figure 5-38 TXQ GPD Execution Flow

Note: If "bus transaction error" occurs, TxQn Endpoint error interrupt will be asserted. The bus transaction error will not occur in normal case.

5.7.7.7 RX Programming Flow

An overview of RXQ programming flow is shown in Figure 5-39. For details about blue blocks, please refer to the following subsections.

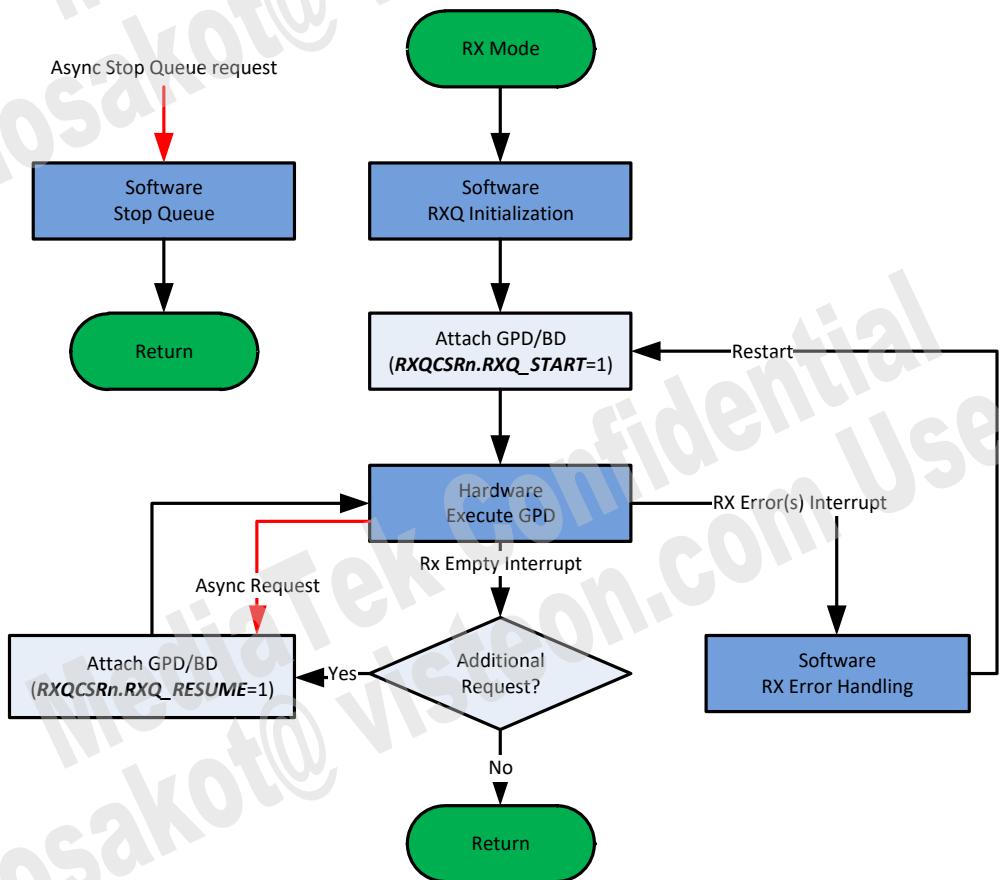


Figure 5-39 RX Programming Flow

5.7.7.7.1 Initialization Flow

Figure 5-40 shows which registers should be set before RXQ is used. Please refer to "MT2712 Registers" for register setting details.

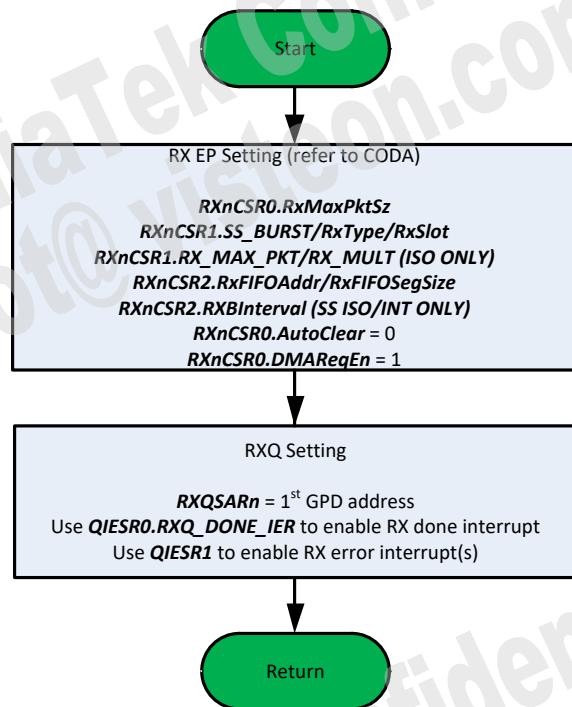


Figure 5-40 RX Initialization Flow

5.7.7.7.2 Stop Queue Flow

When RXQ is operating, some cases may occur; as a result, software has to stop the current transfer. There are two ways to notify the host. One is to STALL EP to notify the host explicitly; the other is to drop packets without notifying the host. The host has to do error handling in the first case. RXQ restart procedure at device side is application-specific.

5.7.7.7.3 STALL (Recommended)

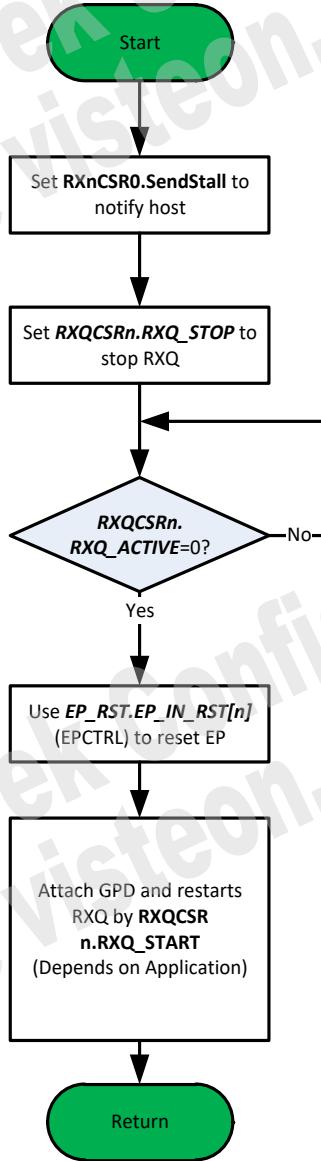


Figure 5-41 RX STALL Flow

5.7.7.4 Drop Packet

Device may not receive data correctly as it stops and restarts RXQ when host is in OUT transfer. Some packets are dropped during OUT transfer, and that results in the corruption of received data. The device has to do error handling under this condition.

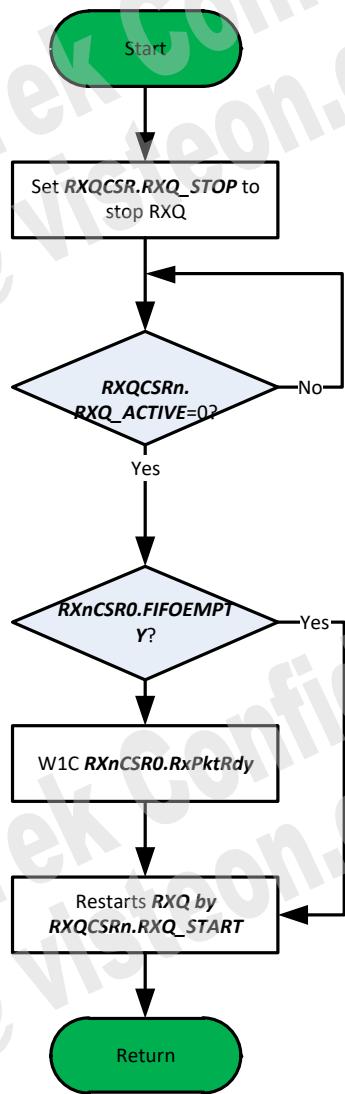


Figure 5-42 RX Drop Packet Flow

5.7.7.7.5 Error Handling Flow

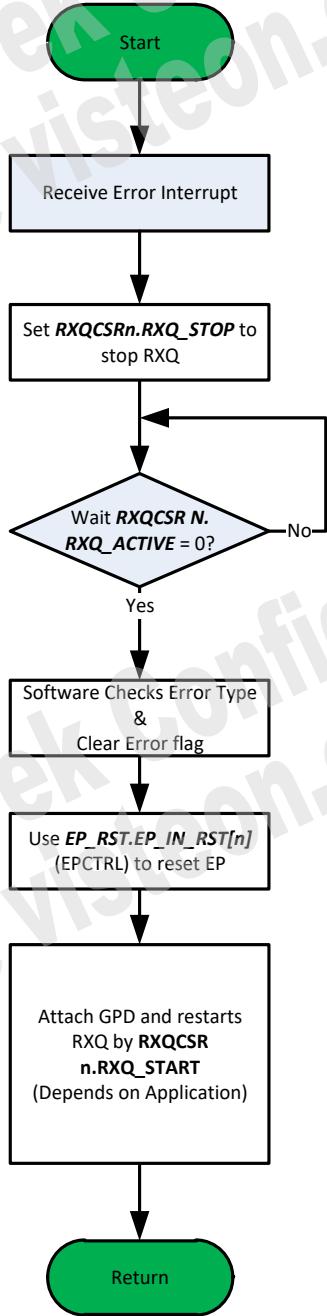


Figure 5-43 RX Error Handling Flow

5.7.7.7.6 GPD Execution Flow

- ▲ TXQn empty interrupt
- ▲ TXQn done interrupt. Base on IOC
- ▲ TXQn checksum error interrupt
- ▲ TXQn length error interrupt
- ▲ TXQn endpoint error interrupt

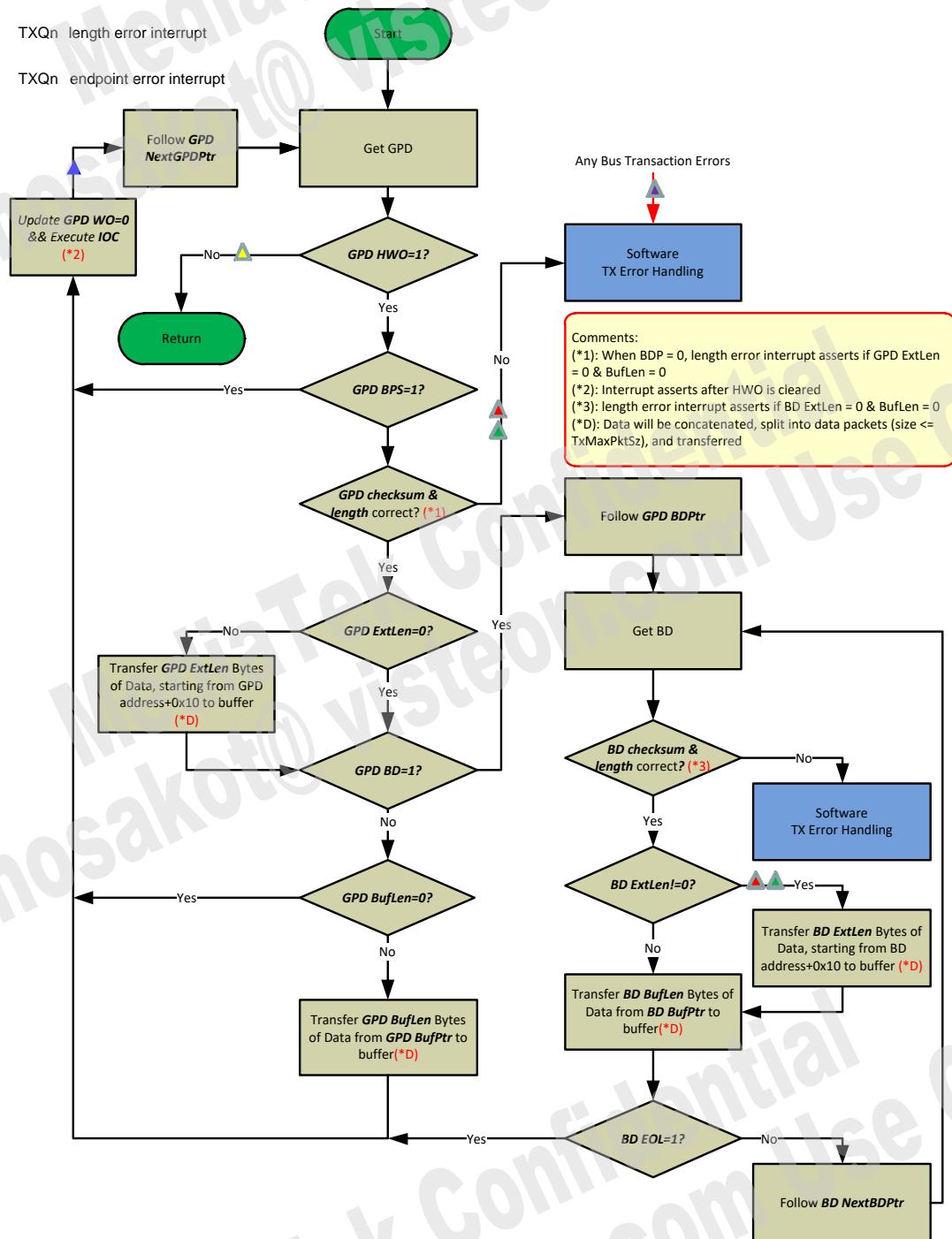


Figure 5-44 RX GPD Execution Flow

Note: If "bus transaction error" occurs, RxQn Endpoint error interrupt will be set by hardware. The bus transaction error shall not occur in normal case.

5.7.8 Reset

This section provides programming note on reset.

5.7.8.1 Address Reset

When the following condition occurs, device address will be reset to 0.

- USB 2.0
 - Upon receiving bus reset

5.7.8.2 Endpoint Reset

Endpoints will be reset by hardware when the following conditions occur:

- USB 2.0
 - Conditions
 - Upon receiving bus reset
 - Reset
 - Data toggle
 - FIFO pointer (FIFO address register is not touched)
 - EP flow control status

Since the following commands may affect endpoint application and hence its configuration:

- SET_CONFIGURATION
- SET_INTERFACE
- CLEAR_FEATURE ENDPOINT_HALT

Software can do the following operations before endpoint re-configuration:

1. Use bits in EP_RST to reset endpoints
2. Stop the corresponding Queue(s)

5.8 PCI Express

5.8.1 Introduction

MT2712 provides two PCI Express (PCIe) ports, and both are compliant to PCIe 2.0 specification. Each port is suitable for both Root Complex (RC) and End Point (EP). The PCIe MAC includes two major functional blocks: pe2_mac_core and pe2_mac_bridge.

5.8.2 Features

The features of PCIe module are:

- PCI Express Base Specification Revision 2.0 compliant
- Support link rate of PCIe Gen1 and Gen2
- Suitable for PCIe RC and EP
- Advanced Error Reporting (AER) support
- ECRC generation, check, and forward support
- Function Level Reset (FLR) support
- Support Express Card v1.1 and CLKREQ# functionality
- Support 64-bit BAR0 and 64-bit BAR1
- Support PCI-PM and ASPM L0s/L1
- Support PME message
- Support MSI and INT message

5.8.3 Block Diagram

As shown in Figure 5-45, PCIe consist of PCIe MAC and PCIe PHY. PCIe MAC is in the infra partition and PCIe PHY is in the top partition. They are connected with PIPE 3.0 interface.

As shown in Figure 5-46, PCIe MAC includes two major functional blocks: pe2_mac_core and pe2_mac_bridge.

- pe2_mac_core provides standard-defined PCIe MAC feature, and this is being partitioned as transaction layer/data link layer/LTSSM/phymac interface. The LTSSM and phymac interface operates at pcie_pipe_ck which is 125 MHz provided by PCIe PHY. The transaction and data link layer operates at pcie_mac_ck which is assumed to be provided by the top clock generator.
- pe2_mac_bridge is treated as an usage interface between application layer and pe2_mac_core. It provides standard AXI bus master interface for MMIO and DMA transaction, and AHB slave interface for CSR access.

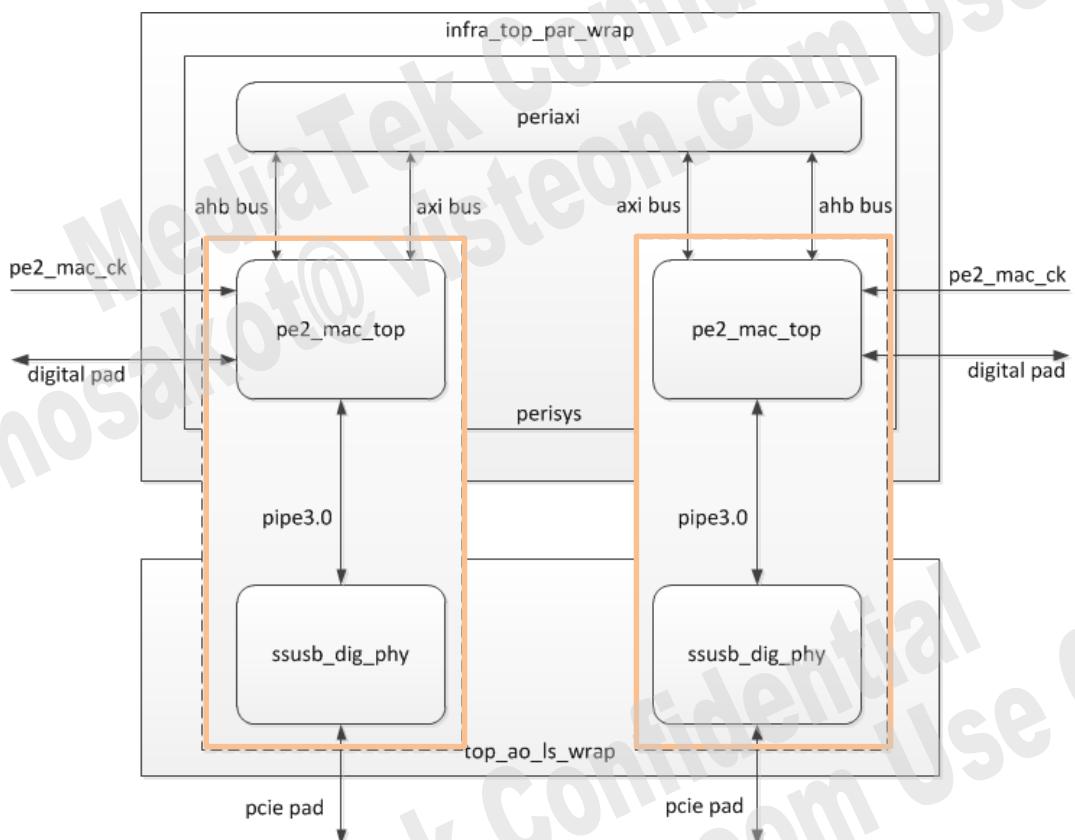


Figure 5-45 MT2712 PCIe Block Diagram

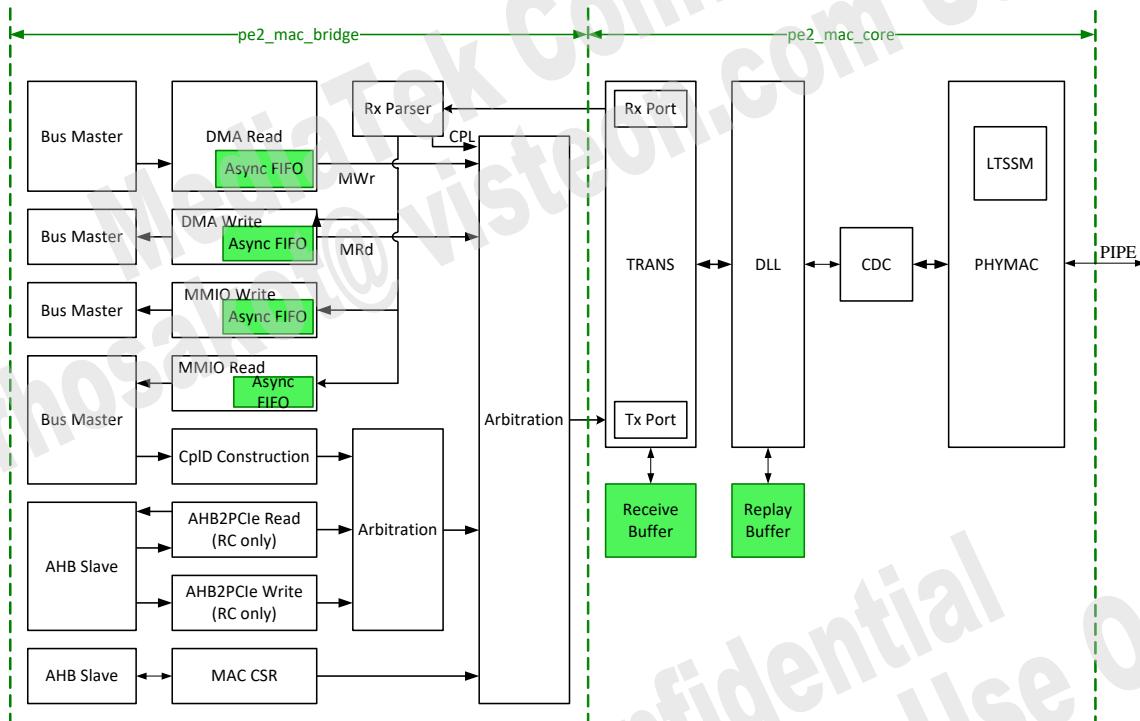


Figure 5-46 PCIe MAC Block Diagram

5.8.4 Clock Scheme

Figure 5-47 shows the clock domain partition of the PCIe MAC.

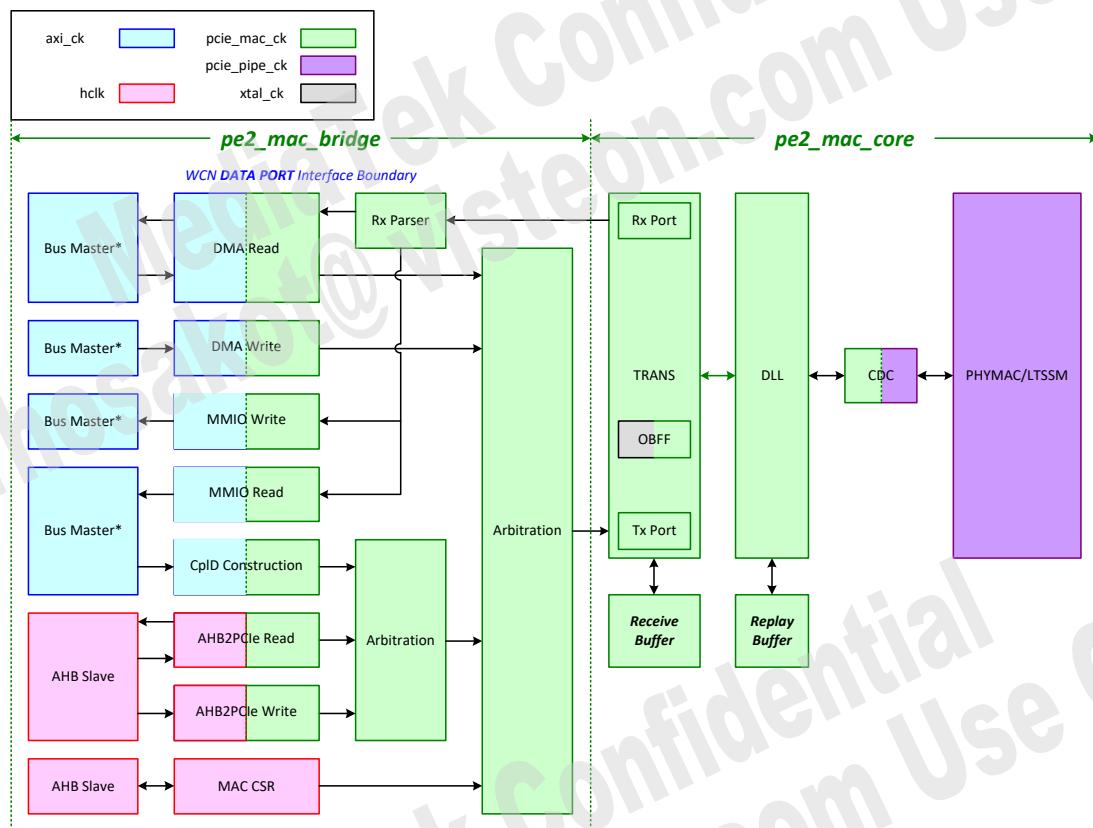


Figure 5-47 PCIe MAC Clock Domain Partition Illustration

5.8.5 Register Definition

For register details, please refer to Chapter 3.6 of “MT2712 IBI Application Processor Registers”.

5.8.6 Programming Guide

5.8.6.1 Power-On/Off Sequence

Please refer to INFRA power-on/off sequence.

5.8.6.2 HW Interrupt

PCIe Interrupt is shown in Table 5-43, and each port has one hardware interrupt pin.

Table 5-43 PCIe Interrupt

PORT	GIC Interrupt ID	Interrupt Name
port0	147	pe2_mac_irq_p0
port1	149	pe2_mac_irq_p1

5.8.6.3 PCIe Controller Initial Sequence

Step	Port Type	Address	Register Name	Local Address	R/W	Value	Description
PCIe Host Controller Initialization							
1	RC/EP	PCIe_MAC Base address +0x0000	port_type	K_GBL_1	W	appropriate value	Port Type must be chosen as Rootport or Endpoint. Please set bit [15:12] of K_GBL_1 register (PCIe_MAC BASE ADDRESS +0x0000), if the default value is not as expected.
Reset Setting							
2	RC/EP	PCIe_MAC Base address +0x0510	ResetPHY ResetMAC ResetBRG ResetCFG	PCI_RSTCR [0] PCI_RSTCR [1] PCI_RSTCR [2] PCI_RSTCR [3]	W	1'b0 1'b0 1'b0 1'b0	Step 2 ~ Step3: Software reset PCI_RSTCR [3:0](PCIe_MAC BASE ADDRESS +0x510) might be asserted and de-asserted for safety purposes.
3	RC/EP	PCIe_MAC Base address +0x0510	ResetMAC ResetPHY ResetBRG ResetCFG	PCI_RSTCR [0] PCI_RSTCR [1] PCI_RSTCR [2] PCI_RSTCR [3]	W	1'b1 1'b1 1'b1 1'b1	Step 4: And then PERST# (PCI_RSTCR [8])
4	RC	PCIe_MAC Base address +0x0510	ResetPE	PCI_RSTCR [8]	W	1'b0	(PCIe_MAC BASE ADDRESS +0x510)) must be de-asserted for the link sequence. PERST# asserted and de-asserted: It can be reached by writing 1'b0 to bit 8 of the PCI_RSTCR (PCIe_MAC BASE ADDRESS +0x510) PERST# de-asserts at least 100ms (TPVPERL) after the clock and power are stable. Then write 1'b1 to bit 8 of the PCI_RSTCR (PCIe_MAC BASE ADDRESS +0x510) to complete PERST# sequence.
Misc. Setting							
5	RC/EP	PCIe_MAC Base address +0x0100	DeviceID VendorID	K_CONF_FUNC0_0 [31:16] K_CONF_FUNC0_0 [15:0]	W	appropriate value	The Vendor ID, Device ID, Classcode and Revision ID can be set if the default value is not as expected. Step 5~6:
6	RC/EP	PCIe_MAC Base address +0x0104	Classcode RevisionID	K_CONF_FUNC0_1 [31:8] K_CONF_FUNC0_1 [7:0]	W	appropriate value	User has the ability to fill the corresponding value to the K_CONF_FUNC0_0 register (PCIe_MAC Base+0x0100) and K_CONF_FUNC0_1 register (PCIe_MAC +0x0104). Note: These bits are tied with eFuse values in MT2712. So the registers mentioned above can only be read.
INT Enable Setting							
8	RC/EP	PCIe_MAC Base address +0x0420	-	INT_MASK [31:0]	W	appropriate value	Enable the corresponding interrupt by setting INT_MASK register (PCIe_MAC Base address+0x0420)
Address Translation Setting							

Step	Port Type	Address	Register Name	Local Address	R/W	Value	Description
9	RC/EP	PCIe_MAC Base address +0x0400~0x040C	-	-	W	appropriate value	Set PCIe Address Translation window for I/O memory, prefetchable and non-prefetchable memory resource. Set the address transaction window to remap PCIe bus address if there is a demand from user. PCIe RC will translate the address from parent bus address to child bus address. The corresponding registers group starts from (PCIe_MAC Base address+0x0400) to (PCIe_MAC Base address+0x044C).
Configuration Space Setting							
10	RC/EP	-	MEM_ENABLE Bus_master_enable	Command Register (Offset 04h) [1] Command Register (Offset 04h) [2]	W	1'b1 1'b1	Bus Master Enable and MEM_ENABLE should be asserted by writing 2'b11 to bit [2:1] of the Command Register (Offset 04h) in configuration space common header.
11	EP	-	-	Base Address Registers (Offset 10h - 24h)	W	appropriate value	Base Address Registers (Offset 10h - 24h) should also be enabled according to user's application.
12	RC	-	Prim_Bus_Num Sec_Bus_Num Sub_Bus_Num	Primary /Secondary/ Subordinate Bus Number (Offset 18h)	W	appropriate value	It is recommended to give an effective Subordinate Bus Number, Secondary Bus Number, and Primary Bus Number by writing appropriate values to bit 23 to 16, bit 15 to 8, and bit 7 to 0 of the configuration space type1 header (Offset 18h). It will be filled up by native driver during Linux boot up.
13	RC	-	IO_Base IO_Limit	I/O Limit I/O Base(Offset 1Ch)	W	appropriate value	I/O Base and I/O Limit should be specified, according to the downstream device I/O Range, to bit 15 to 12 and bit 7 to 4 of the configuration space type1 header (Offset 1Ch). It will be filled up by a native driver during Linux boot up.
14	RC	-	Mem_Base Mem_Limit	Memory Limit Memory Base(Offset 20h)	W	appropriate value	Memory Limit and Memory Base should be specified, according to the downstream device Memory Range, to bit 31 to 20 and bit 15 to 4 of the configuration space type1 header (Offset 20h). It will be filled up by a native driver during Linux boot up.

5.8.6.4 Memory Transactions

Memory transactions are memory space read and write requests (MRd and MWr TLPs), which are generated by PCIe device and sent over the PCI Express link. The respective completion TLPs are received in return.

The following features are supported by a master Memory requester:

- Maximum Memory read request of 128 Bytes
- Maximum Memory write request of 128 Bytes
- 32-bit or 64-bit addressing

For initiating Memory read/write requests, Memory Space Enable should be asserted by writing 2'b11 to bit [2:1] of the Command Register (Offset 04h) in configuration space common header. Then user can initiate memory access based on Memory-mapped address. The related TLP should be passed to target PCIe component based on Address-routing rule.

5.8.6.5 Configuration Transactions

Configuration transactions are Configuration space read and write requests (CfgRd0, CfgWr0, CfgRd1 and CfgWr1 TLPs), which are generated by PCIe device and sent over the PCI Express link. The respective completion TLPs are received in return.

The following features are supported by a master Configuration requester:

- Maximum Configuration read request of 4 Bytes
- Maximum Configuration write request of 4 Bytes
- Extended register number support (4 KB extended PCI Express configuration header space)

For initiating Configuration read or write request to target endpoint, it should specify the Bus number, Device number and Function number of corresponding Endpoint to the configuration space type1 header (Offset 18h), and user can trigger the desired Configuration request by programming from (PCIe_MAC Base address+0x0460) to (PCIe_MAC Base address+0x0470), and then set bit[0] of the (PCIe_MAC Base address+0x0488). When Cfg TLP is triggered by application layer, this bit will keep asserting until a Cpl/CpID is received. The related TLP should be passed to the target PCIe component based on ID-routing rule.

5.8.6.6 Message Transactions

Message transactions use Message space to support all prior sideband signals, such as interrupts and power-management requests. They are used to support in-band communication of events between devices.

The following groups of Messages are not supported:

- Locked Transaction support
- OBFF(Optimized Buffer Flush/Fill) Message

The following groups of Messages are supported:

- INTx Interrupt Signaling
- Power Management
- Error Signaling
- Slot Power Limit Support
- Vendor-Defined Messages
- LTR Message

5.8.6.7 Error Handling Function

PCIe supports hardware-controlled autonomous error reporting and handling, to improve fault isolation and recovery solution. The following description defines these functions.

- Error Recovery: The Data Link layer handles error recovery. It is based on 32-bit CRC (Cyclic Redundant Check) error detection, TLP sequence number, Replay buffer, and ACK/NAK DLLP exchange.
- Replay Buffer: The Replay buffer is located in the Data Link Layer and stores a copy of a transmitted TLP until the transmitted packet is acknowledged by the receiving side of the Link.
- Completion Timeout: PCIe controller uses a timeout mechanism (which is design-specific) for failed Read Request transaction in order to report Error Messages and to free Completion resources.

5.8.6.8 Extended Description

PCIe Memory map is shown in Table 5-44 , and MMIO space is different for PCIe port0 and port1.

0x2000_0000~0x2FFF_FFFF is the 256 MB MMIO space for PCIe port0.

0x1130_0000~0x116F_FFFF is the 4 MB MMIO space for PCIe port1.

0x1170_0000~0x1170_0FFF is the 4 KB control register space for PCIe port0.

0x112F_F000~0x112F_FFFF is the 4 KB control register space for PCIe port1.

Table 5-44 PCIe MAC Memory Map

Start Address	End Address	Size	Description
0x112F_F000	0x112F_FFFF	4 KB	PCIe port1 control register space
0x1130_0000	0x116F_FFFF	4 MB	PCIe port1 MMIO space
0x1170_0000	0x1170_0FFF	4 KB	PCIe port0 control register space
0x2000_0000	0x2FFF_FFFF	256 MB	PCIe port0 MMIO space

5.9 SSUSB_PCIE_PHY

5.9.1 Introduction

The USB SuperSpeed and PCI Express Combo PHY (SSUSB_PCIE_PHY) implements the Physical Layer functions defined in USB3.0 and PCIe2.0 Base Specification, and can be configured as the USB SuperSpeed or PCI Express PHY mode.

The SSUSB_PCIE_PHY handles the low level protocol and signaling. This includes features such as data serialization and de-serialization, 8b/10b encoding/decoding, analog buffers, elastic buffers and receiver detection. The primary focus of this block is to shift the clock domain of data from the PCI Express rate or USB SuperSpeed rate to one that is compatible with the general logic.

5.9.2 Features

The USB SuperSpeed PHY key features are listed below:

- Standard PHY interface provides a target interface for USB SuperSpeed PHY vendors
- Supports 5.0 GT/s serial data transmission rate
- Utilizes 32-bit parallel interface to transmit and receive USB SuperSpeed data
- Allows integration of high-speed components into a single functional block
- Clock and Data Recovery (CDR) from serial stream on the USB SuperSpeed bus
- Holding registers to stage transmit and receive data
- Supports direct disparity control for use in transmitting compliance pattern (s)
- 8b/10b encode/decode and error indication
- Receiver detection
- Low Frequency Periodic Signaling (LFPS) Transmission
- Selectable TX Margining

The PCI Express PHY key features are listed below:

- Standard PHY interface provides a target interface for PCI Express PHY vendors
- Supports 2.5 GT/s and 5.0 GT/s serial data transmission rate
- Utilizes 16-bit and 32-bit parallel interface to transmit and receive PCI Express data
- Allows integration of high speed components into a single functional block
- Clock and Data Recovery (CDR) from serial stream on the PCI Express bus
- Holding registers to stage transmit and receive data
- Supports direct disparity control for use in transmitting compliance pattern (s)
- 8b/10b encode/decode and error indication
- Receiver detection
- Selectable TX Margining, TX De-emphasis and signal swing values

5.9.3 Block Diagram

Figure 5-48 shows the system-level block diagram of MT2712's SSUSB_PCIE_PHY. There are two main sub-modules in SSUSB_PCIE_PHY block: one is PHYA (Physical Layer Analog Block), and the other is PHYD (Physical Layer Digital Block).

PHYA includes TX driver which is used to output differential pair signals, RX Front-End which is used to receive differential pair data, CDR which is used to recover timing information from serial data stream, and TX Serializer and De-Serializer which are used to convert data between serial and parallel data interfaces in each direction. PHYD includes 8b/10b encoder and decoder (for 2.5 GT/s and 5.0 GT/s), and elastic buffers which are used to compensate for differences in frequencies between bit rates at the two ends of a Link.

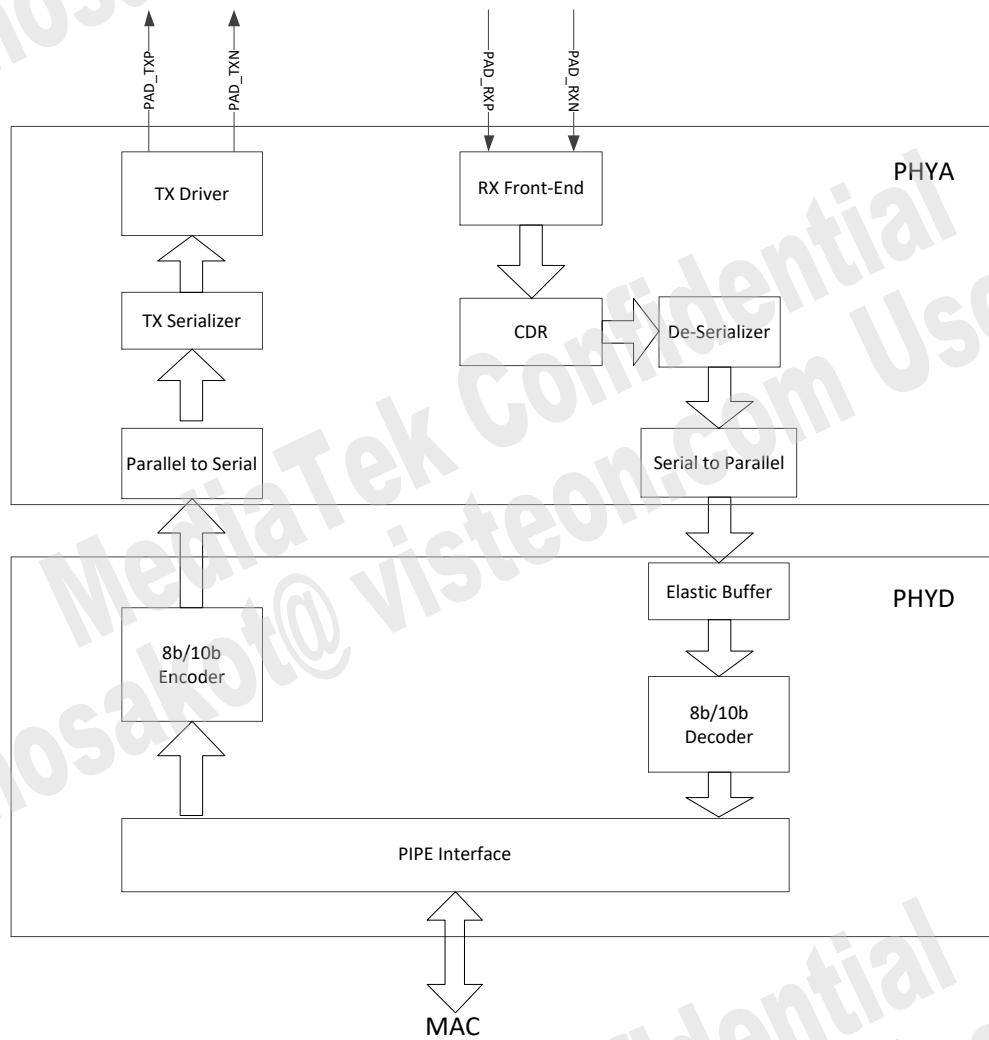


Figure 5-48 SSUSB_PCIE_PHY Block Diagram

5.9.4 USB 2.0 High Speed Controller AC Timing

5.9.4.1 USB 2.0 Introduction

USB, a short abbreviation for Universal Serial Bus, is an industry standard that is developed to define cables, connectors and protocols for connection, communication and power supply between personal computers and their peripheral devices. USB 2.0 adds a higher maximum signaling rate of 480 Mbit/s (High Speed), in addition to the USB 1.x Full Speed signaling rate of 12 Mbit/s.

5.9.4.2 USB 2.0 Block Diagram

Figure 5-49 is the block diagram of the external interrupt controller in MT2712.

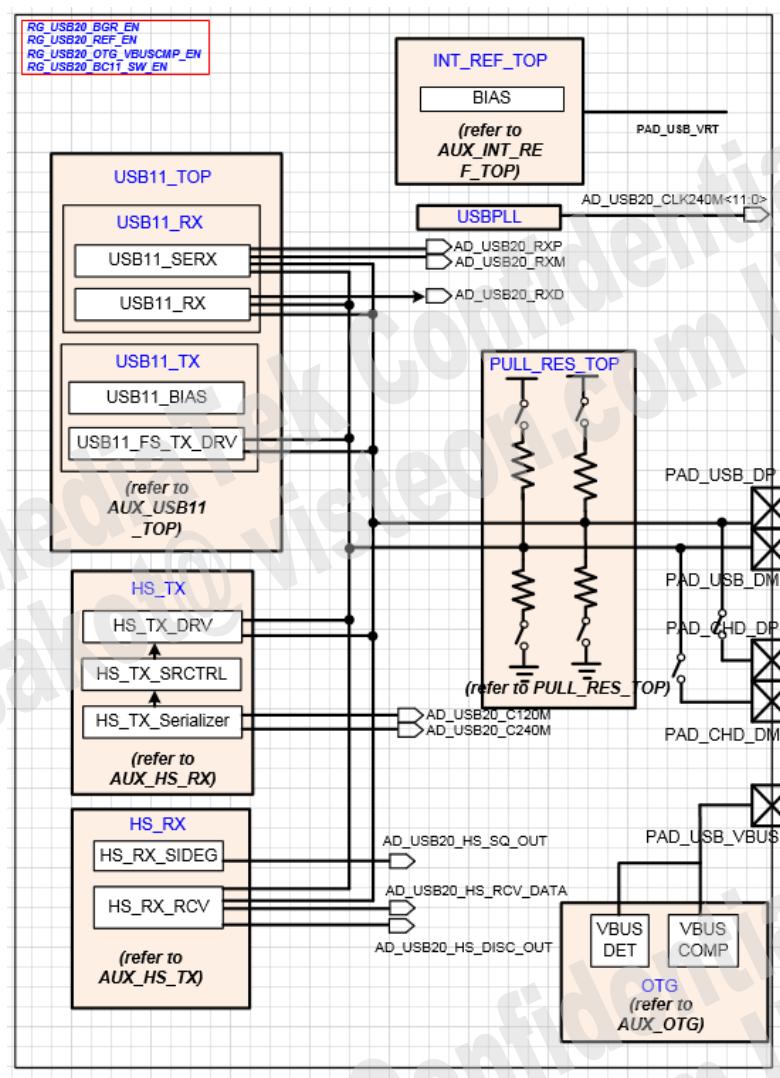


Figure 5-49 USB 2.0 Top Block Diagram

5.9.4.3 USB 2.0 Data Signal Rise and Fall Eye Patterns

The following sections specify the rise and fall times of the data signal for the low-speed and full-speed signaling with the rise time and eye patterns for high-speed signaling.

5.9.4.3.1 Low-speed and Full-speed Data with Signal Rise and Fall

For low-speed and full-speed, the output rise times and fall times are measured between 10% and 90% of the signal (Figure 5-50). Rise and fall time requirements apply the different transitions and transitions between differential and single-ended signaling.

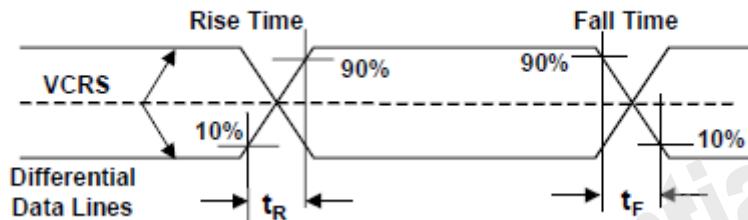


Figure 5-50 Data Signal Rise and Fall Time

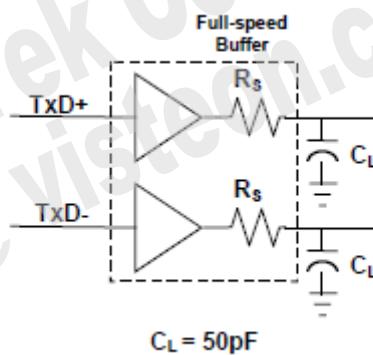
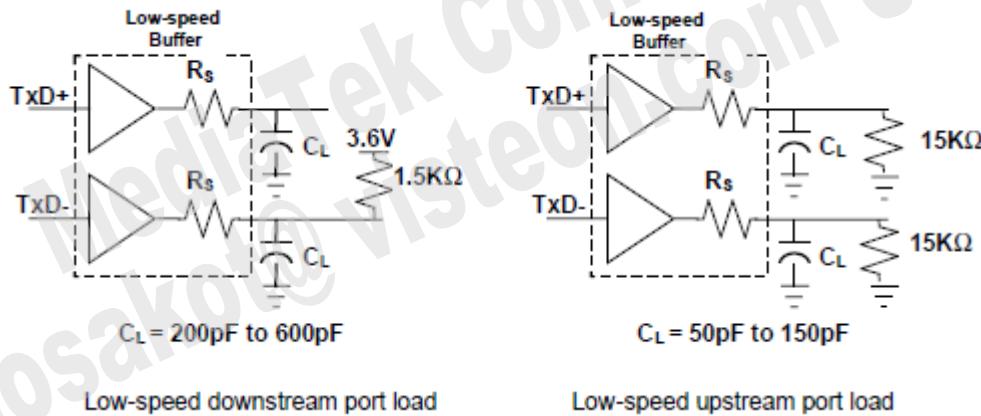


Figure 5-51 Full-Speed Load

The rise and fall times of full-speed buffers are measured by the load as shown in Figure 5-51. The rise and fall times must be between 4 ns~20 ns and matched within 10% to minimize the RFI emissions and signal skew. The transitions must be monotonic.

The rise and fall times of low-speed buffers are measured by the load as shown in Figure 5-52. The capacitive-load as shown in Figure 5-52 represents the worst-case load allowed by the specification.

**Figure 5-52 Low-Speed Port Loads****5.9.4.3.2 High-speed Signaling Eye Patterns and Rise and Fall Time**

The following specifications apply the high-speed mode signaling. All bits including the first and the last bit of a packet must meet the following requirements of eye patterns for the timing and amplitude.

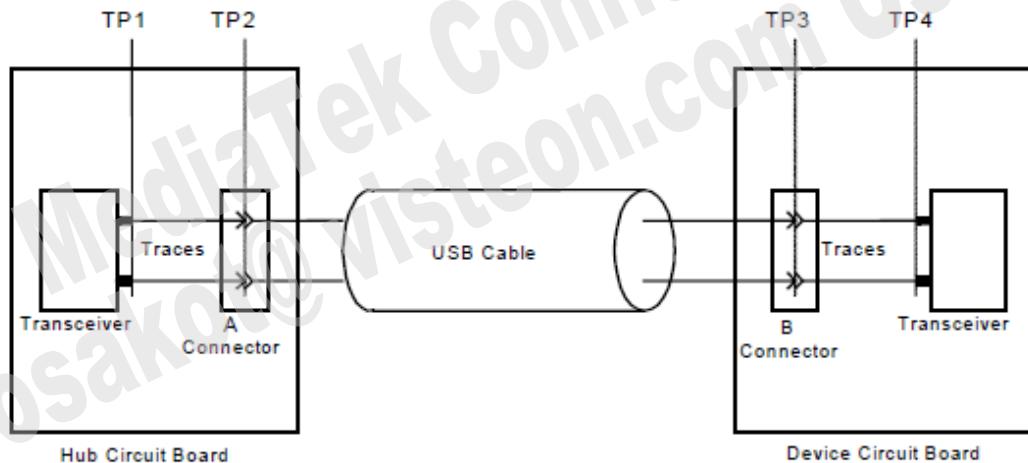
**Figure 5-53 Measurement Planes**

Figure 5-53 defines four test planes which will be referenced in this section. TP1 and TP4 are the points where the transceiver IC pins are soldered to the hub and device circuit boards respectively. TP2 is at the mated pins of the A connector and TP3 is at the mated pins of the B connector (or in the case of a captive cable where the cable is attached to the circuit board).

The following differential templates of eye patterns specify the transmit waveform and receiver sensitivity requirements at various points under various conditions. Transmit eye patterns specify the minimum and maximum limits as well as the limits on timing jitter. In it a driver must drive signals at each of the specified test

planes. Received eye patterns specify the minimum and the maximum limits as well as limits on timing jitter in which a receiver must recover data. Conformance to the templates 1, 2, 3 and 4 is required for the USB 2.0 hubs and devices:

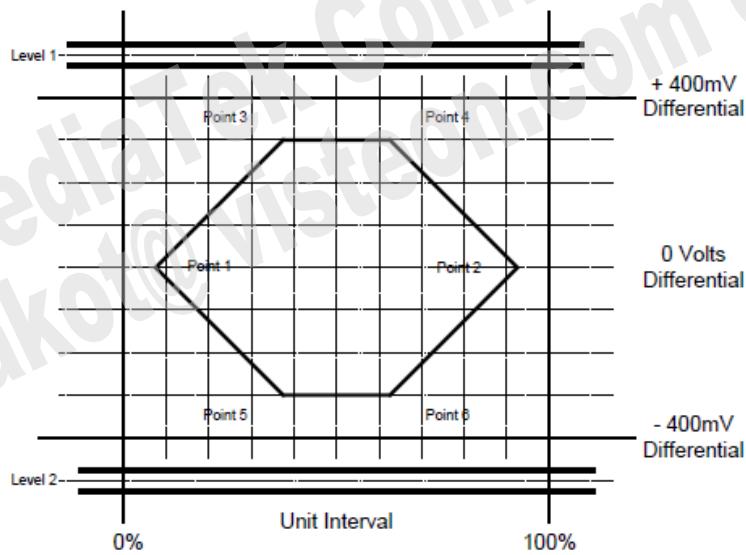
- Template 1:** Transmit waveform requirements for the hub measured at TP2 and for device (without a captive cable) measured at TP3.
- Template 2:** Transmit waveform requirements for device (with a captive cable) measured at TP2.
- Template 3:** Receiver sensitivity requirements for the device (with a captive cable) when signal is applied at TP2.
- Template 4:** Receiver sensitivity requirements for the device (without a captive cable) when signal is applied at TP3 and for hub when signal is applied at TP2.

Templates 5 and 6 constitute the recommended guidelines for designers:

- Template 5:** Transmit waveform requirements for the hub transceiver measured at TP1 and for the device transceiver measured at TP4.
- Template 6:** Receiver sensitivity requirements for the device transceiver when signal is applied at the TP4 and for the hub transceiver when signal is applied at TP1.

Template 1

Figure 5-54 shows the transmit waveform requirements for a hub measured at TP2 and for the device (without a captive cable) measured at TP3.

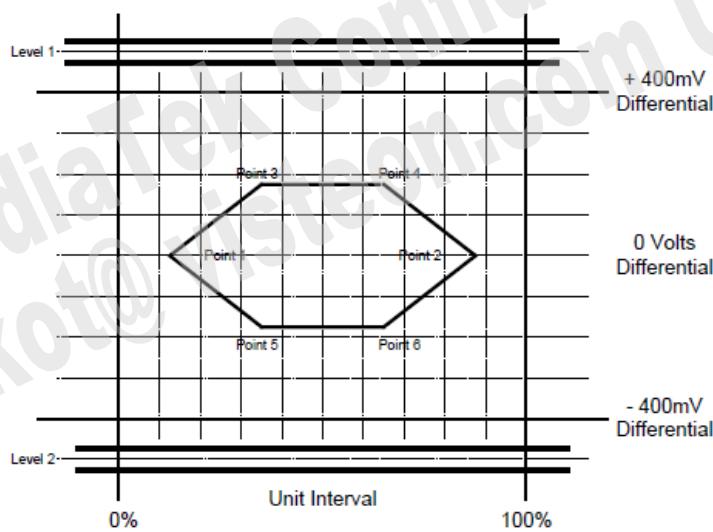


	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 mV in all others	N/A
Point 1	0 V	7.5% UI
Point 2	0 V	92.5% UI
Point 3	300 mV	37.5% UI
Point 4	300 mV	62.5% UI
Point 5	-300 mV	37.5% UI
Point 6	-300 mV	62.5% UI

Figure 5-54 Template 1

Template 2

Figure 5-55 shows transmit waveform requirements for the device (with a captive cable) measured at TP2.

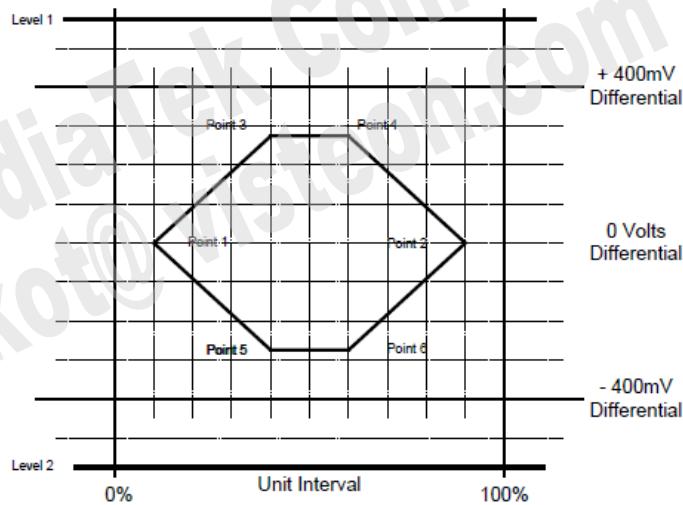


	Voltage Level ($D^+ - D^-$)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 mV in all others	N/A
Point 1	0 V	12.5% UI
Point 2	0 V	87.5% UI
Point 3	175 mV	35% UI
Point 4	175 mV	65% UI
Point 5	-175 mV	35% UI
Point 6	-175 mV	65% UI

Figure 5-55 Template 2

Template 3

Figure 5-56 shows receiver sensitivity requirements for the device (with a captive cable) when a signal is applied at TP2.



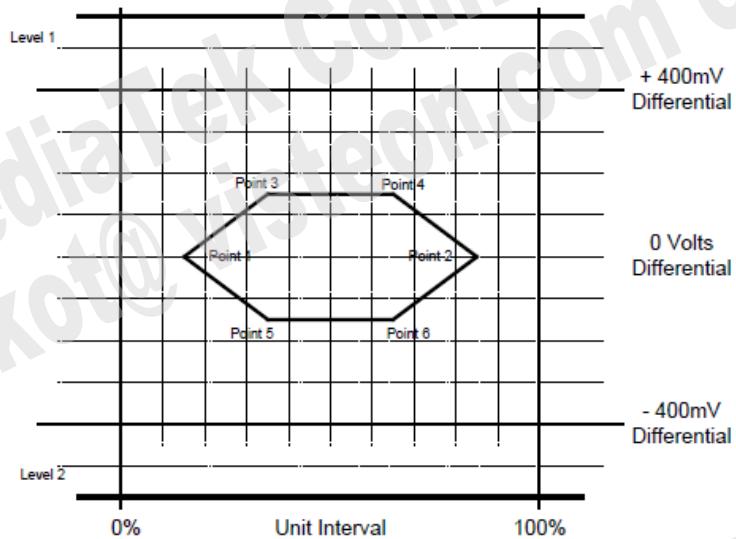
	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	10% UI
Point 2	0 V	90% UI
Point 3	275 mV	40% UI
Point 4	275 mV	60% UI
Point 5	-275 mV	40% UI
Point 6	-275 mV	60% UI

Figure 5-56 Template 3

Note: This eye is intended to specify the differential requirements of data receiver sensitivity. Level 1 and 2 are outside the Disconnect Threshold values but the disconnection is detected at the source (after a minimum of 32-bit times without any transitions) not at the target receiver.

Template 4

Figure 5-57 shows the receiver sensitivity requirements for a device (without a captive cable) when a signal is applied at the TP3 and for a hub when a signal is applied at TP2.



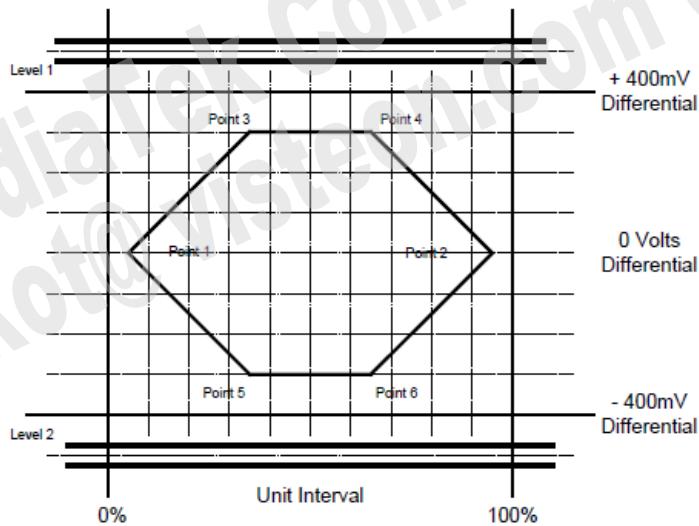
	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	15% UI
Point 2	0 V	85% UI
Point 3	150 mV	35% UI
Point 4	150 mV	65% UI
Point 5	-150 mV	35% UI
Point 6	-150 mV	65% UI

Figure 5-57 Template 4

Note: This eye is intended to specify the differential data receiver sensitivity requirements. Level 1 and 2 are outside the Disconnect Threshold values but disconnection is detected at the source (after a minimum of 32-bit times without any transitions) not at the target receiver.

Template 5

Figure 5-58 shows transmitted waveform requirements for a hub transceiver measured at TP1 and for a device transceiver measured at TP4.

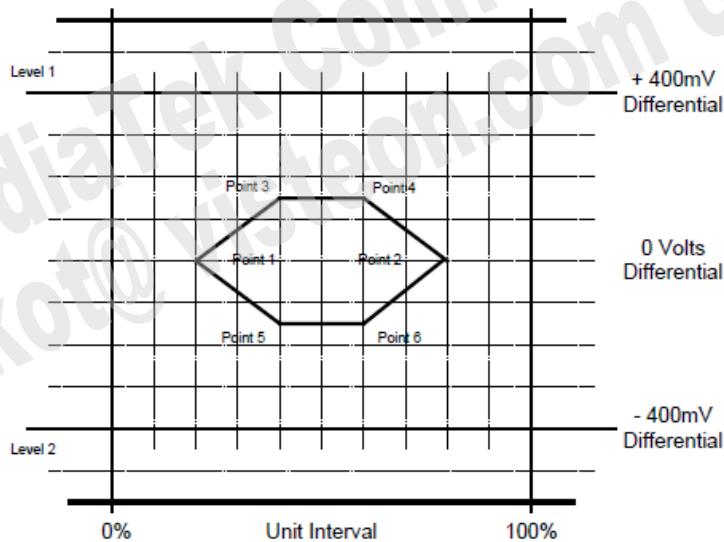


	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 mV in all others	N/A
Point 1	0 V	5% UI
Point 2	0 V	95% UI
Point 3	300 mV	35% UI
Point 4	300 mV	65% UI
Point 5	-300 mV	35% UI
Point 6	-300 mV	65% UI

Figure 5-58 Template 5

Template 6

Figure 5-59 shows receiver sensitivity requirements for a device transceiver when a signal is applied at TP4 and for a hub transceiver when a signal is applied at TP1.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	20% UI
Point 2	0 V	80% UI
Point 3	150 mV	40% UI
Point 4	150 mV	60% UI
Point 5	-150 mV	40% UI
Point 6	-150 mV	60% UI

Figure 5-59 Template 6

Note: This eye is intended to specify the differential data receiver sensitivity requirements. Level 1 and 2 are outside the Disconnect Threshold values but disconnection is detected at the source (after a minimum of 32-bit times without any transitions), not at the target receiver.

5.9.4.3.3 High-speed Signaling Rise and Fall Times

The transition time of a high-speed driver must be no less than the specified minimum allowable differential rise and fall times (THSR and THSF).

Transition times are measured when driving a reference load of 45 Ohm to the ground on D+ and D-.

Figure 5-60 shows a recommended “Transmitter Test Fixture” for performing these measurements.

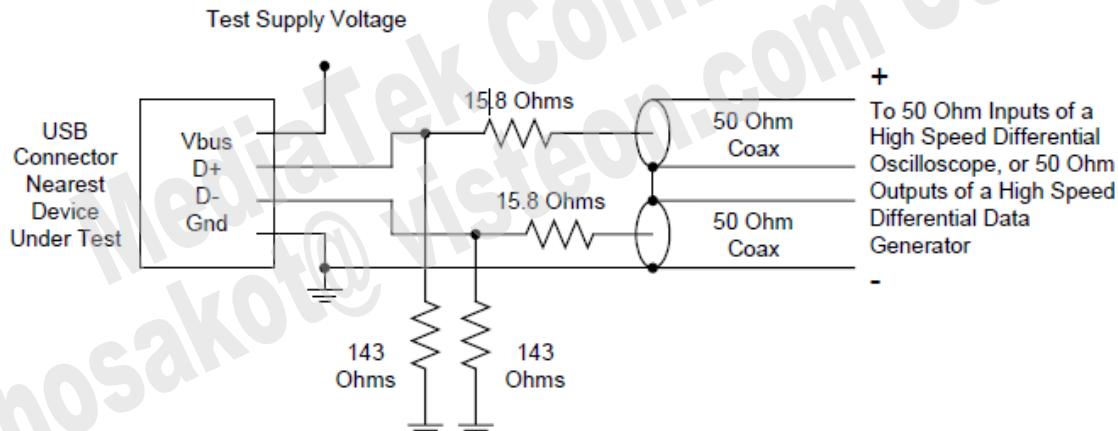


Figure 5-60 Transmitter/Receiver Test Fixture

For a hub or for a device with the detachable cable, the 10- 90 % high-speed differential rise and fall times must be between 500-ps or longer, when measured at the A or B receptacles respectively. For a device with a captive cable assembly, it is a recommended design guideline that the 10% to 90% high speed differential rise and fall times must be 500-ps or longer when measured at the point where the cable is attached to the device circuit board. It is required that high-speed data transitions should be monotonic over the minimum vertical openings specified in the preceding eye pattern templates.

5.9.5 USB 3.0 High Speed Controller AC Timing

5.9.5.1 SSUSB PHY Parameters

The SSUSB Interface complies with the USB 3.0 specification. It can be referred to as USB 3.0 Compliance Test Report.

5.9.5.2 SSUSB_REXT Reference Resistor Connection

The impedance calibration process requires connection to a reference resistor 5.1k Ohm. 1% precision resistor on PAD_VRT pad to ground.

5.9.6 PCIe PHY AC Timing

5.9.6.1 PCIe PHY Parameters

The PCIe Interface complies with PCI Express 1.1/2.0 specification. It can be referred to as PCIe Compliance Test Report.

5.9.6.1.1 PCIe_REXT Reference Resistor Connection

The impedance calibration process requires a reference resistor. It can be implemented either by connecting a 5.1 Kohm_1% precision resistor on PAD_VRT pad to ground or using built-in resistor, which is the default option with the PAD_VRT floating.

5.9.7 Clock Relationship

Figure 5-61 shows the Clocking block diagram of MT2712 SSUSB_PCIE_PHY.

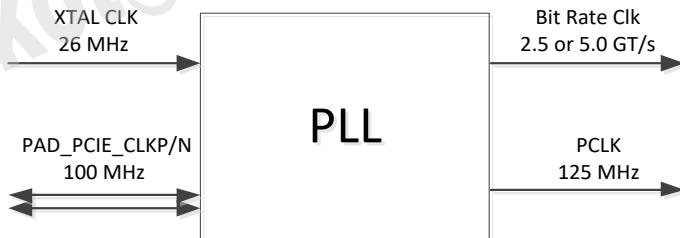


Figure 5-61 SSUSB_PCIE_PHY Clocking Block Diagram

5.9.8 Register Definition

For register details, please refer to Chapter 3.7 of “MT2712 IVI Application Processor Registers”.

5.9.9 Programming Guide

5.9.9.1 Power-On/Off Sequence

Please refer to the USB SuperSpeed and PCI Express Media Access Layer (MAC) power-on/off sequence.

5.10 SPI Interface

5.10.1 Introduction

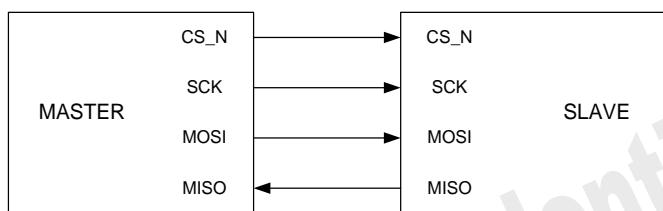


Figure 5-62 Pin Connection between SPI Master and SPI Slave

The Serial Peripheral Interface (SPI) interface is a bit-serial, four-pin transmission protocol. Figure 5-62 is an example of the connection between the SPI master and SPI slave. MT2712 includes five SPI masters (spi0/2/3/4/5) and one SPI slave (spi1).

5.10.2 Features

SPI Master Features:

- Configurable CS_N setup time, hold time and idle time
- Programmable Serial Clock (SCK) high time and low time
- Configurable transmitting and receiving bit order
- Two configurable modes for the source of the data to be transmitted
- In TX DMA mode, the SPI controller automatically fetches the transmitted data (to be put on the MOSI line) from memory.
- In TX FIFO mode, the data to be transmitted on the MOSI line are written to FIFO before the start of the transaction.
- Two configurable modes for destination of the data to be received.
- In RX DMA mode, the SPI controller automatically stores the received data (from MISO line) to memory;
- In RX FIFO mode, the received data stay in RX FIFO of the SPI controller temporarily. The processor must read back the data by itself.
- Adjustable endian order from/to memory system
- Programmable byte length for transmission
- Unlimited length for transmission. This is achieved by the operation of PAUSE mode. In PAUSE mode, the CS_N signal will stay active (low) after the transmission. At this time, the SPI controller is in PAUSE_IDLE state, ready to receive the resume command. The state transition is shown in Figure 5-63.
- Configurable option to control CS_N de-assert between byte transfers. The controller supports a special transmission format called CS_N de-assert mode. Figure 5-64 illustrates the waveform in this transmission format.

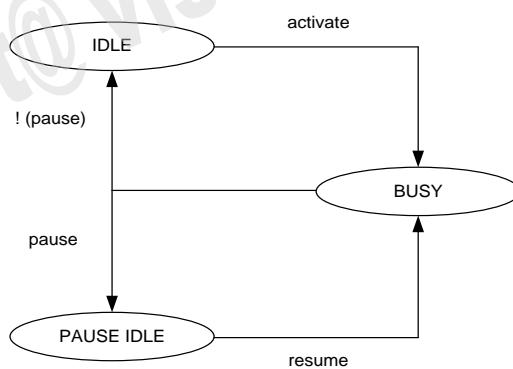


Figure 5-63 Operation Flow with or without PAUSE Mode



Figure 5-64 CS_N De-assert Mode

SPI Slave Features:

- The supported SPI_CLK is up to 26 MHz. (Only when SPI master can adjust sample clock delay, the maximum value of SPI slave's clock can be 50 MHz)
- Configurable transmitting and receiving bit order
- Four communication modes are available (MODE 0, 1, 2, 3), and each mode is formally defined with a pair of parameters, called 'clock polarity' (CPOL) and 'clock phase' (CPHA). When SPI slave is configured as CPOL=0 and CPHA =0 (MODE0) or CPOL=1 and CPHA=1 (MODE3), it samples the MOSI line on the rising edge of SCK. When SPI slave is configured as CPOL=0 and CPHA =1 (MODE1) or CPOL=1 and CPHA=0 (MODE2), it samples the MOSI line on the falling edge of SCK. There is a common rule which describes when SPI slave toggles the MISO line under the four communication modes: SPI slave puts the 1st data bit on the MISO line eight module clock cycles after detecting a negative edge of the SPI_CS signal; SPI slave shifts subsequent data bit four module clock cycles after the sampling edge (defined by CPOL and CPHA); SPI slave needs two extra module clock cycles to load next word at every 32-bit boundary, so it puts 33rd, 65th , 97th ... data bit on the MISO line six module clock cycles after the sampling edge. Figure 5-65 shows the timing sequence of the SPI_SLAVE_MISO signal when CPOL=0 and CPHA=0. SPI slave follows the same rule under the other three modes.

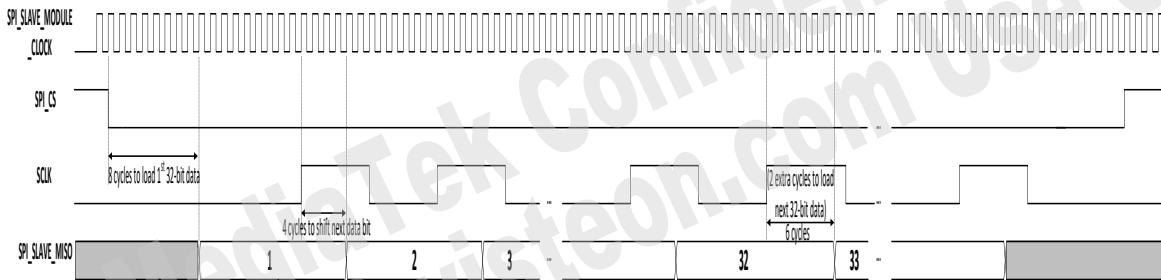


Figure 5-65 Timing Sequence of SPI_SLAVE_MISO Line When CPOL=0 and CPHA=0

- Enable/disable Transmit and Receive mode
- Default TX FIFO data (default is 0x00)
- If there is no data in TX FIFO, but SPI master wants to get data from SPI slave, then SPI slave will output configurable constant byte value on SPI bus, and the default value is 0x00.
- RX/TX FIFO data status
- There will be RX/TX FIFO pointer, Number of Bytes transmitted/received in status register. These can be read for status check.
- Interrupt support
- There will be RX full interrupt and Transfer done interrupt for indication.
- Support PIO mode and DMA mode transfer
- Both DMA/PIO mode are supported on SPI slave TX/RX channel.
- Programmable Byte length for transmission
- TX DMA length can be between 1 Byte and 1M Bytes.
- TX/RX's each FIFO depth: 128 x 4 Bytes.

5.10.3 Transmission Formats

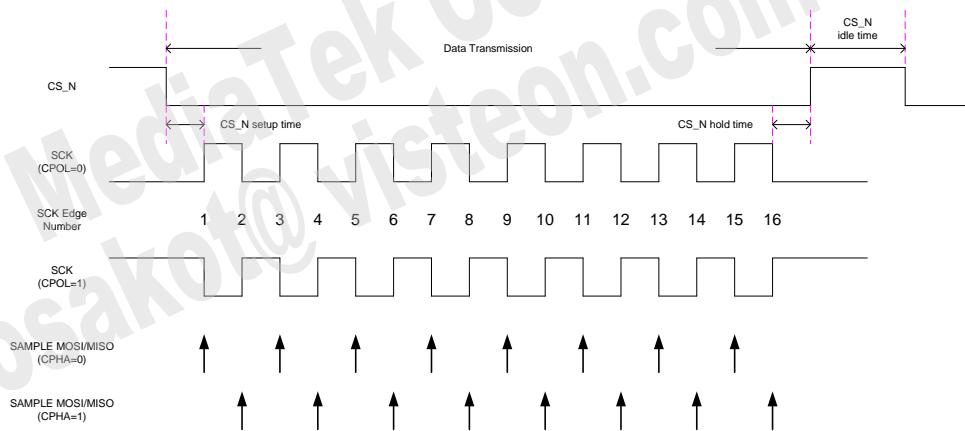


Figure 5-66 SPI Transmission Formats

Figure 5-66 shows the waveform during the SPI transmission. The low active CS_N determines the start point and end point of one transaction. The CS_N setup time, hold time and idle time are also depicted. CPOL defines the clock polarity in the transmission. Two types of polarity can be adopted, i.e. polarity 0 and polarity 1. Figure 5-66 shows both of the CPOL as examples. CPHA defines the legal timing to sample MOSI and MISO. Two different methods can be adopted.

5.10.4 Block Diagram

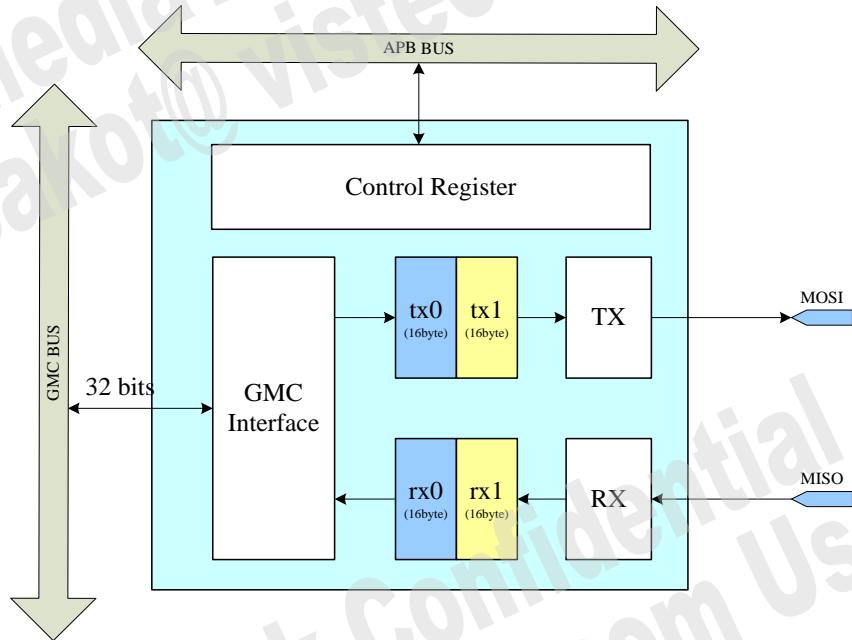


Figure 5-67 SPI Master Block Diagram

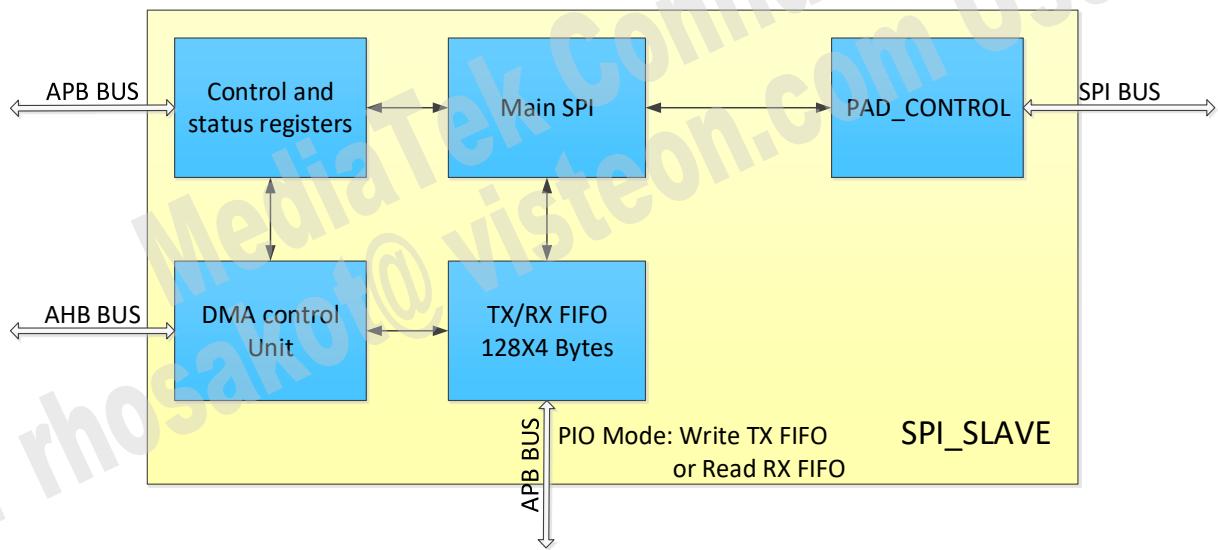


Figure 5-68 SPI Slave Block Diagram

Figure 5-68 shows the block diagram of SPI slave. It consists of Control and Status Register, DMA Control Unit, Main SPI, TX/RX FIFO and PAD_CONTROL. The PAD_CONTROL controls the SPI data capture and data transmit to/from SPI BUS. The Control and Status Register receives commands from system. The DMA control Unit is used to communicate with memory when SPI is set to the DMA mode. Both TX and RX have a 128 x 4-Byte FIFO for restoring data. Main SPI is the functional unit.

5.10.5 SPI AC Timing

5.10.5.1 SPI Electrical Characteristics

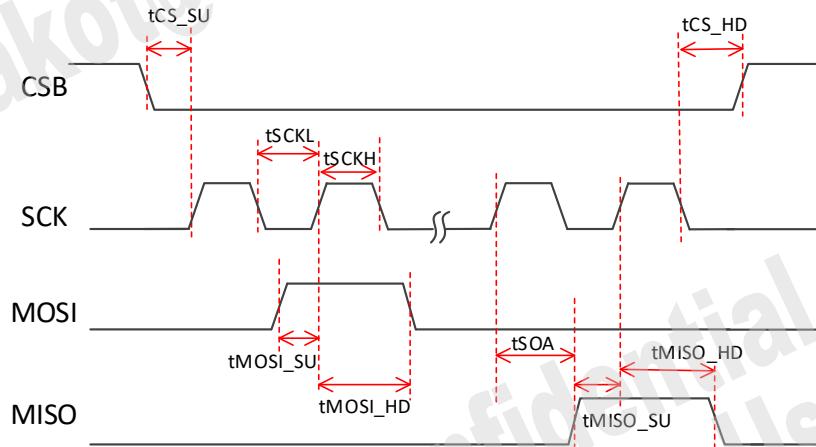


Figure 5-69 SPI Master/Slave Timing Diagram

Table 5-45 SPI Master Electrical Specifications

Symbol	Description	Performance			Unit
		Min.	Typ.	Max.	
fSPI _m ¹	SPI _m Bus SCK	-	-	51	MHz
tMOSI_SU	SDO to SCK Rising setup Time	9	-	-	ns
tMOSI_HD	SCK Rising to SDO hold Time	8	-	-	ns
tSCKL	SCK Low Pulse	8.4	-	-	ns
tSCKH	SCK High Pulse	7.2	-	-	ns
tCS_SU	CSB Falling to SCK Rising Setup Time	9.5	-	1235	ns
tCS_HD	SCK Falling to CSB Rising Hold Time	12	-	1635	ns
tMISO_SU ²	SDI to SCK Rising Setup Time requirement	0	-	-	ns
tMISO_HD ³	SCK Rising to SDI Hold Time requirement	0	-	-	ns

Notes:

1. The suggested max speed cannot be more than 2MHz for open drain 5V tolerance IO, and the value depends on the RC parameters.
2. To achieve the min value of tMISO_SU, the internal sample clock delay of SPI master should be adjusted.
3. MISO data valid time should be one cycle of fSPI_m.

Table 5-46 SPI Slave Electrical Specifications

Symbol	Description	Performance			Unit
		Min.	Typ.	Max.	
fSPI _s ¹	SPI _s Bus SCK	-	-	26	MHz
tMOSI_SU	SDO to SCK Rising setup Time requirement	1.5	-	-	ns
tMOSI_HD	SCK Rising to SDO hold Time requirement	7.5	-	-	ns
tSCKL	SCK Low Pulse	15	-	-	ns
tSCKH	SCK High Pulse	15	-	-	ns
tCS_SU	CSB Falling to SCK Rising Setup Time	15	-	-	ns
tCS_HD	SCK Falling to CSB Rising Hold Time	15	-	-	ns
tMISO_SU ²	SDI to SCK Rising Setup Time	1.5	-	-	ns
tMISO_HD	SCK Rising to SDI Hold Time	25	-	-	ns
tSOA	MISO(Data Output) Access time	31	-	37	ns

Notes:

1. If the SPI Master can adjust sample clock delay, the max value of SPI_s clock can be up to 51MHz, this case has shown in Table 5-47 below.
2. tMISO_SU value is based on 26MHz SPI_s clock.

Table 5-47 SPI Slave Electrical Specifications

Symbol	Description	Performance			Unit
		Min.	Typ.	Max.	
fSPI _s	SPI _s Bus SCK	-	-	51	MHz

Symbol	Description	Performance			Unit
		Min.	Typ.	Max.	
tMOSI_SU	SDO to SCK Rising setup Time requirement	1.5	-	-	ns
tMOSI_HD	SCK Rising to SDO hold Time requirement	7.5	-	-	ns
tSCKL	SCK Low Pulse	8.4	-	-	ns
tSCKH	SCK High Pulse	7.2	-	-	ns
tCS_SU	CSB Falling to SCK Rising Setup Time	9.5	-	-	ns
tCS_HD	SCK Falling to CSB Rising Hold Time	12	-	-	ns
tMISO_SU	SDI to SCK Rising Setup Time	-	-	-	ns
tMISO_HD	SCK Rising to SDI Hold Time	-	-	-	ns
tSOA	MISO(Data Output) Access time	31	-	37	ns

5.10.6 Register Definition

For register details, please refer to Chapter 3.8 of “MT2712 IVI Application Processor Registers”.

5.10.7 Programming Guide

Follow the steps below to perform SPI transmission:

1. Prepare the data in the memory with its start address being the “source address”.
2. Set up the timing and protocol for the SPI transmission (see Figure 5-66 above for detailed setup parameters).
3. Fill the “destination address”, which is the start address that users would like to place the received data, and “source address”, which is the start address to place the data to be transmitted, into registers SPI_RX_DST and SPI_TX_SRC, respectively.
4. Set up the CMD_ACT (bit0 of SPI_CMD) to start the transfer.
5. Get the data received from the buffer prepared starting from “destination address”.

Note: Figure 5-70 shows the generation of SCK, and detailed parameters are from register SPI_CFG0 of Chapter 3.9 of “MT2712 IVI Application Processor Registers”.

(e.g. Source clock is 100 MHz, SCL_LOW_COUNT is 1 and SCL_HIGH_COUNT is also 1, then the SCK is 25 MHz)

$$\text{SCK} = \frac{\text{Source clock}}{(\text{SCK_LOW_COUNT}+1) + (\text{SCK_HIGH_COUNT}+1)}$$

Figure 5-70 SCK Generation

5.11 MMC and SD Controller

5.11.1 Introduction

MMC and SD Controller (MSDC) fully supports

- SD memory card specification version 3.0
- SDIO card specification version 3.0
- MMC/eMMC5.0

5.11.2 Features

There are four ports in MT2712: MSDC0, MSDC1, MSDC2 and MSDC3.

- MSDC0 is used as MMC/eMMC
- MSDC1 and MSDC2 are used as SD/SDIO interface
- MSDC3 is used as SDIO

The detailed features of each MSDC are listed as follows:

MSDC0:

- Interface with MCU by AXI bus
- 64-bit data access on AXI bus
- 32-bit access for control registers
- 8-bit/16-bit/32-bit access for First In First Out (FIFO) in PIO mode
- Built-in 128-byte FIFO buffers for transmit and receive
- Built-in CRC circuit
- Support Basic DMA mode, Basic Descriptor mode and Enhanced Descriptor mode
- Interrupt capabilities
- Support eMMC speed mode:
 - Backwards compatibility with legacy MMC card
 - High Speed SDR mode
 - High Speed DDR mode
 - HS200 mode
 - HS400 mode
 - Support eMMC Boot up mode
 - Support MMC/eMMC bus width: 1 bit/4 bits/8 bits
 - Support AES (ECB/CBC/CTR/OFB/CFB mode)

MSDC1:

- Interface with MCU by AHB bus

- 32-bit access on AHB bus
- 32-bit access for control registers
- 8-bit/16-bit/32-bit access for FIFO in PIO mode
- Built-in 128-byte FIFO buffers for transmit and receive
- Built-in CRC circuit
- Support Basic DMA mode, Basic Descriptor mode and Enhanced Descriptor mode
- Interrupt capabilities
- Support bus speed mode:
- Default Speed mode
- High Speed mode
- SDR12 mode
- SDR25 mode
- SDR50 mode
- SDR104 mode
- DDR50 mode
- Support SD bus width: 1 bit/4 bits
- Support SDIO bus width: 1 bit/4 bits

MSDC2:

- Interface with MCU by AHB bus
- 32-bit access on AHB bus
- 32-bit access for control registers
- 8-bit/16-bit/32-bit access for FIFO in PIO mode
- Built-in 128byte FIFO buffers for transmit and receive
- Built-in CRC circuit
- Support Basic DMA mode, Basic Descriptor mode and Enhanced Descriptor mode
- Interrupt capabilities
- Support bus speed mode:
- Default Speed mode
- High Speed mode
- SDR12 mode
- SDR25 mode
- SDR50 mode
- SDR104 mode
- DDR50 mode
- Support SD bus width: 1 bit/4 bits
- Support SDIO bus width: 1 bit/4 bits

MSDC3:

- Interface with MCU by AXI bus
- 64-bit data access on AXI bus

- 32-bit access for control registers
- 8-bit/16-bit/32-bit access for FIFO in PIO mode
- Built-in 128-byte FIFO buffers for transmit and receive
- Built-in CRC circuit
- Support Basic DMA mode, Basic Descriptor mode and Enhanced Descriptor mode
- Interrupt capabilities
- Support bus speed mode:
- Default Speed mode
- High Speed mode
- SDR12 mode
- SDR25 mode
- SDR50 mode
- SDR104 mode
- DDR50 mode
- Support SDIO bus width: 1 bit/4 bits

5.11.3 Block Diagram

The system level block diagram of MSDC in MT2712 is shown as Figure 5-71.

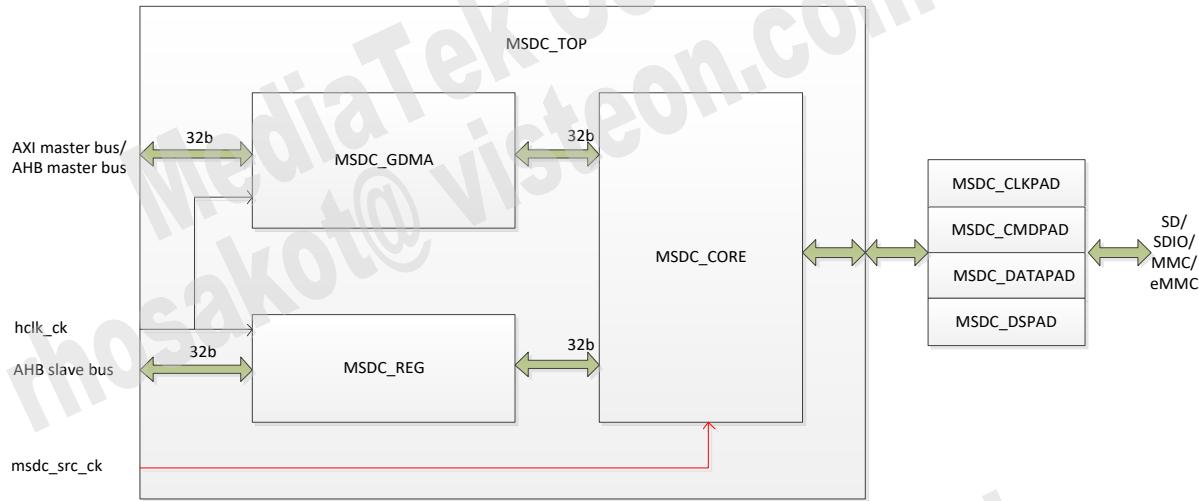


Figure 5-71 MSDC Block Diagram

MSDC is mainly composed of three parts: MSDC_GDMA, MSDC_CORE and MSDC_REG.

MSDC_REG: registers to configure the MSDC.

MSDC_CORE: main controller of MSDC, which implements the transfer between host and device.

MSDC_GDMA: DMA engine, which transfers data between MSDC and memory.

5.11.4 MSDC AC Timing

5.11.4.1 eMMC5.0 Interface

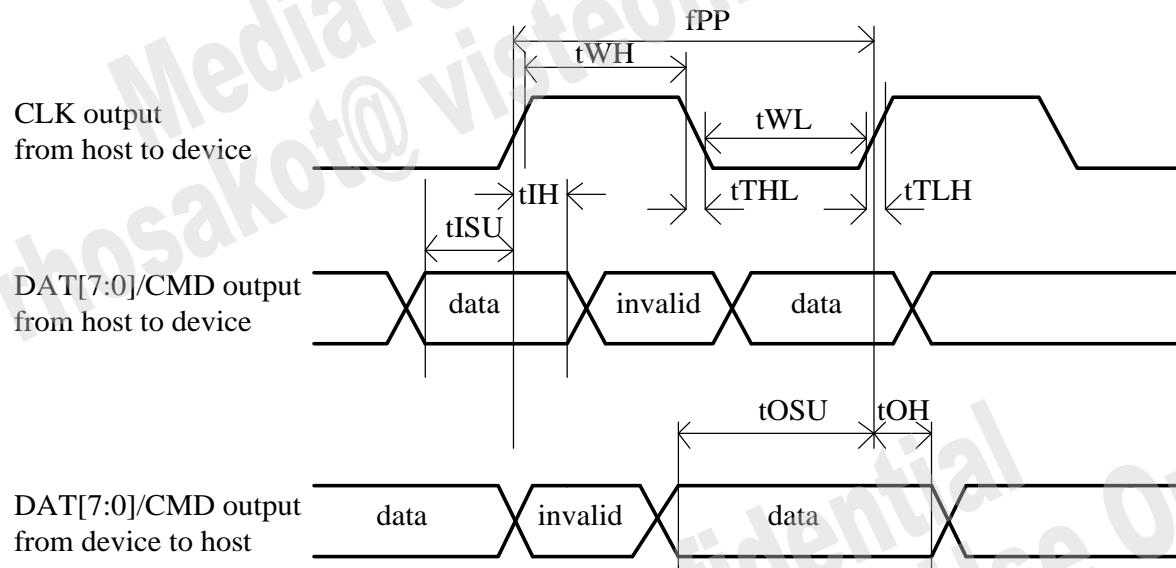


Figure 5-72 eMMC5.0 Timing Diagram (backward-compatible device interface)

Table 5-48 eMMC5.0 Timing Parameter (backward-compatible device interface)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK frequency	fPP	-	26	MHz
CLK low time	tWL	12	-	ns
CLK high time	tWH	12	-	ns
CLK rise time	tTLH	-	4	ns
CLK fall time	tTHL	-	4	ns
Host CMD/DAT output (reference to CLK)				
Output setup time	tISU	10	-	ns
Output hold time	tIH	10	-	ns
Host CMD/DAT input (reference to CLK)				
Input setup time ⁽¹⁾	tOSU	10	-	ns
Input hold time ⁽¹⁾	tOH	7	-	ns
Input setup time ⁽²⁾	tOSU	$t_{PERIOD}^{(3)}/2+7$	-	ns
Input hold time ⁽²⁾	tOH	$t_{PERIOD}^{(3)}/2$	-	ns
NOTE 1. Timing requirements when software configures the host internal sample clock to the falling edge by using the register for the host to sample the data from the device				

Parameter	Symbol	Min.	Max.	Unit
NOTE 2. Timing requirements when software configures the host internal sample clock to the rising edge by using the register for the host to sample the data from the device; Only need to meet either tOSU or tOH. Rising edge is used when BOOT uses backward-compatible mode.				
NOTE 3. tPERIOD = 1000/fPP				

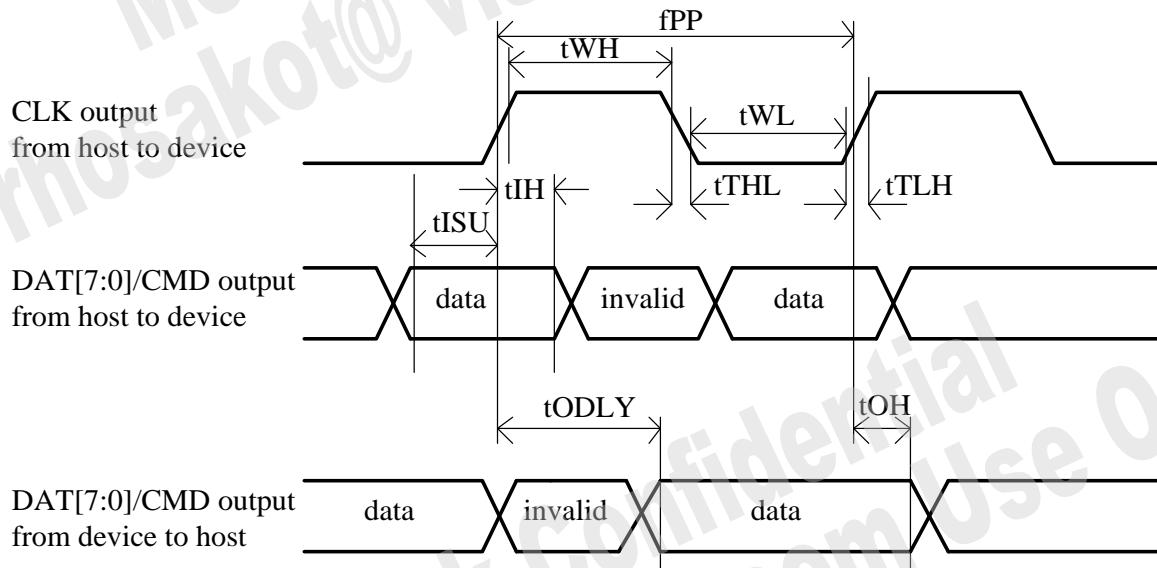


Figure 5-73 eMMC5.0 Timing Diagram (high-speed mode)

Table 5-49 eMMC5.0 Timing Parameter (high-speed mode)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK frequency	fPP	-	50	MHz
CLK low time	tWL	7.5	-	ns
CLK high time	tWH	7.5	-	ns
CLK rise time	tTLH	-	2	ns
CLK fall time	tTDL	-	2	ns
Host CMD/DAT output (reference to CLK)				
Output setup time	tISU	7	-	ns
Output hold time	tIH	7	-	ns
Host CMD/DAT input (reference to CLK)				
Input delay time ⁽²⁾	tTDL	-	14.7 ⁽¹⁾	ns
Input hold time ⁽²⁾	tOH	2	-	ns
Input delay time ⁽³⁾	tTDL	-	tPERIOD ⁽⁴⁾ /2-7	ns
Input hold time ⁽³⁾	tOH	tPERIOD ⁽⁴⁾ /2	-	ns
NOTE 1. Simulation result				
NOTE 2. Timing requirements when software configures the host internal sample clock to the falling edge by using the register for the host to sample the data from the device				

Parameter	Symbol	Min.	Max.	Unit
NOTE 3. Timing requirements when software configures the host internal sample clock to the rising edge by using the register for the host to sample the data from the device; Only need to meet either tODLY or tOH. Rising edge is used when BOOT uses high-speed mode.				
NOTE 4. tPERIOD = 1000/fPP				

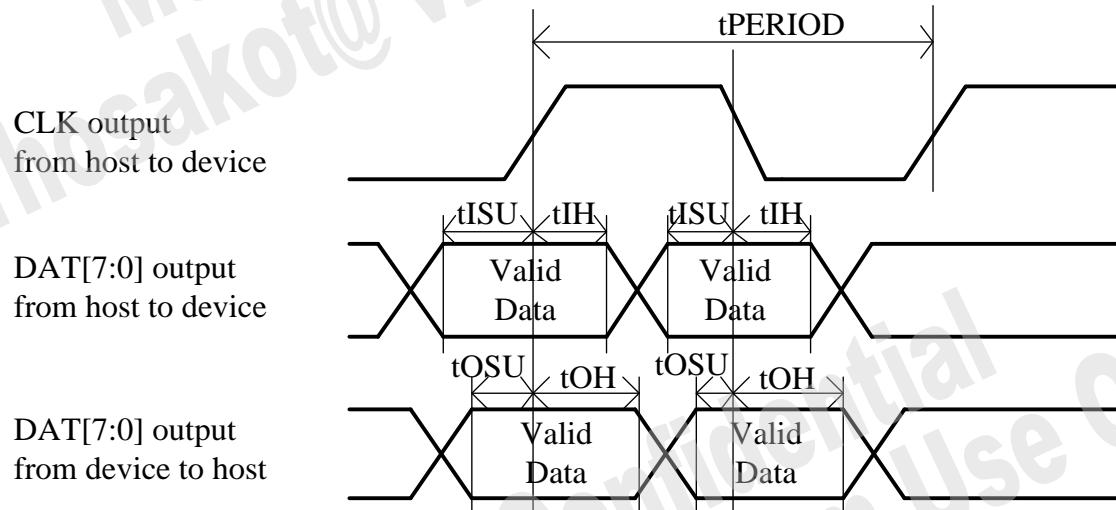


Figure 5-74 eMMC5.0 Timing Diagram (high-speed dual rate mode)

Table 5-50 eMMC5.0 Timing Parameter (high-speed dual rate mode)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK period	tPERIOD	20	-	ns
CLK Duty Cycle	-	47	53	%
Host DATA output (reference to CLK) ⁽¹⁾				
Output setup time	tISU	3	-	ns
Output hold time	tIH	3	-	ns
Host DATA input (reference to CLK) ⁽¹⁾				
Input setup time	tOSU	6 ⁽²⁾	-	ns
Input hold time	tOH	1	-	ns
NOTE 1. High speed dual rate mode CMD timing same to high-speed mode				
NOTE 2. Simulation result				

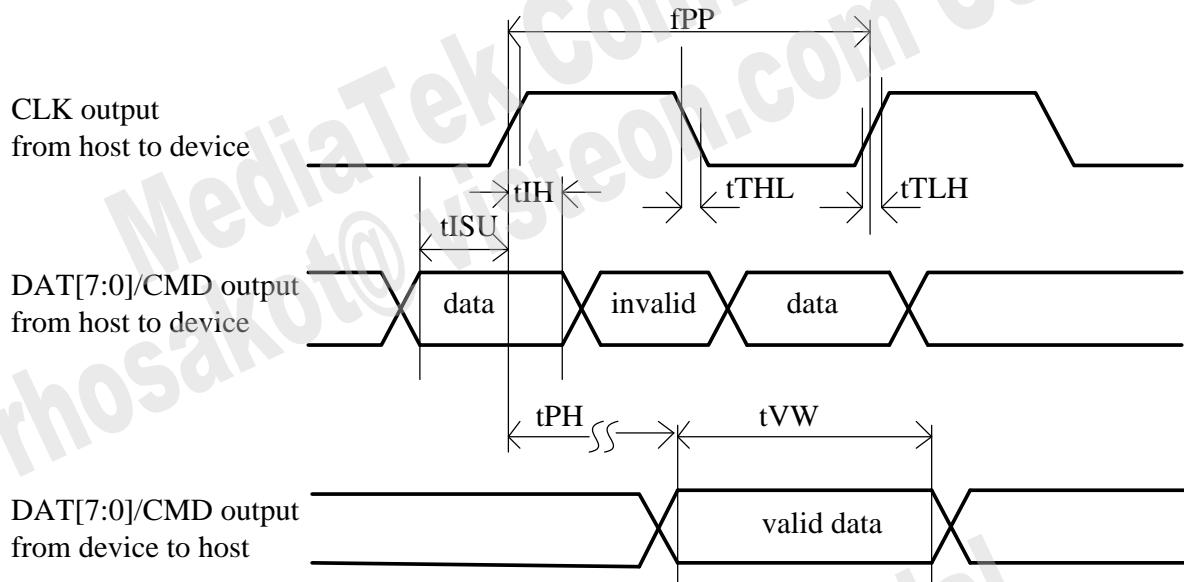


Figure 5-75 eMMC5.0 Timing Diagram (high-speed 200 mode)

Table 5-51 eMMC5.0 Timing Parameter (high-speed 200 mode)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK frequency	fPP	-	200	MHz
CLK rise time	tTLH	-	0.6	ns
CLK fall time	tTHL	-	0.6	ns
CLK duty cycle	-	40	60	%
Host CMD/DAT output (reference to CLK)				
Output setup time	tISU	1.8	-	ns
Output hold time	tIH	1.5	-	ns
Host CMD/DAT input (reference to CLK)				
Input data momentary from CLK output to DAT or CMD input	tPH	0	3	UI ⁽¹⁾
Input data valid window	tVW	2.6	-	ns

NOTE 1. Unit interval (UI) is one bit nominal time. For example, UI=5ns at 200MHz.

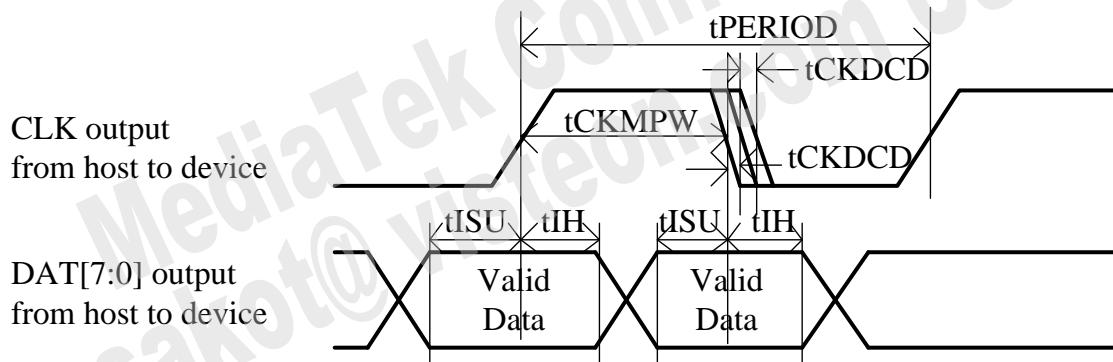


Figure 5-76 eMMC5.0 Timing Diagram (high-speed 400 mode host output)

Table 5-52 eMMC5.0 Output Parameter (high-speed 400 mode host output)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK period	tPERIOD	5	-	ns
CLK Duty distortion	tCKDCC	-	0.2	ns
Minimum pulse width	tCKMPW	2.3	-	ns
Host DAT output (reference to CLK) ⁽¹⁾				
Output setup time	tISU	0.7 ⁽²⁾	-	ns
Output hold time	tIH	0.7 ⁽²⁾	-	ns
NOTE 1. High speed 400 mode CMD timing same to HS200 mode				
NOTE 2. Simulation result				

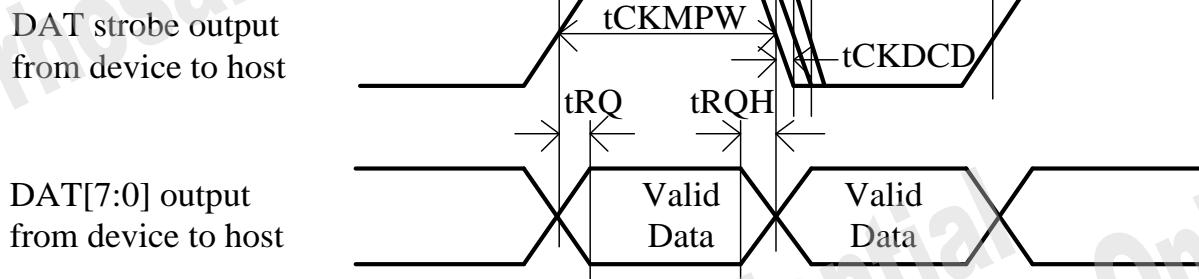
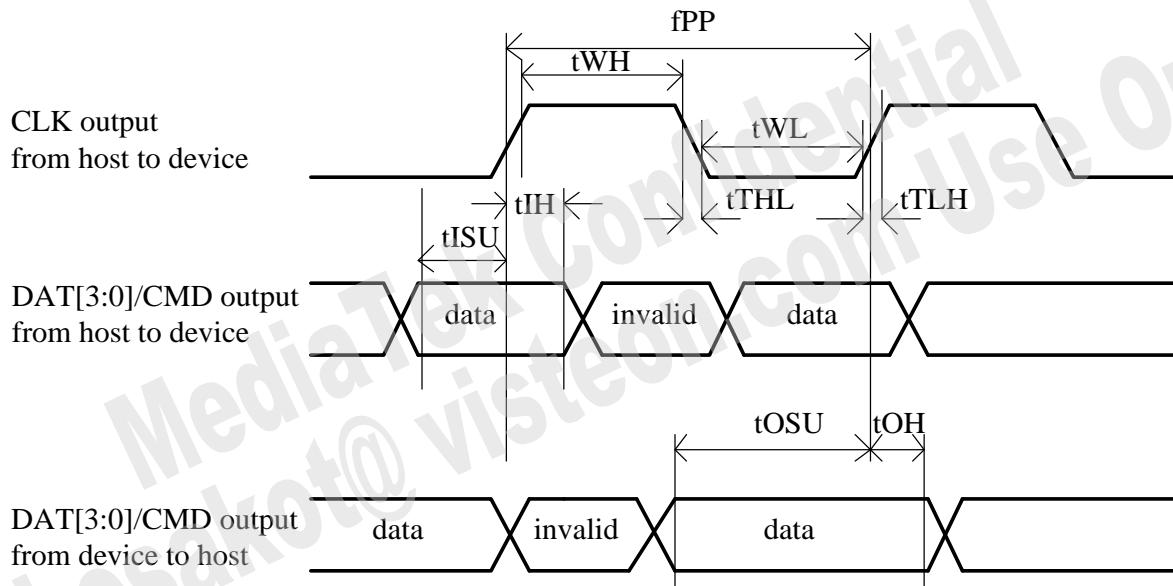


Figure 5-77 eMMC5.0 Timing Diagram (high-speed 400 mode host input)

Table 5-53 eMMC5.0 Output Parameter (high-speed 400 mode host input)

Parameter	Symbol	Min.	Max.	Unit
Data strobe output from device				
Data strobe period	tPERIOD	5	-	ns
Data strobe duty distortion	tCKDCD	-	0.3	ns
Minimum pulse width	tCKMPW	1.9	-	ns
Host DAT input (reference to data strobe)				
Input skew	tRQ	-	0.5	ns
Input hold skew	tRQH	-	0.5	ns
NOTE 1. Host controller does not care tRQ and tRQH mostly, but tPERIOD/2 – tRQ – tRQH should be more than 1.2ns.				

5.11.4.2 SD3.0 Interface**Figure 5-78 SD Timing Diagram (default-speed mode)****Table 5-54 SD Timing Parameter (default-speed mode)**

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK frequency	fPP	-	25	MHz
CLK low time	tWL	12	-	ns
CLK high time	tWH	12	-	ns
CLK rise time	tTLH	-	4	ns
CLK fall time	tTHL	-	4	ns
Host CMD/DAT output (reference to CLK)				

Parameter	Symbol	Min.	Max.	Unit
Output setup time	tISU	10	-	ns
Output hold time	tIH	10	-	ns
Host CMD/DAT input (reference to CLK)				
Input setup time ⁽¹⁾	tOSU	10	-	ns
Input hold time ⁽¹⁾	tOH	7	-	ns
Input setup time ⁽²⁾	tOSU	tPERIOD ⁽³⁾ /2+8	-	ns
Input hold time ⁽²⁾	tOH	tPERIOD ⁽³⁾ /2	-	ns
NOTE 1. Timing requirements when software configures the host internal sample clock to the falling edge by using the register for the host to sample the data from the device				
NOTE 2. Timing requirements when software configures the host internal sample clock to the rising edge by using the register for the host to sample the data from the device; Only need to meet either tOSU or tOH.				
NOTE 3. tPERIOD = 1000/fPP				

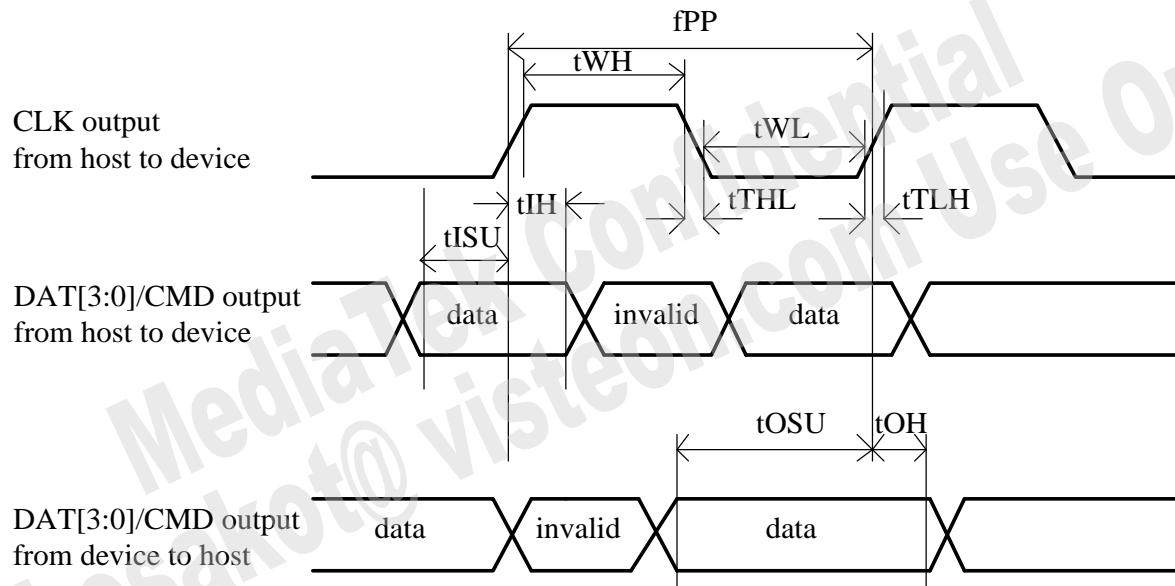


Figure 5-79 SD Timing Diagram (high-speed mode/SDR12/SDR25 mode)

Table 5-55 SD Timing Parameter (high-speed mode/SDR12/SDR25 mode)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK frequency	fPP	-	50 ⁽¹⁾	MHz
CLK low time	tWL	7.5	-	ns
CLK high time	tWH	7.5	-	ns
CLK rise time	tTLH	-	2	ns
CLK fall time	tTHL	-	2	ns
Host CMD/DAT output (reference to CLK)				

Parameter	Symbol	Min.	Max.	Unit
Output setup time	tISU	7	-	ns
Output hold time	tIH	4	-	ns
Host CMD/DAT input (reference to CLK)				
Input setup time ⁽³⁾	tOSU	8 ⁽²⁾	-	ns
Input hold time ⁽³⁾	tOH	1	-	ns
Input setup time ⁽⁴⁾	tOSU	tPERIOD ⁽⁵⁾ /2+8	-	ns
Input hold time ⁽⁴⁾	tOH	tPERIOD ⁽⁵⁾ /2	-	ns
NOTE 1. Max CLK frequency of SDR12 mode is 25MHz				
NOTE 2. Simulation result				
NOTE 3. Timing requirements when software configures the host internal sample clock to the falling edge by using the register for the host to sample the data from the device				
NOTE 4. Timing requirements when software configures the host internal sample clock to the rising edge by using the register for the host to sample the data from the device; Only need to meet either tOSU or tOH.				
NOTE 5. tPERIOD = 1000/fPP				

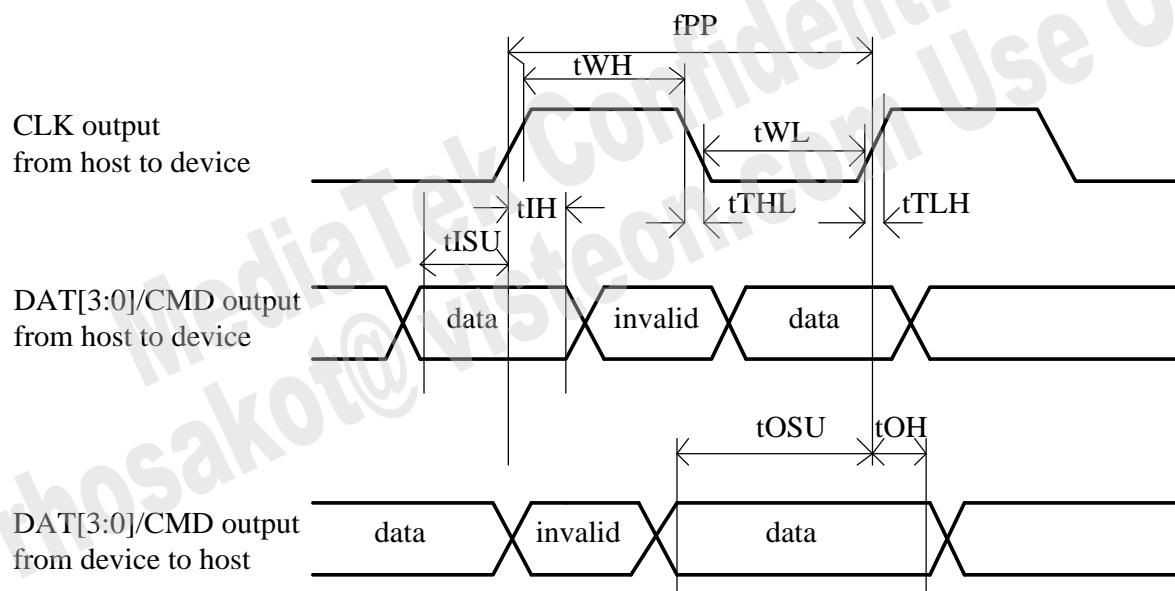


Figure 5-80 SD Timing Diagram (SDR50 mode)

Table 5-56 SD Timing Parameter (SDR50 mode)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK frequency	fPP	-	100	MHz
CLK low time	tWL	3	-	ns
CLK high time	tWH	3	-	ns

Parameter	Symbol	Min.	Max.	Unit
CLK rise time	tTLH	-	1.5	ns
CLK fall time	tTHL	-	1.5	ns
Host CMD/DAT output (reference to CLK)				
Output setup time	tISU	3.3	-	ns
Output hold time	tIH	3	-	ns
Host CMD/DAT input (reference to CLK)				
Input setup time ⁽¹⁾	tOSU	8	-	ns
Input hold time ⁽¹⁾	tOH	0	-	ns
Input setup time ⁽²⁾	tOSU	2 ⁽³⁾	-	ns
Input hold time ⁽²⁾	tOH	6 ⁽³⁾	-	ns
NOTE 1. Timing requirements when software configures the host internal sample clock to the falling edge by using the register for the host to sample the data from the device				
NOTE 2. Timing requirements when software configures the host internal sample clock to the rising edge by using the register for the host to sample the data from the device				
NOTE 3. Simulation result				

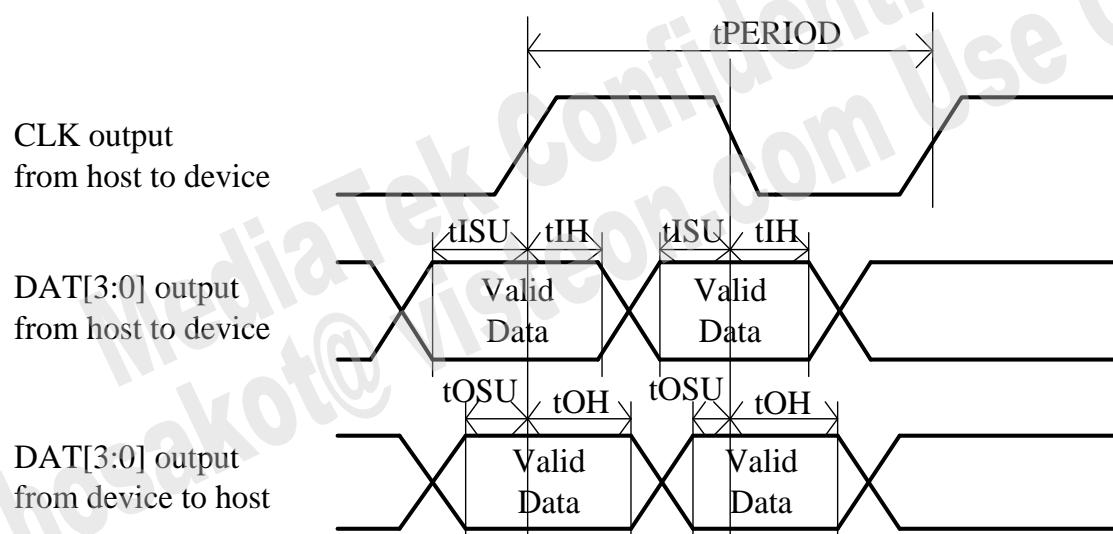


Figure 5-81 SD Timing Diagram (DDR50 mode)

Table 5-57 SD Timing Parameter (DDR50 mode)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK period	t_{PERIOD}	20	-	ns
CLK duty cycle	-	47	53	%
Host DAT output (reference to CLK) ⁽¹⁾				
Output setup time	tISU	3	-	ns
Output hold time	tIH	3	-	ns

Parameter	Symbol	Min.	Max.	Unit
Host DAT input (reference to CLK) ⁽¹⁾				
Input setup time	tOSU	7 ⁽²⁾	-	ns
Input hold time	tOH	1	-	ns
NOTE 1. DDR50 CMD timing same to high-speed mode				
NOTE 2. Simulation result				

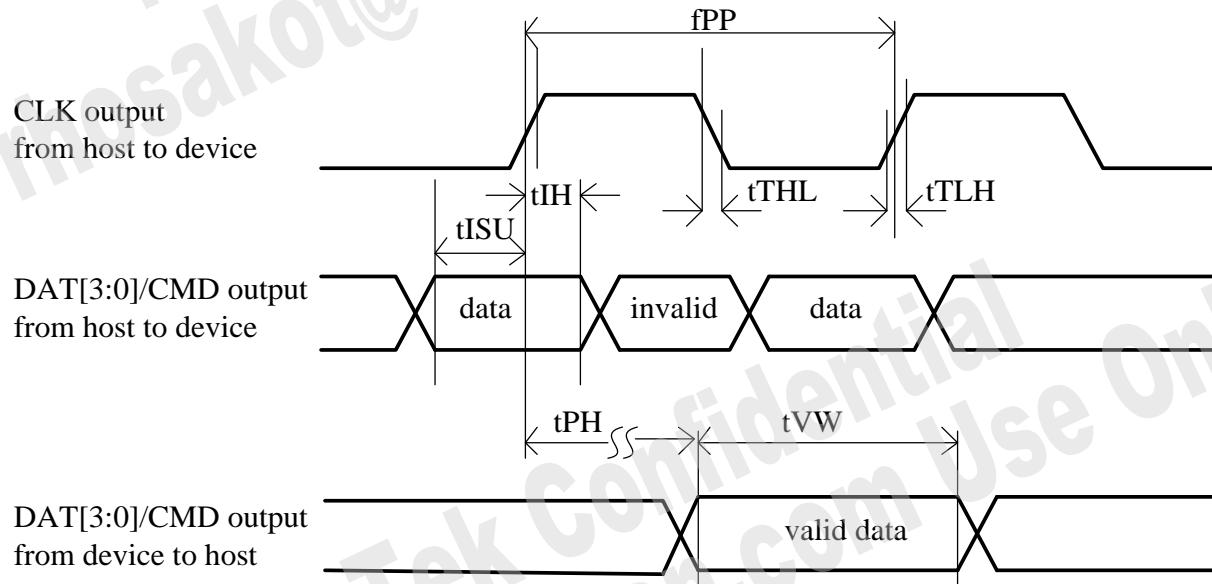


Figure 5-82 SD Timing Diagram (SDR104 mode)

Table 5-58 SD Timing Parameter (SDR104 mode)

Parameter	Symbol	Min.	Max.	Unit
CLK output from host				
CLK frequency	fPP	-	200	MHz
CLK rise time	tTLH	-	0.7	ns
CLK fall time	tTHL	-	0.7	ns
CLK duty cycle	-	40	60	%
Host CMD/DAT output (reference to CLK)				
Output setup time	tISU	1.8	-	ns
Output hold time	tIH	1.7	-	ns
Host CMD/DAT input (reference to CLK)				
Input data momentary from CLK output to DAT or CMD input	tPH	-	3	UI
Input data valid window	tVW	2.6	-	ns

5.11.5 Register Definition

For register details, please refer to Chapter 3.9 of “MT2712 IVI Application Processor Registers”.

5.11.6 Programming Guide

5.11.6.1 MSDC Read/Write

5.11.6.1.1 MSDC PIO Read

When the MSDC_CFG.PIO_MODE is set, MSDC works in PIO mode and receives data from the device. Software (SW) can read data from PIO_RXDATA register when the MSDC_FIFOCS.RXFIFO_CNT is not 0 (Data buffer empty). After the transfer is completed, an interrupt will be generated. SW needs to clear the interrupt bit after receiving it.

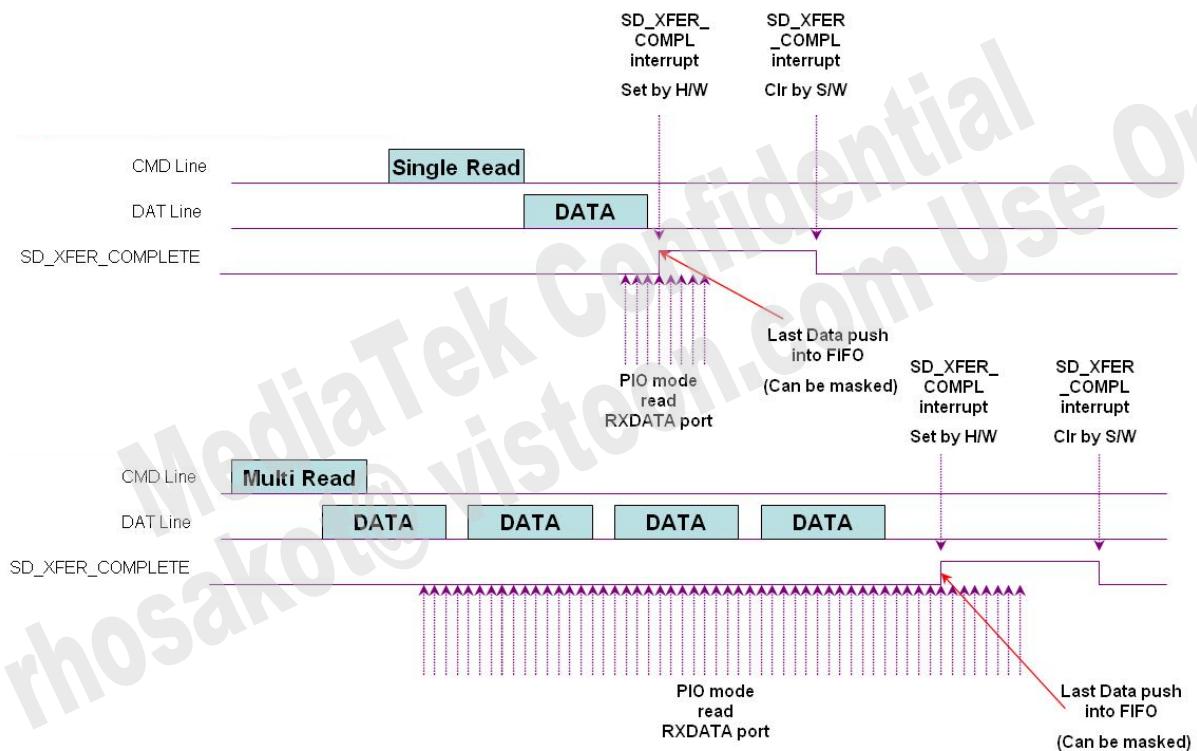


Figure 5-83 Single Block and Multi Block Read in PIO Mode

Figure 5-83 provides an example configuration of a system that supports a card slot and shared bus. The card slot bus and shared bus are separated, so that removing card does not influence the devices on shared bus. SD Bus signals except clock signal are connected together on the shared bus. Each device is selected by individual clock pins. Stopping clock for unselected device enables the system to reduce power consumption of devices. Even if SD bus clock is stopped, a SDIO device can generate interrupt by INT# pin to request a service to Host System. INT is an asynchronous, low-active, open-drain interrupt that can be wired or with interrupt pin of another device. Pull-up resistor is required for INT# signal.

5.11.6.1.2 MSDC PIO Write

When the MSDC_CFG.PIO_MODE is set, MSDC works in PIO mode and sends data to device. SW can write data to PIO_TXDATA register when the MSDC_FIFOCS.TXFIFOCNT is not 8'h80 (Data buffer full). After the transfer is completed, an interrupt will be generated. SW needs to clear the interrupt bit after receiving it.

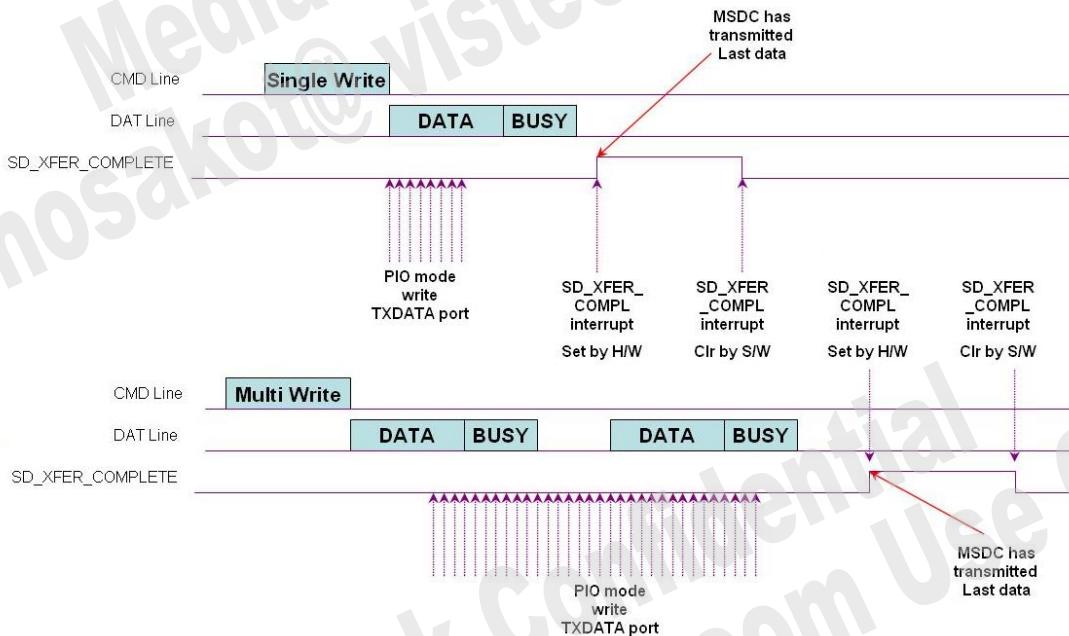


Figure 5-84 Single Block and Multi Block Write in PIO Mode

5.11.6.1.3 MSDC DMA Read

When the MSDC_CFG.PIO_MODE is not set, MSDC works in DMA mode. MSDC receives data from device and writes it to the target Dynamic Random-Access Memory (DRAM) address through MSDC_GDMA control. SW needs to configure the DMA_SA register as the start address in DRAM. After the transfer is completed, two interrupts will be generated. SW needs to clear the interrupt bit after receiving it.

- SD_XFER_COMPLETE is set when DMA controller has transferred all the data and CRC has been checked.
- (DMA_CTRL.LAST_BUF= 0) DMA_DONE is set when DMA controller has transmitted the total data set in DMA controller.
- (DMA_CTRL.LAST_BUF= 1) DMA_DONE is set at the same time as SD_XFER_COMPLETE.

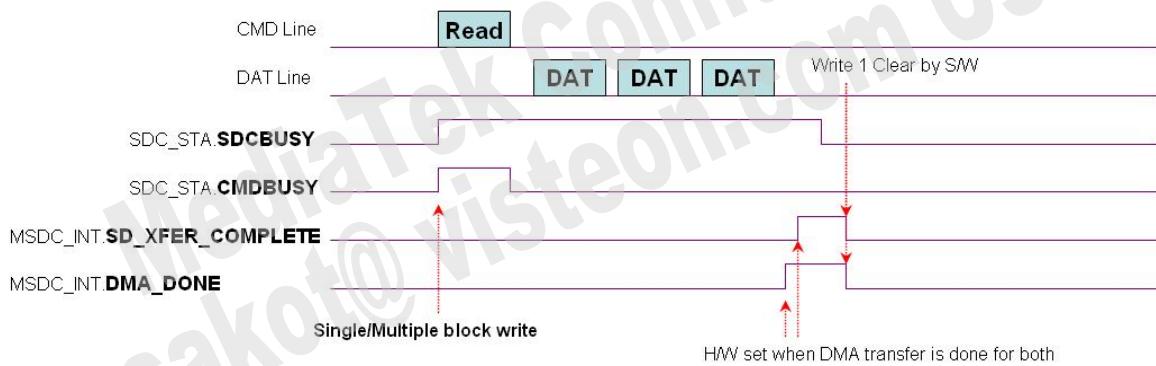


Figure 5-85 Single Block and Multi Block Read in DMA Mode

5.11.6.1.4 MSDC DMA Write

When the MSDC_CFG.PIO_MODE is not set, MSDC works in DMA mode. MSDC gets data from DRAM through MSDC_GDMA control and then writes them to device. SW needs to configure the DMA_SA register as the start address in DRAM. After the transfer is completed, two interrupts will be generated. SW needs to clear the interrupt bit after receiving it.

- SD_XFER_COMPLETE is set when DMA controller has transferred all the data and CRC has been checked.
- (DMA_CTRL.LAST_BUF = 0) DMA_DONE is set when DMA controller has transmitted the total data set in DMA controller.
- (DMA_CTRL.LAST_BUF= 1) DMA_DONE is set at the same time as SD_XFER_COMPLETE.

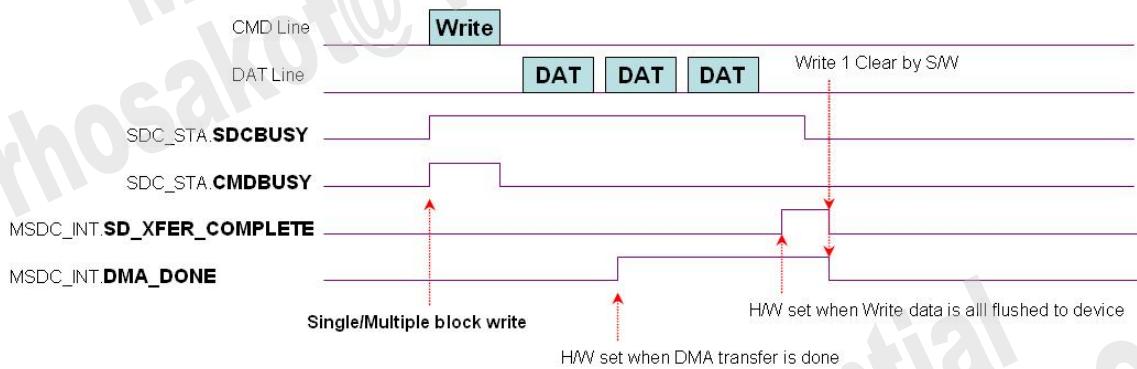


Figure 5-86 Single Block and Multi Block Write In DMA Mode

5.11.6.2 EMMC Boot

The eMMC4.5 specification defines two modes to access the boot code at pre-idle state. The detailed programming guide for these two modes is described in the following sections.

5.11.6.2.1 eMMC Boot Mode 0

For Mode 0 (EMMC_CFG0.BOOT_MODE = 0):

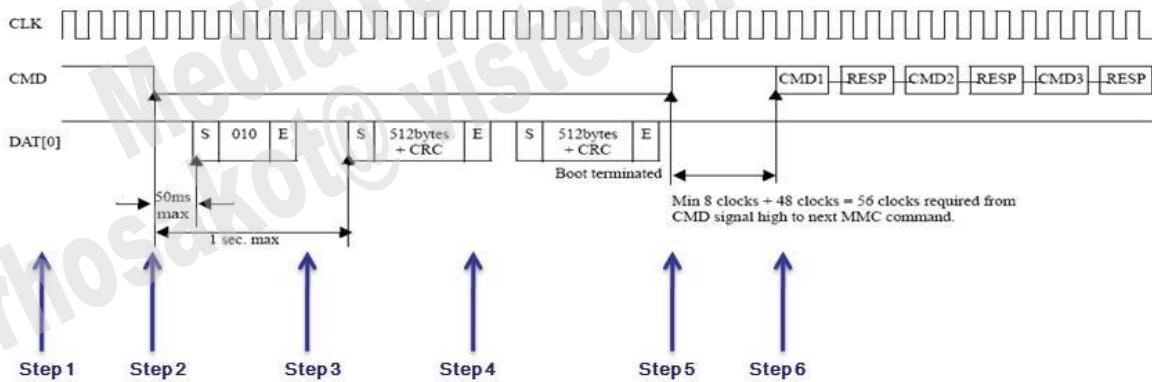


Figure 5-87 eMMC Boot Mode 0 Timing Diagram

[Step 1]

1. To force card into pre-idle state, SW can issue:
 - A command 0 with SDC_ARG = 0xFOFOFOFO, or
 - A HW reset by setting EMMC_IOCON.BOOT_RST = 1.

[Step 2]

1. Wait for SDC_STA.SDC_BUSY = 0.
2. Program EMMC_CFG0.BOOT_SUPPORT = 1.
3. Program EMMC_CFG0.BOOT_MODE = 0.
4. Program SDC_BLOCK_NUM = block number of boot data to be read.
5. If device's EXT_CSD[179].BOOT_ACK is 1'b0, SW should set EMMC_CFG0.BOOT_ACK_CHK_DIS = 1 and proceed to Step 7.
6. Program EMMC_CFG1.BOOT_ACK_TOC to a suitable value (about 50 ms), except 12'hFFF. The start point of BOOT_ACK_TOC counter is the same as the start point of "50 ms max" in the diagram.
7. Program EMMC_CFG1.BOOT_DAT_TOC to a desired value (about 1 s), except 20'hFFFF. The start point of BOOT_DAT_TOC counter is the same as the start point of "1 sec. max" in the diagram.
8. Program SDC_CMD = 0x02001000. (Multiple block read).
9. Set EMMC_CFG0.BOOT_START = 1 (The command line will go to low immediately).

[Step 3]

1. Poll EMMC_STS.BOOT_UP_STATE = 1.
2. Poll EMMC_STS.BOOT_ACK_RECV/ BOOT_ACK_TO/ BOOT_ACK_ERR.
3. If the ACK is time-out, the boot code cannot be read. So SW should proceed to Step 5.
4. If the ACK is CRCERR, there is still possibility for boot code to be outputted by card. After that MSDC will push it into RXFIFO. So after Step 5, SW should check the RXFIFOCNT and clears the RXFIFO.

5. If the ACK is received without error (BOOT_ACK_ERR/TO = 0 and BOOT_ACK_RECV = 1), then proceed to Step 4.

[Step 4]

1. Read the boot code in PIO mode. The size of boot code should be known by SW.

[Step 5]

1. Program SDC_ARG = 0x00000000.
2. Program SDC_CMD = 0x00001000.
3. Program EMMC_CFG0.BOOT_WAIT_DELAY to a suitable value (e.g. > 56 cycles).
4. Set EMMC_CFG0.BOOT_STOP = 1.
5. Poll EMMC_STS.BOOT_UP_STATE = 0.

[Step 6]

1. Program EMMC_CFG0.BOOT_SUPPORT = 0.
2. SW can issue the next command after checking and finding SDC_BUSY = 0.
3. Before performing data transfer (data line is involved) in normal mode. SW should clear FIFO first to make sure that the FIFO is all empty.
4. SW should clear MSDC_INT.CMD_RDY bit. (It is generated by CMD0).

5.11.6.2.2 eMMC Boot Mode 1

For Mode 1 (EMMC_CFG0.BOOT_MODE = 1):

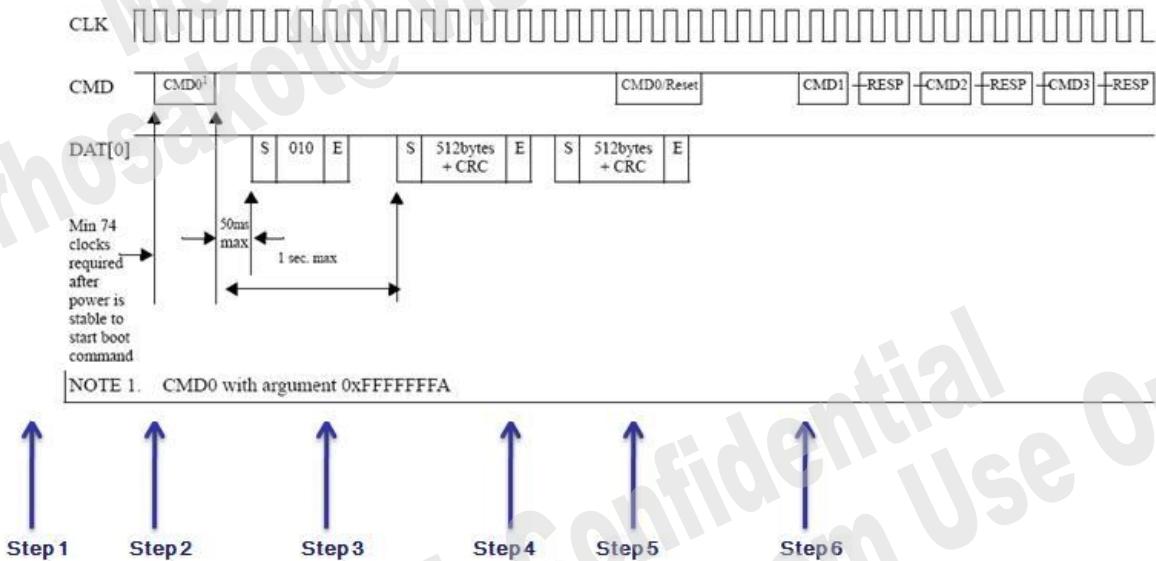


Figure 5-88 eMMC Boot Mode 1 Timing Diagram

[Step 1]

1. To force card into pre-idle state, SW can issue:
 - A command 0 with SDC_ARG = 0xFOFOFOFO, or
 - A HW reset by setting EMMC_IOCON.BOOT_RST = 1.

[Step 2]

1. Wait for SDC_STA.SDC_BUSY = 0.
2. Program EMMC_CFG0.BOOT_SUPPORT = 1.
3. Program EMMC_CFG0.BOOT_MODE = 1.
4. Program SDC_BLOCK_NUM = block number of boot data to be read.
5. If device's EXT_CSD[179].BOOT_ACK is 1'b0, SW should set EMMC_CFG0.BOOT_ACK_CHK_DIS = 1 and proceed to Step 7.
6. Program EMMC_CFG1.BOOT_ACK_TOC to a suitable value (about 50 ms), except 12'hFFF. The start point of BOOT_ACK_TOC counter is the same as the start point of "50 ms max" in the diagram.
7. Program EMMC_CFG1.BOOT_DAT_TOC to a desired value (about 1 s), except 20'hFFFFFF. The start point of BOOT_DAT_TOC counter is the same as the start point of "1 sec. max" in the diagram.
8. Program SDC_ARG = 0xFFFFFFFFA.
9. Program SDC_CMD = 0x02001000. (Multiple block read)
10. Set EMMC_CFG0.BOOT_START = 1 (The command 0 will be outputted immediately). Note that, there must be an enough time interval between setting SDC_CMD and EMMC_CFG0.BOOT_START. The interval should be at least one SDC_CLK clock period.

[Step 3]

1. Poll EMMC_STS.BOOT_UP_STATE = 1.
2. Poll EMMC_STS.BOOT_ACK_RECV/BOOT_ACK_TO/BOOT_ACK_ERR. If SW insists to terminate boot mode before EMMC_STS.BOOT_ACK_RECV is set, SW would proceed to Step 5.
3. If the ACK is time-out, the boot code cannot be read. So SW should proceed to Step 5.
 4. If the ACK is CRCERR, there is possibility for boot code to be outputted by card. Then MSDC will push it into RXFIFO. So after Step 5, SW should check the RXFIFOCNT and clear the RXFIFO.
5. If the ACK is received without error (BOOT_ACK_ERR/TO = 0 and BOOT_ACK_RECV = 1), then proceed to Step 4.

[Step 4]

1. Read the boot code in PIO mode. The size of boot code should be known by SW.

[Step 5]

1. Program SDC_ARG = 0x00000000.
2. Program SDC_CMD = 0x00001000.
3. Program EMMC_CFG0.BOOT_WAIT_DELAY to a suitable value (e.g. > 56 cycles).
4. Set EMMC_CFG0.BOOT_STOP = 1.
5. Poll EMMC_STS.BOOT_UP_STATE = 0.

6. If boot mode termination operation is initiated by SW before EMMC_STS.BOOT_ACK_RECV is set, as described in Step 3, then SW should repeat all steps in Step 5 again.

[Step 6]

1. Program EMMC_CFG0.BOOT_SUPPORT = 0.
2. SW can issue next command after checking SDC_BUSY = 0.
3. Before performing data transfer (data line is involved) in normal mode, SW should clear FIFO first to make sure that the FIFO is all empty.
4. SW should clear MSDC_INT.CMD_RDY bit. (It is generated by CMD0)
5. If BOOT_ACK_TO occurs, then SW should add one more CMD0 to ensure that card goes back to IDLE state.

5.11.6.3 MSDC Embedded DMA Control

5.11.6.3.1 Basic DMA Mode

The operation in basic DMA mode is the same as the conventional DMA operation. In this mode, the DMA controller moves a bulk of data from the source to MSDC.

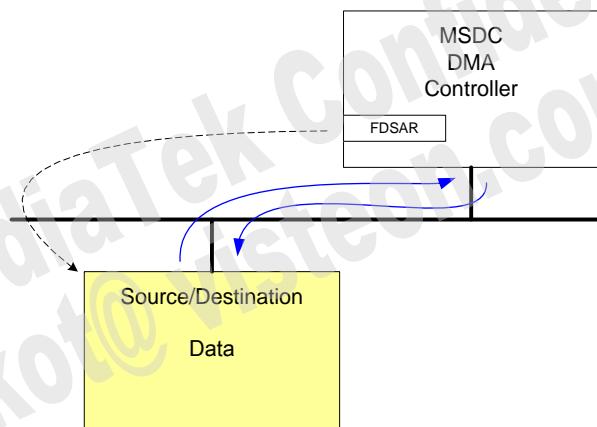


Figure 5-89 Basic DMA

5.11.6.3.2 Linked-List Based DMA Mode

The linked-list based DMA utilizes the descriptor structure to describe the data. Two types of DMA descriptors are defined for the purpose: General Packet Descriptor (GPD) and Buffer Descriptor (BD). For the fragmented data, one or more BDs are included to describe the discrete data. One GPD descriptor link can generate one SD command transaction.

For the flexibility of the descriptor structure, the linked-list DMA mode provides the hardware merging function to copy the fragmented source data to a continuous destination data buffer.

For the case of multiple source data buffers, as shown in Figure 5-90, each data buffer at the source is pointed by a DMA BD. All the DMA BDs associated with the multiple source data buffers are linked together as a list, and the list is pointed by a DMA GPD. The DMA control copies the fragmented source data into a single destination data buffer. It provides the hardware-implemented data merging function to reduce the computational power consumption of the embedded processor on the data copying.

Figure 5-90 shows the example of the linked-list based DMA mode on the DMA channel. Multiple destination data buffers are allowed in this mode. Each time the DMA controller finishes the DMA transfer operation for the data which belong to one DMA GPD/BD, the controller depends on the INT bit in GPD/BD to generate an interrupt to inform the firmware.

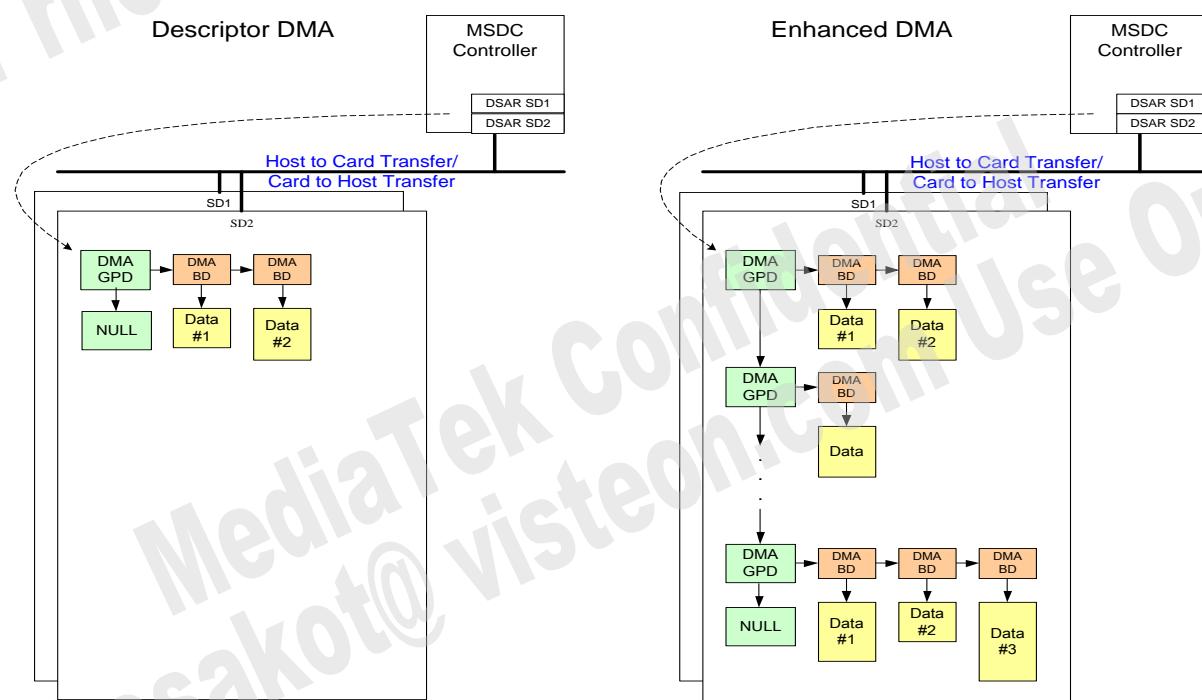


Figure 5-90 Descriptor and Enhanced DMA

5.11.6.4 SD3.0 Bus Voltage Switch

5.11.6.4.1 Voltage Switch Command

Figure 5-91 shows Voltage Switch Command (CMD11) definition. CMD11 can be executed in ready state and does not change its state. Even if the card is locked, CMD11 can be executed. Returning R1 type response means the card starts voltage switch sequence. If host detects no response, power cycle should be executed. There are four cases – listed as below – where the card indicates no response to CMD11.

1. The card does not support voltage switch.

2. The card supports voltage switch but ACMD41 is received with S18R = 0.
3. The card receives CMD11 not in ready state.
4. Signaling level is already switched to 1.8 V.

For all the above cases, CMD11 is treated as an illegal command.

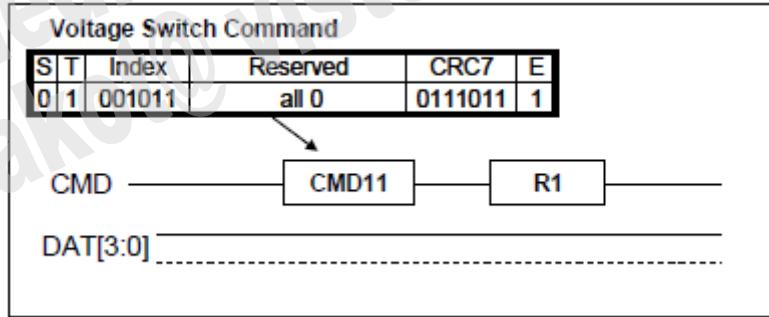


Figure 5-91 Voltage Switch Command

5.11.6.4.2 Initialization Sequence for UHS-I SD3.0 Card

When signaling level is 3.3 V, host repeats issuing ACMD41 with HCS = 1 and S18R = 1 until the response indicates device is ready. The argument (HCS and S18R) of the first ACMD41 is effective, but all the following ACMD41 should be issued with the same argument. If Bit 31 indicates ready, host needs to check CCS and S18R. The card indicates S18A = 0, which means that voltage switch is not allowed and host needs to use the current signaling level. S18A = 1 means that voltage switch is allowed and host issues CMD11 to invoke voltage switch sequence. By receiving CMD11, the card returns R1 response and starts voltage switch sequence. No response of CMD11 means that S18A was 0 and therefore host should not have sent CMD11. Completion of voltage switch sequence is checked by high level of DAT[3:0].

When entering the tran state, CARD_IS_LOCKED status in the R1 response should be checked (it is indicated in the response of CMD7). If the card is locked, CMD42 is required to unlock the card. If the card is unlocked, CMD42 can be skipped.

Figure 5-92 shows the sequence of commands to perform voltage switch and Figure 5-93 shows the initialization flow chart for UHS-I.

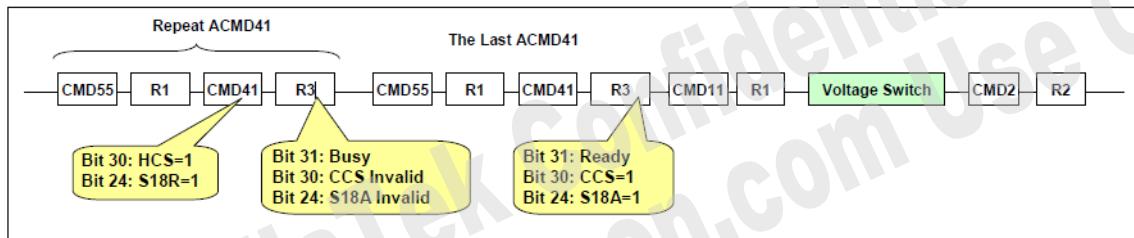


Figure 5-92 ACM41 Timing Followed by Voltage Switch Sequence

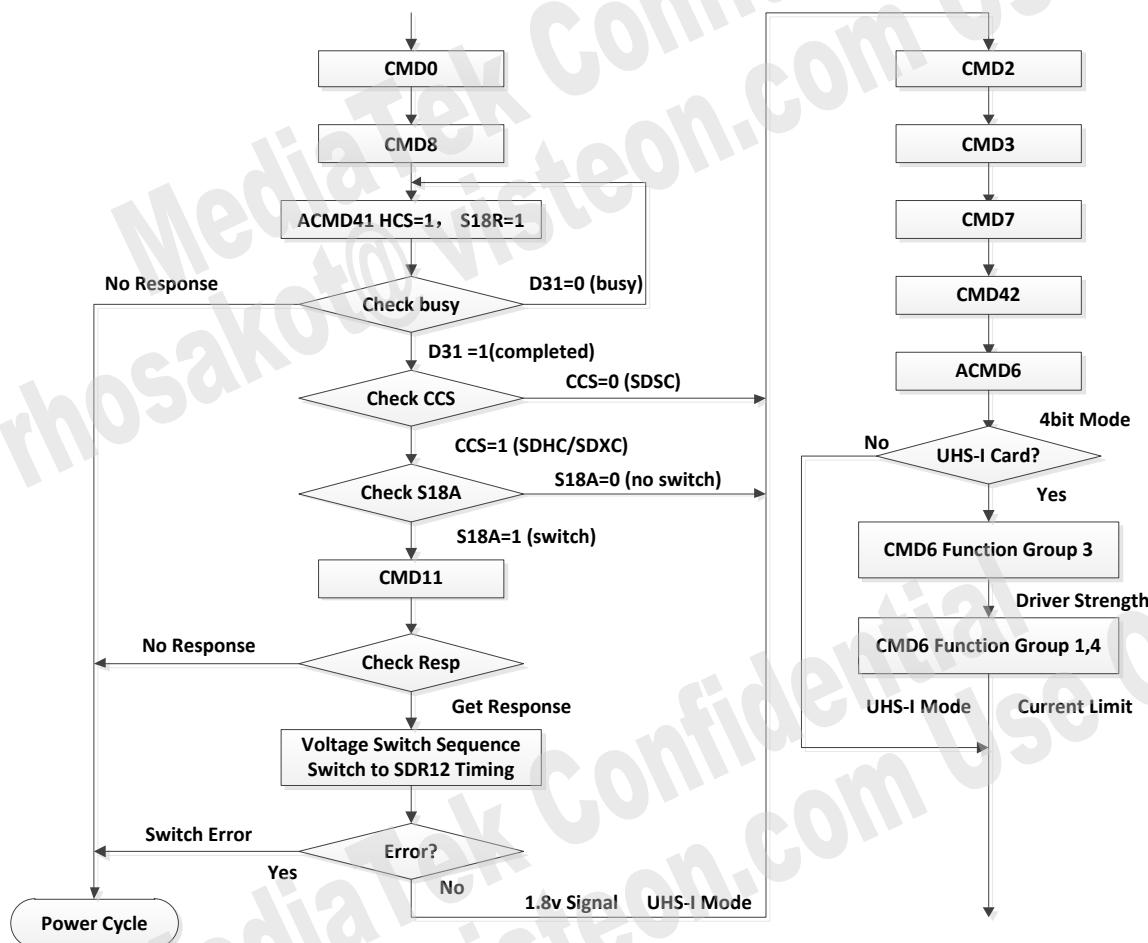


Figure 5-93 UHS-I Host Initialization Flow Chart

Clock frequency range shall be between 100 kHz – 400 kHz during the initialization sequence. Table 5-59 shows command (S18R) and response (S18A) combinations to switch signal voltage in ACMD41. S18R is defined in the command argument and indicates signal voltage switch request by host. S18A is defined in the response and indicates voltage switch acceptance by the card (voltage is not switched here). If signaling level is already 1.8 V, S18R is ignored and signal voltage switch sequence is not started. S18A=0 means that current signaling level is maintained.

Table 5-59 S18R and S18A Combinations

Current Signaling Level	S18R	S18A	Comment
3.3 V	0	0	1.8 V signaling is not requested
	1	0	The card does not support 1.8 V signaling
	1	1	Start signal voltage sequence
1.8 V	X	0	Already switched to 1.8 V

5.11.6.4.3 Voltage Switch Sequence

The detailed sequence description is shown in Figure 5-94. Change signaling level at the same time between host (MSDC) and card.

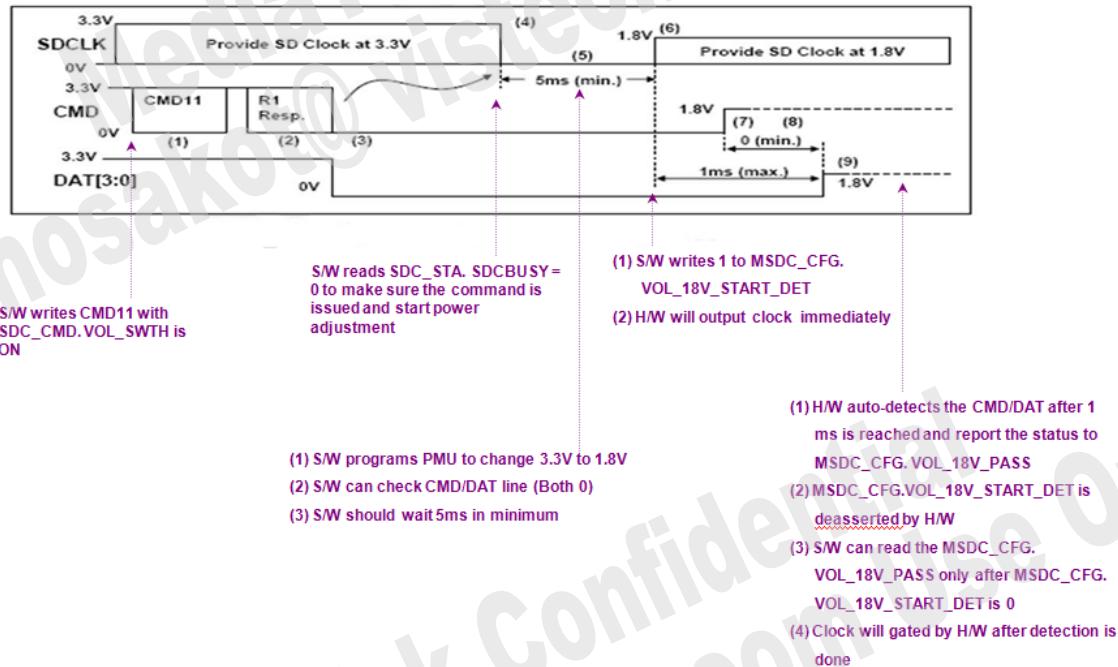


Figure 5-94 Signal Voltage Switch Sequence

1. SW writes CMD11 with SDC_CMD.VOL_SWTH on, to start voltage switch sequence.
2. Card returns R1 response.
3. Card drives CMD and DAT[3:0] to low immediately after the response.
4. SW reads SDC_STA.SDCBUSY = 0 to make sure CMD11 is issued, then stop CLK and start power adjustment.
5. SW programs PMU to change signaling level from 3.3 V to 1.8 V.
6. SW can detect whether the sequence starts by checking the signaling level of CMD and DAT[3:0]. If low level is detected, it means that sequence starts normally. If low level is not detected, SW should abort the sequence and execute power cycle.
7. SW should wait for a minimum of 5 ms, to keep CLK low for at least 5 ms.
8. After 5 ms from (4) and host voltage is stable, SW writes 1 to MSDC_CFG.VOL_18V_START_DET, and MSDC will output CLK at 1.8 V.
9. By detecting CLK, card drives CMD to high at 1.8 V for at least one clock and then stops driving.
10. If switching to 1.8 V signaling is completed successfully, card drives DAT[3:0] to high at 1.8 V for at least one clock and then stops driving. DAT[3:0] shall be high within 1 ms from the start of providing CLK.
11. MSDC will automatically detect the status of CMD and DAT[3:0] after 1 ms is reached. Status will be reported to MSDC_CFG.VOL_18V_PASS. MSDC_CFG.VOL_18V_START_DET is deasserted by MSDC self.

Note: SW can read MSDC_CFG.VOL_18V_PASS only after MSDC_CFG.VOL_18V_START_DET is 0.

5.12 NAND Flash Interface

5.12.1 Introduction

The NAND Flash Interface (NFI) and ECC engine (in NFI mode) can automatically generate Error Correction Code (ECC) syndrome bits when programming or reading from the device. If users allow it to store the syndrome bits in the spare area for each page, the HW_ECC mode can be used. Otherwise, users can prepare the data (may contain operating system information or ECC syndrome bits) for the spare area with other arrangements. In the former case, the NFI and ECC engine (in NFI mode) checks the syndrome bits when reading from the device. The ECC module features BCH code, which is capable of correcting up to 80-bit errors within one sector.

5.12.2 Features

MT2712 provides NAND Flash Interface SLC/MLC NAND. The NFI supports the following features:

- ONFI NAND Flash timing control
- Toggle NAND (v1.0) Flash timing control
- ECC (BCH code) acceleration, capable of 80-bit error correction (with ECC engine)
- Programmable page size and spare size
- Programmable FDM data size and protected FDM data size
- Word/byte access through APB bus
- DMA for massive data transfer
- Latch sensitive interrupt to indicate the ready state for read, program and erase operation
- Programmable wait states, command/address setup and hold time, read enable hold time and write enable recovery time
- 2-chip selection for NAND Flash parts.
- 8-bit TOGGLE/ONFI NAND interface.

5.12.3 Block Diagram

Figure 5-95 shows the architecture of NFI architecture.

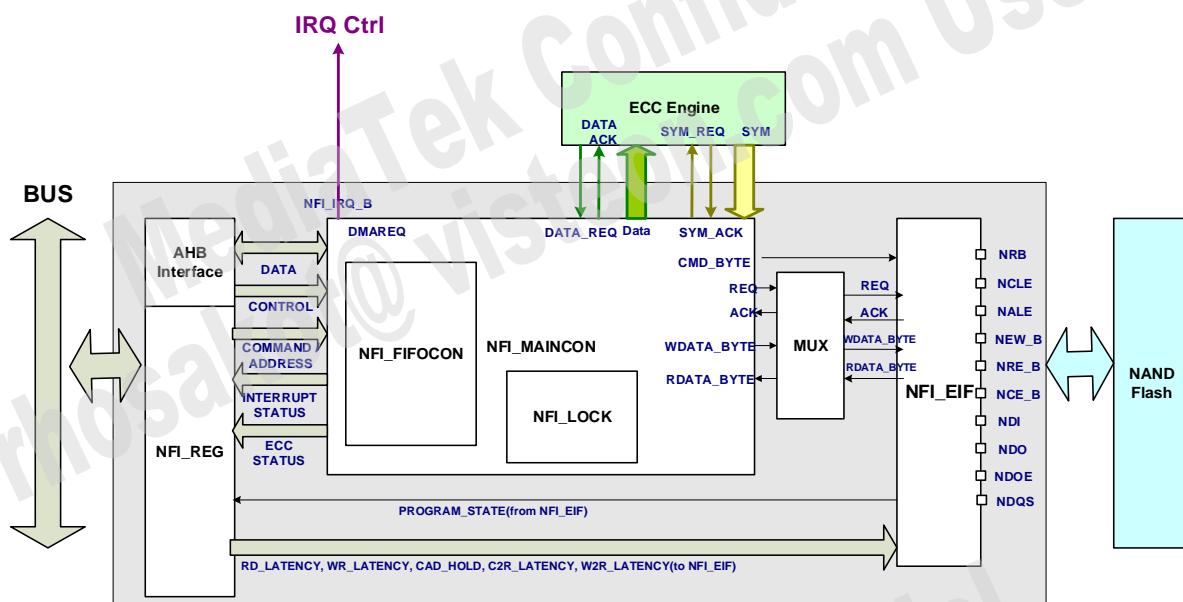


Figure 5-95 NFI Block Diagram

NFI uses APB slave bus for accessing register configuration and data read/write, and uses AHB master bus for faster data read/write. It also supports Interrupt Request (IRQ) which is level active for the interrupt process. The ECC Engine is used for encoding/decoding user data when needed. NFI uses standard protocol for communication with NAND devices.

The NAND Flash controller has three clocks: bus clock for apb and ahb bus, module clock for NFI controller logic and ecc clock for ECC Engine speeding up if better performance is needed.

5.12.4 NFI AC Timing

5.12.4.1 ONFI1.0 Interface Output Timing

The period of nfi clock (operation frequency 91 MHz). The AC timing for output timing is shown in the table below.

Table 5-60 NFI ONFI1.0 Command/Address/Data Write Access Timing

Parameter	Description	Min.	Max.	Unit
t_{DS}	Write data setup time	22	-	ns
t_{DH}	Write data hold time	8	-	ns
t_{WP}	Write low pulse time	22	-	ns
t_{WH}	Write high pulse time	10	-	ns
t_{CS}	CE setup time	58	-	ns
t_{CH}	CE hold time	32	-	ns
t_{CLS}	Command latch enable setup time	54	-	ns
t_{CLH}	Command latch enable hold time	20	-	ns

Parameter	Description	Min.	Max.	Unit
t_{ALS}	Address latch enable setup time	56	-	ns
t_{ALH}	Address latch enable hold time	18	-	ns
t_{WC}	Write cycle time	32	-	ns

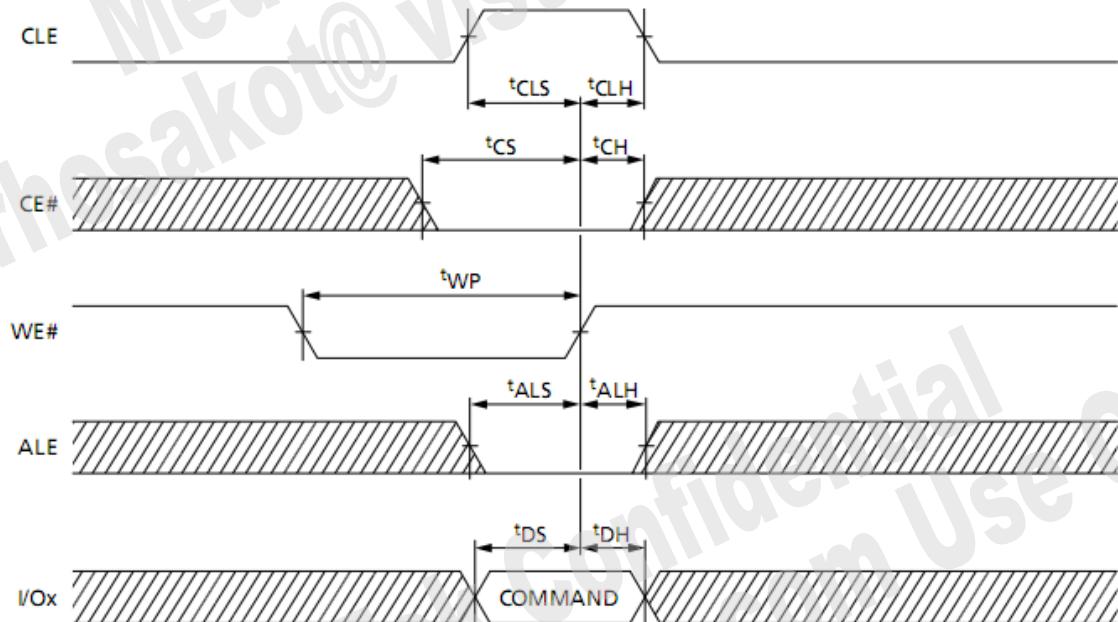


Figure 5-96 NFI ONFI1.0 Command Input Cycle

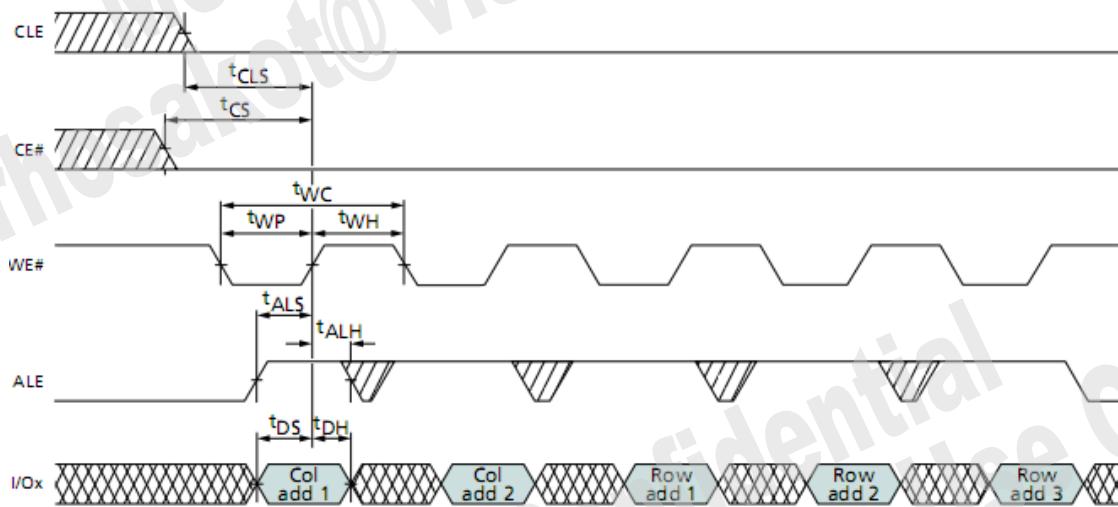


Figure 5-97 NFI ONFI1.0 Address Input Cycle

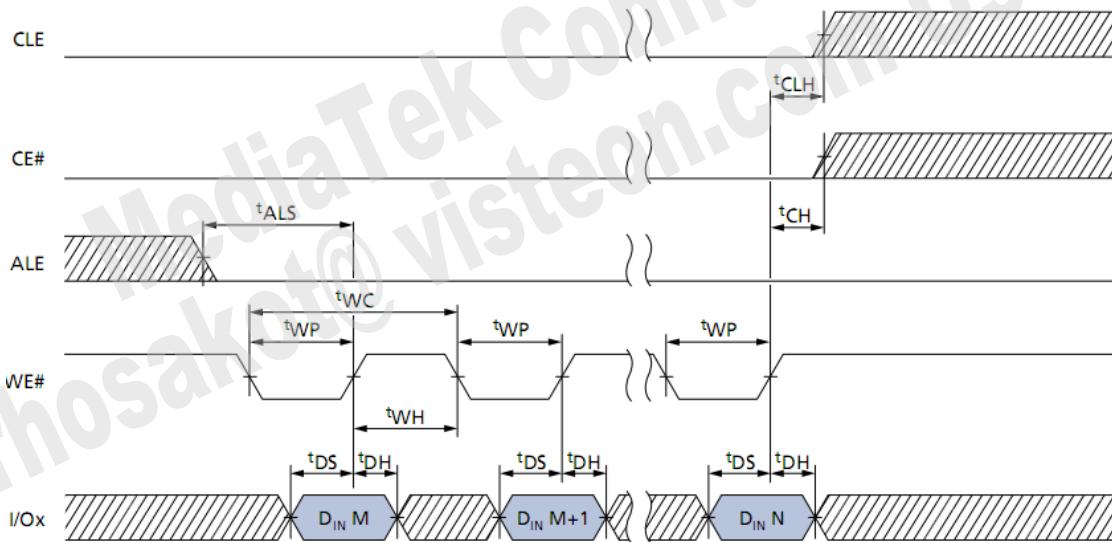


Figure 5-98 NFI ONFI1.0 Consecutive Data Write Cycles

5.12.4.2 ONFI1.0 Interface Input Timing

The AC timing for input timing is shown in the table below.

Table 5-61 NFI ONFI1.0 Read Access Timing

Parameter	Description	Min.	Max.	Unit
t _{REA}	RE access time	10	40	ns
t _{RR}	Ready to RE LOW	130	-	ns
t _{RP}	Read low pulse time	21	-	ns
t _{REH}	Read high pulse time	11	-	ns
t _{RHOH}	RE HIGH to output hold	15	-	ns
t _{RHZ}	RE HIGH to output High-Z	0	340	ns
t _{COH}	CE HIGH to output hold	32	-	ns
t _{RC}	Read cycle time	32	-	ns

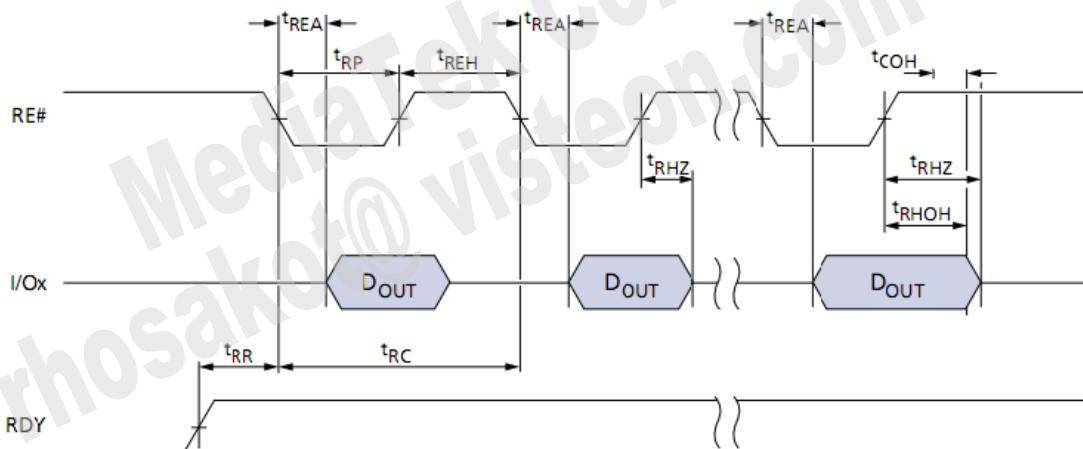


Figure 5-99 NFI ONFI1.0 Serial Read Cycle

5.12.4.3 Toggle1.0 Interface Output Timing

The period of NFI clock (operation frequency 91MHz). The AC timing for output timing is shown in the table below.

Table 5-62 NFI Toggle1.0 Write Access Timing

Parameter	Description	Min.	Max.	Unit
t_{CS}	CE setup time	232	-	ns
t_{CH}	CE hold time	142	-	ns
t_{CALS}	CLE/ALE setup time	230	-	ns
t_{CALH}	CLE/ALE hold time	18	-	ns
t_{WP}	WE low pulse time	24	-	ns
t_{CAS}	CMD/ADR setup time	20	-	ns
t_{CAH}	CMD/ADR hold time	10	-	ns
t_{WH}	WE high pulse time	8	-	ns
t_{WPRE}	Write Preamble	220	-	ns
t_{DS}	Data setup time	4	-	ns
t_{DH}	Data hold time	4	-	ns
t_{DQSH}	DQS high pulse	5.5	-	ns
t_{DQLS}	DQS low pulse	5.5	-	ns
t_{DSC}	Data strobe cycle time	11	-	ns
t_{WPST}	Write postamble	185	-	ns
t_{WPSTH}	Write postamble hold time	230	-	ns

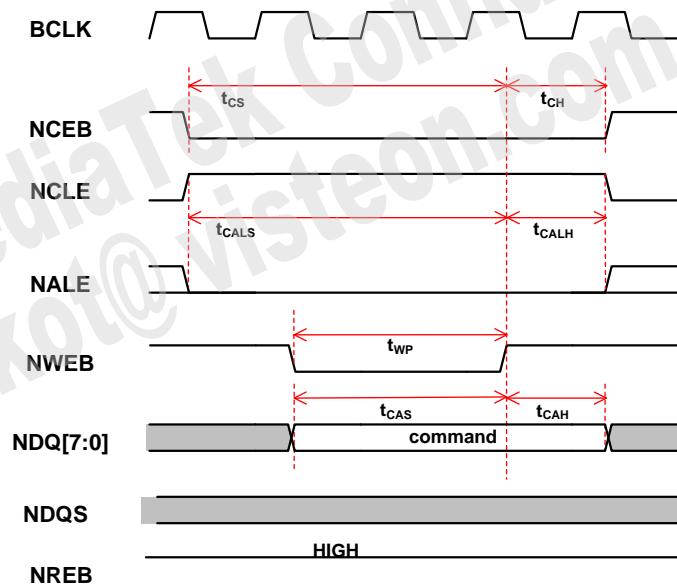


Figure 5-100 NFI Toggle1.0 Command Input Cycle

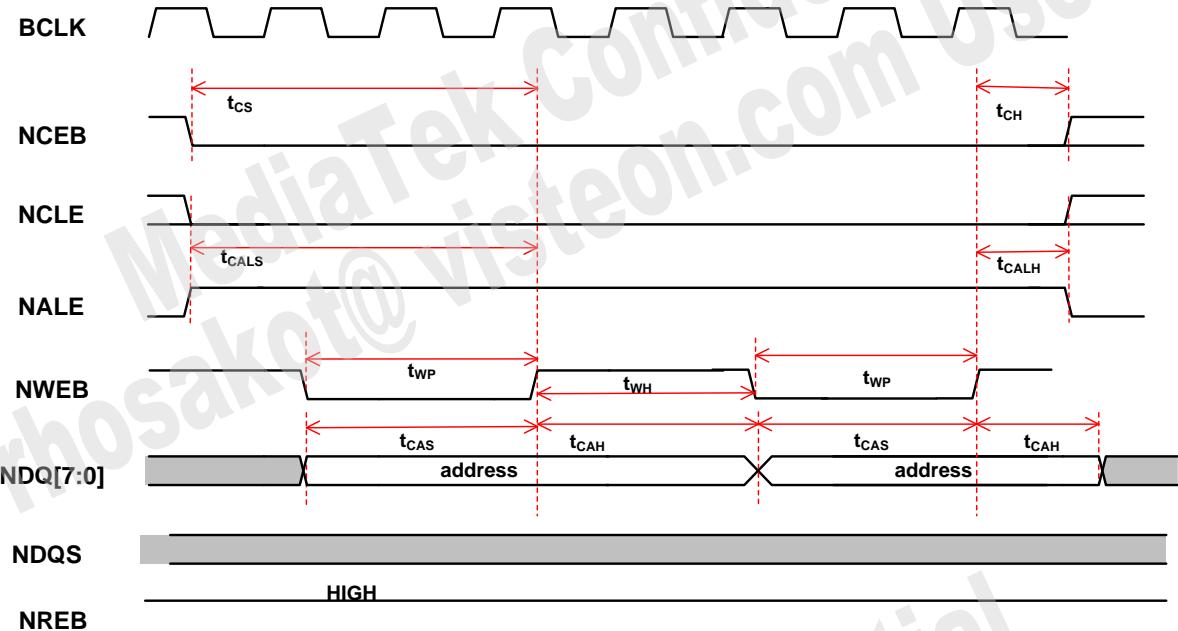


Figure 5-101 NFI Toggle1.0 Address Input Cycle

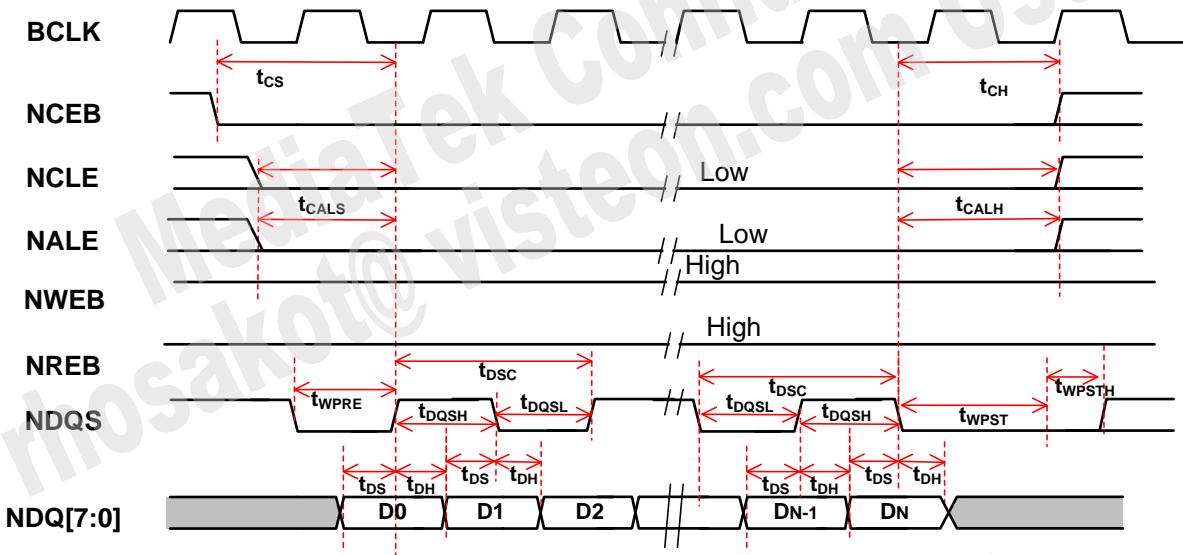


Figure 5-102 NFI Toggle1.0 Data Write Cycle

5.12.4.4 Toggle1.0 Interface Input Timing

The AC timing for input timing is shown in the table below.

Table 5-63 NFI Toggle1.0 Read Access Timing

Parameter	Description	Min.	Max.	Unit
t_{RPRE}	Read Preamble time	18	-	ns
t_{REH}	RE high pulse time	5.5	-	ns
t_{RP}	RE low pulse time	5.5	-	ns
t_{RC}	RE read cycle time	11	-	ns
t_{RPST}	Read postamble time	212	-	ns
t_{RPSTH}	Read postamble hold time	32	-	ns
t_{CHZ}	CE High to output z	30	-	ns
t_{DQSRE}	RE to DQS/DQ delay	0	30	ns
t_{DQSQ}	Output skew among DQ and DQS	-	800	ps
t_{QH}	Output hold time from DQS	Min (tREH, tRP) - tQHS	-	ns
t_{QHS}	DQS hold skew factor (up to device)	-	800	ps
t_{DVW}	Output data valid window (up to device)	$t_{QH} - t_{DQSQ}$	-	ns

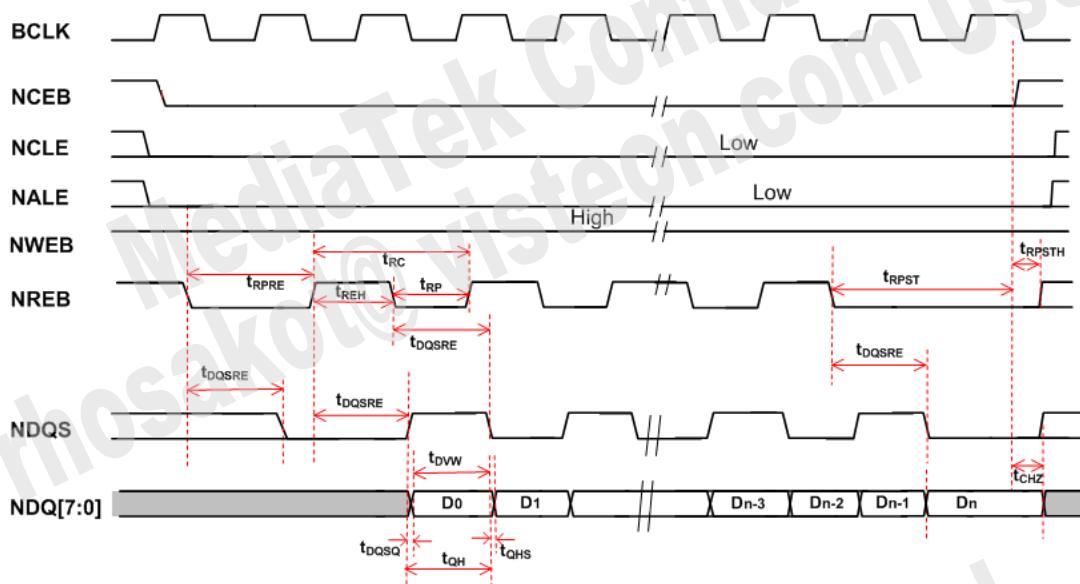


Figure 5-103 NFI Toggle1.0 Data Read Cycle

5.12.5 Register Definition

For register details, please refer to Chapter 3.10 of “MT2712 IVI Application Processor Registers”.

5.12.6 Programming Guide

The function of NFI is to control the IC plug-in NAND devices for data Program and Read, by means of sending control signals, such as CE, ALE, CLE, WE, and RE, to NAND. The control signals may also need to occupy the DATA line to send Command, Address and Data. Users can complete the operation of the NAND, such as Reset, Read NAND ID, Read Status, Block Erase, Page Program, Page Read, and etc. The control signals that satisfy the demand of NAND device timing diagram can be sent to NAND by configuring the specific registers of the NFI. Through the related registers, software adjusts the timing of NAND according to the actual situation of the timing parameter, which can easily handle the NAND as per user’s demand.

5.13 Serial Flash Controller

5.13.1 Introduction

The Serial Flash Controller is a platform which can send different commands to flash to program or read. It has two different modes to read flash data: Central Processing Unit (CPU) direct access or flash Direct Memory Access (DMA). The system also can boot up from flash.

5.13.2 Features

The module supports the following features:

- CPU accesses serial flash
 - Checksum for serial flash read data
 - System boots up from serial flash.
 - Support 4-Byte address mode, and compatible with 3-Byte address mode.
 - Support 4-bit output & 4-bit I/O read mode, and compatible with single-bit mode and dual-bit mode.
 - Read serial flash data through direct memory map, or DMA path.

5.13.3 Block Diagram

The module has three buses:

- APB slave for serial flash controller reading/writing register.
 - AXI slave for CPU directly accessing SPI NOR device through memory map
 - AXI master for serial flash DMA moving data from SPI NOR to DRAM/SRAM.

As shown in Figure 5-104, flash_arb insures only one mode accesses the SPI NOR device; sf_prefetch has a 32*128 SRAM to store pre-fetch data; macro_sf_top has a delay-chain to adjust I/O bus skew.

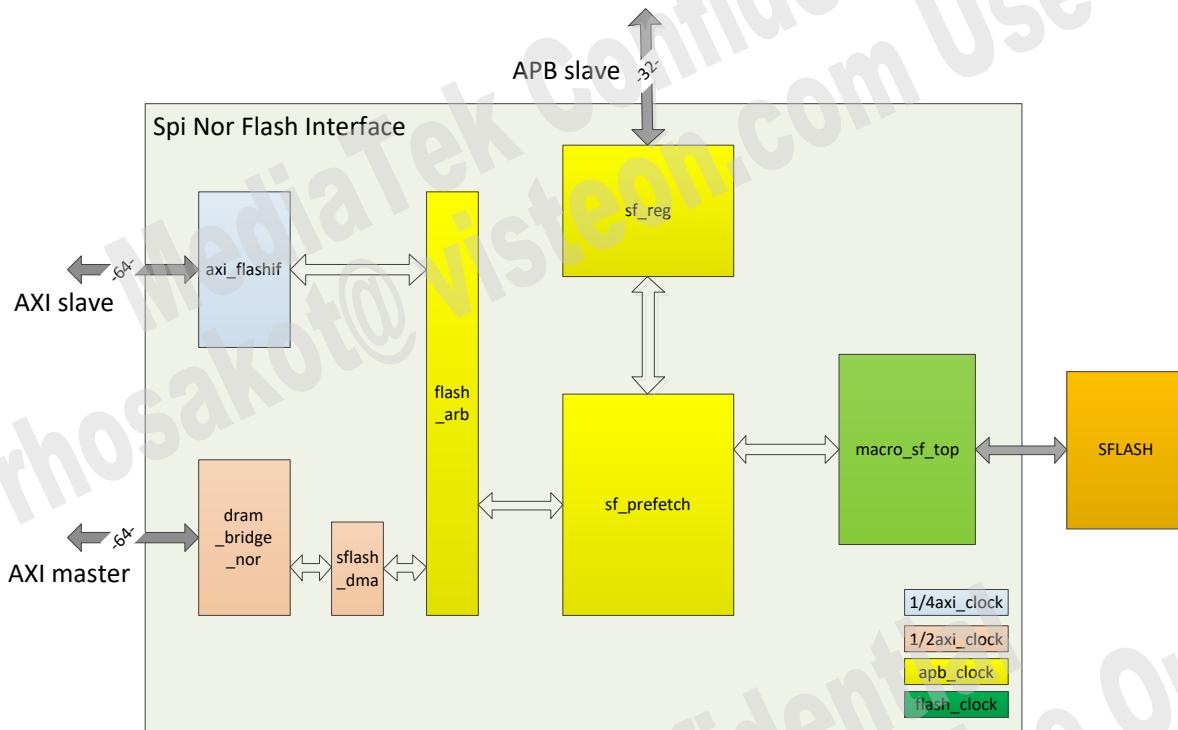


Figure 5-104 Flashif Block Diagram

5.13.4 Flashif AC Timing

5.13.4.1 Serial Flash Electrical Characteristics

The serial NOR flash interface AC timing characteristics are described in Figure 5-105:

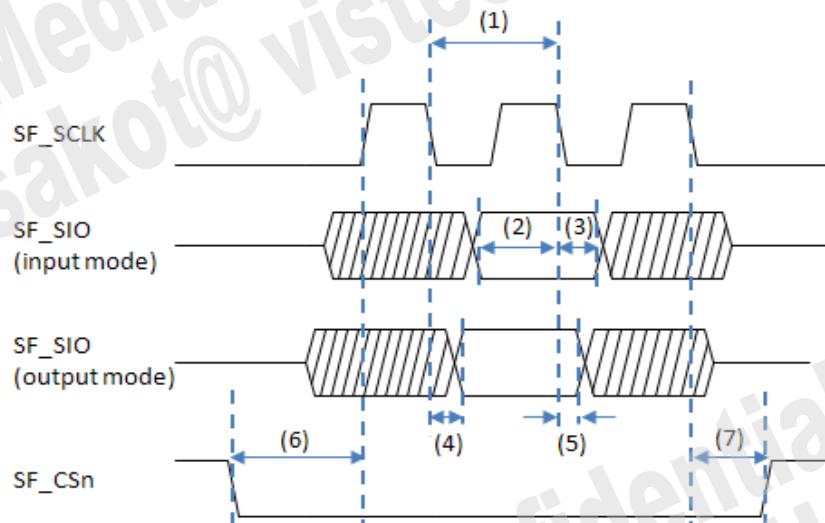


Figure 5-105 Serial NOR Flash Interface AC Timing

- Serial NOR Flash AC characteristics are depicted in Table 5-64.

Table 5-64 Serial NOR Flash Interface AC Timing Diagram Key (Tentative)

Symbol	Description	Min.	Max.	Unit
(1)	SF_SCLK frequency	-	52	MHz
	SF_SCLK duty	45	55	%
(2)	Input setup time	10	-	ns
(3)	Input hold time	0	-	ns
(4)	Output delay	0.3	0.5	ns
(5)	Output hold time	0.3	-	ns
(6)	CS low to SCLK rising edge(read)	1.5/SCLK	9.5/SCLK	ns
(7)	SCLK falling edge to CS high(read)	3/SCLK	10/SCLK	ns

5.13.5 Register Definition

For register details, please refer to Chapter 3.11 of “MT2712 IVI Application Processor Registers”.

5.13.6 Programming Guide

5.13.6.1 Read Serial Flash ID

This operation is for reading the serial NOR flash device ID, including the manufacturer ID of 1 Byte, followed by the device ID of 2 Bytes. The operation sequence is shown in Figure 5-106.

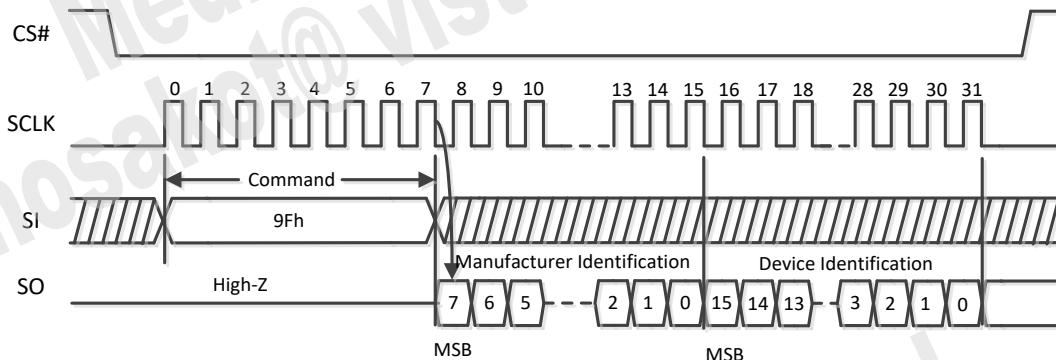


Figure 5-106 RDID Operation Sequence

The controller programming flow is as follows:

1. Write operation command (0x9F) to REG_SF_PRGDATA5 (SF_Base+0x34) and dummy data (0x00) to REG_SF_PRGDATA4 (SF_Base+0x30), REG_SF_PRGDATA3 (SF_Base+0x2C) and REG_SF_PRGDATA2 (SF_Base+0x28).
2. Write command cycles (0x20) to REG_SF_CNT (SF_Base+0x04).
3. Set REG_SF_CMD (SF_Base+0x00) [2] 1'b1 to trigger.
4. Wait for REG_SF_CMD (SF_Base+0x00) [2] to reset to 1'b0; Flash ID was saved in REG_SF_SHREG0 (SF_Base+0x38), REG_SF_SHREG1 (SF_Base+0x3C) and REG_SF_SHREG2 (SF_Base+0x40).

5.13.6.2 Erase Serial Flash

The sequence is shown in Figure 5-107.

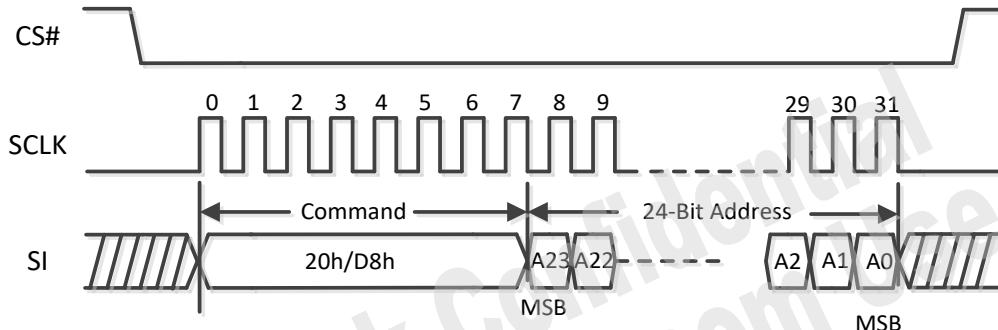


Figure 5-107 SE and BE Operation Sequence

The controller programming flow for SE and BE is as below:

1. Write operation command (0x20 or 0xD8) to REG_SF_PRGDATA5 (SF_Base+0x34).
2. Write erase address to REG_SF_PRGDATA4 (SF_Base+0x30) REG_SF_PRGDATA3 (SF_Base+0x2C) and REG_SF_PRGDATA2 (SF_Base+0x28). In the 4-Byte address mode, the sequence also includes writing the 4th Byte of address to REG_SF_PRGDATA1 (SF_Base+0x24).
3. Write command cycles (0x20 or 0x28) to REG_SF_CNT (SF_Base+0x04).
4. Set REG_SF_CMD (SF_Base+0x00) [2] 1'b1 to trigger.
5. Wait for REG_SF_CMD (SF_Base+0x00) [2] to reset to 1'b0.
6. Write REG_SF_CMD (SF_Base+0x00) [1] 1'b1.
7. Read REG_SF_RDSR (SF_Base+0x08) [0] to check that erase process is done.

The CE operation sequence is shown in Figure 5-108.

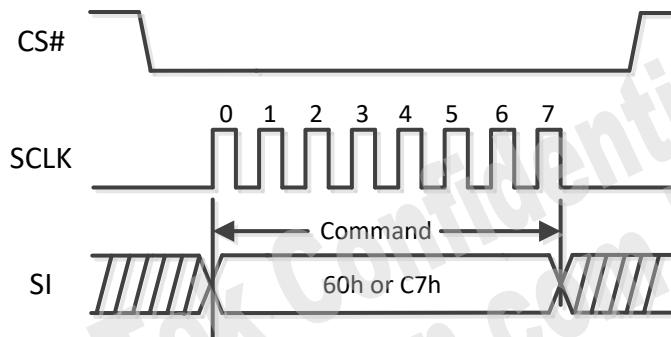


Figure 5-108 CE Operation Sequence

The controller programming flow for CE is shown as below:

1. Write operation command (0x60 or 0xC7) to REG_SF_PRGDATA5 (SF_Base+0x34).
2. Write command cycles (0x08) to REG_SF_CNT (SF_Base+0x04).
3. Set REG_SF_CMD (SF_Base+0x00) [2] 1'b1 to trigger.
4. Wait for REG_SF_CMD (SF_Base+0x00) [2] to reset to 1'b0.
5. Write REG_SF_CMD (SF_Base+0x00) [1] 1'b1.
6. Read REG_SF_RDSR (SF_Base+0x08) [0] to check that erase process is done.

5.13.6.3 Program Serial Flash

The program command is used for programming the memory to be "0". A WREN (Write Enable) command must execute the WEL (Write Enable Latch) bit before the program process. The max. program size of program command one time is a whole page size (256 Bytes for most devices). The program operation sequence is shown in Figure 5-109.

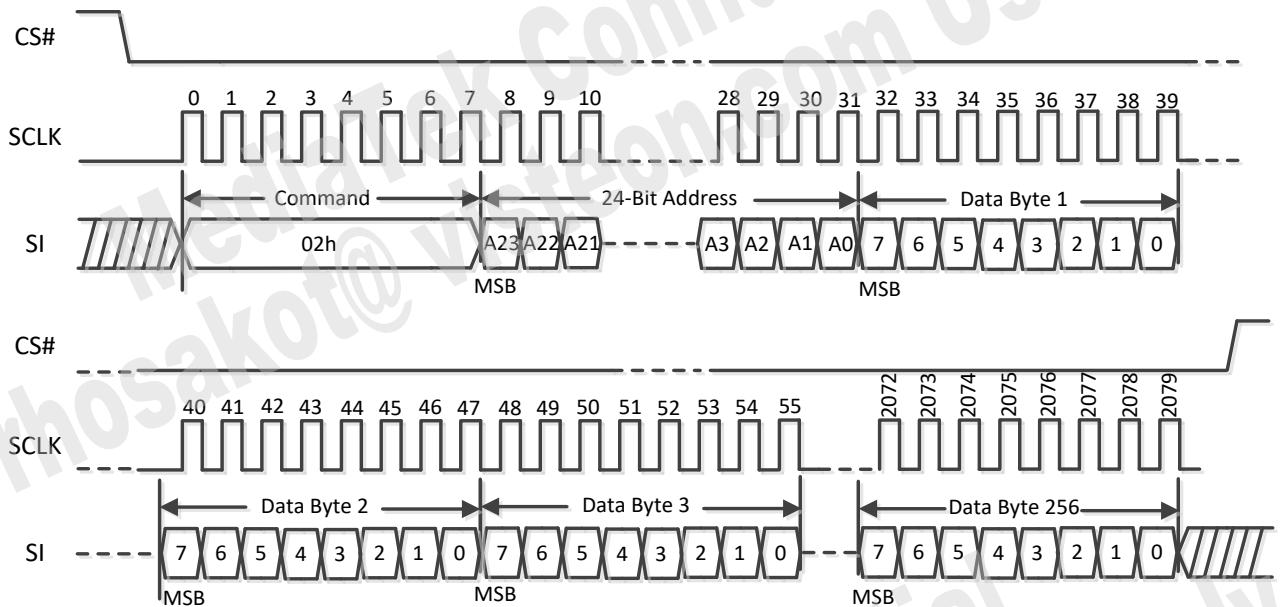


Figure 5-109 Program Operation Sequence

5.13.6.4 Write Data to Serial Flash 1-Byte One Time (PIO Write Mode)

1. Write program data to REG_SF_WDATA (SF_Base+0x1C).
2. Write REG_SF_RADR2 (SF_Base+0x18), REG_SF_RADR1 (SF_Base+0x14) and REG_SF_RADRO (SF_Base+0x10) to set the program address.
3. Write REG_SF_CMD (SF_Base+0x00) [4] 1'b1 to trigger page program.
4. Write REG_SF_CMD (SF_Base+0x00) [1] 1'b1 to send read flash status command.
5. Read REG_SF_RDSR (SF_Base+0x08) [0] to check that page program is done.

5.13.6.5 Read Flash

Standard-read (SPI) operation sequence is shown in Figure 5-110.

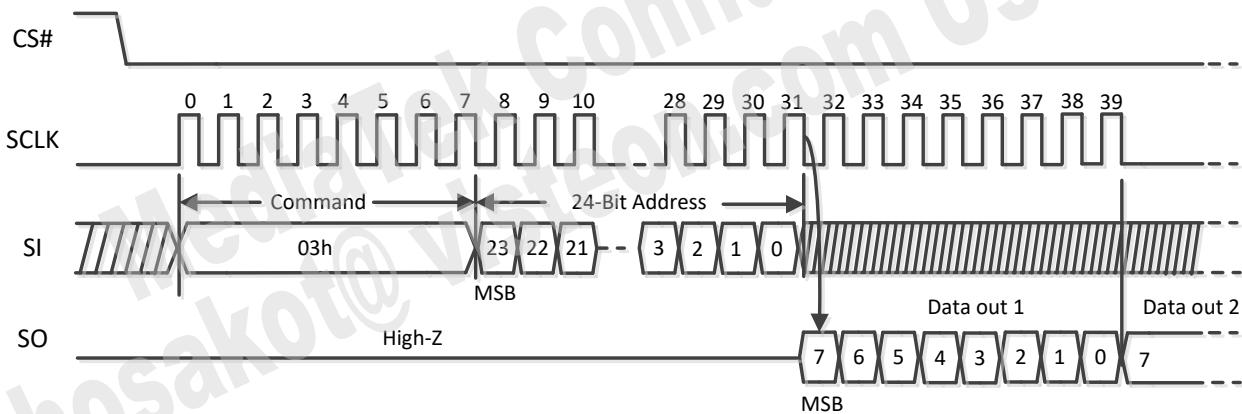


Figure 5-110 Read Operation Sequence

In the 4-Byte address mode, the address cycles will be increased from 24-bit to 32-bit. Quad I/O and quad output read are also supported. There will be 4 data pins in this case.

5.13.6.6 PIO Read Mode

The controller reads the register REG_SF_RDATA (SF_Base+0x0C) to get the NOR flash data. Program flow is shown as below:

1. Write REG_SF_RADR2(SF_Base+0x18), REG_SF_RADR1 (SF_Base+0x14) and REG_SF_RADR0 (SF_Base+0x10) to set the start read address; in the 4-Byte (32 bits) address mode, need write addr [31:24] to REG_SF_RADR3(SF_Base+0xC8).
2. Write REG_SF_CMD (SF_Base+0x00) [7] [0] 1'b1 to trigger read process.
3. Wait REG_SF_CMD (SF_Base+0x00) bit0 to be reset to 0.
4. Read REG_SF_RDATA (SF_Base+0x0C) to get the NOR flash data.

5.13.6.7 Direct Read Mode

Direct Read means that data of NOR flash can be directly read by CPU through address offset. For instance, users would like to get the address of 0x100 data in NOR flash. Users can capture data directly by reading the contents of address offset 0x100 via the direct read mode.

5.13.6.8 Enter 4-Bit Read Mode

The controller supports the 4-bit SPI read mode to enhance the read performance. This includes the following two read sequences. The difference is whether the address is sent in the 4-bit mode, as shown in Figure 5-111 & Figure 5-112.

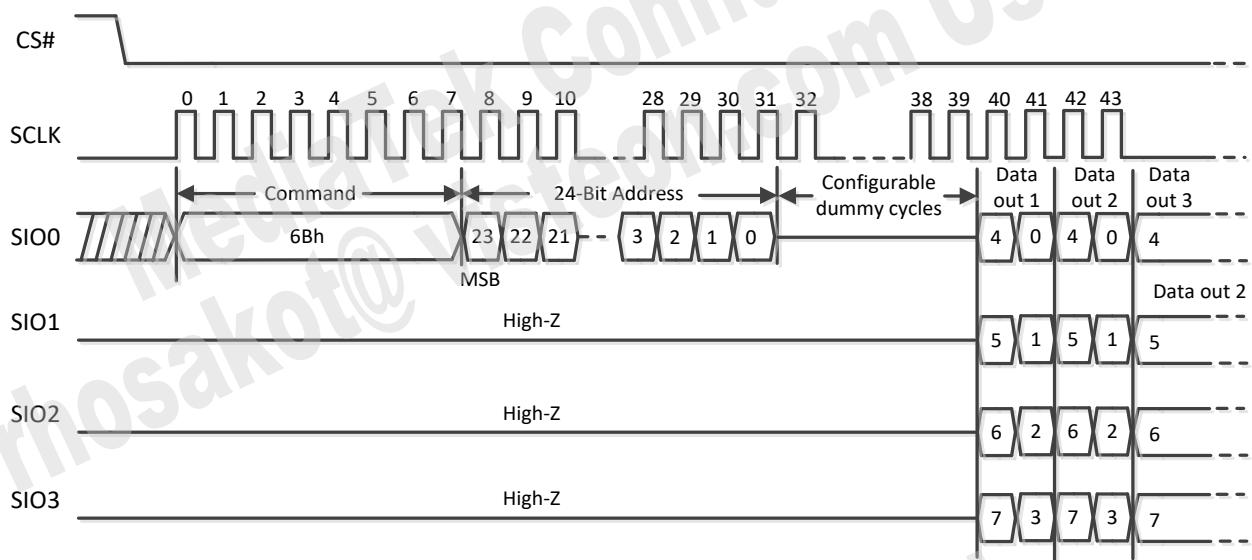


Figure 5-111 Quad Output Read Mode Sequence (Address is sent in single-bit mode)

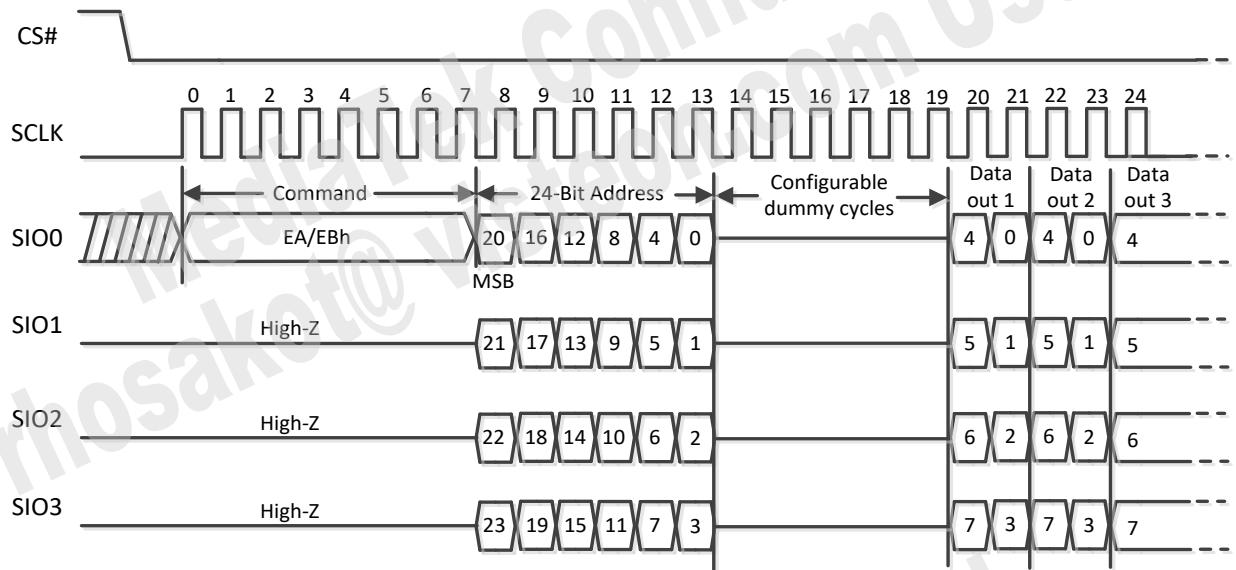


Figure 5-112 Quad I/O Read Mode Sequence (Address is sent in 4-bit mode)

Enter the 4-bit read mode, then the read operation will be in the 4-bit read mode.

The program flow is shown as below:

1. Read NOR flash status register to check whether the QE (Quad Enable) bit is enabled or not and back up status register bit field.

Note: For QE enable bit's location difference among flash vendors, please refer to their respective flash datasheet.

2. Write NOR flash status register QE 1'b1 to enable SIO2 and SIO3.
3. Set REG_SF_DUAL (SF_Base+0xCC) [3] (Add_Quad) and [2] (Quad_Read_En) to enable serial flash controller's 4-bit mode.

5.13.6.9 Exit 4-Bit Read Mode

Please follow the steps below to exit the 4-bit mode:

1. Set REG_SF_DUAL (SF_Base+0xCC) [3] (Add_Quad) 1b'0 and [2] (Quad_Read_En) 1b'0.
2. Write NOR flash status register QE (Quad Enable) 1'b0 to disable SIO2 and SIO3.

5.13.6.10 Enter 4-Byte Address Mode

For most of the NOR flash devices, the default address mode is 3-Byte address mode.

To access the space larger than 16 MB, please follow the steps below:

1. Send operation command (0xB7) to make NOR flash enter the 4-Byte address mode.
2. Set REG_SF_DUAL (SF_Base+0xCC) [4] (Large_Addr_En).
3. Wait for REG_SF_DUAL (SF_Base+0xCC) [4] to be 1.

5.13.6.11 Exit 4-Byte Address Mode

Please follow the steps below to exit the 4-Byte address mode:

1. Reset REG_SF_DUAL (SF_Base+0xCC) [4] (Large_Addr_En).
2. Wait for REG_SF_DUAL (SF_Base+0xCC) [4] to be 0.
3. Send operation command (0xE9) to make NOR flash exit the 4-Byte address mode.

5.14 AUXADC

5.14.1 Introduction

The Auxiliary Analog/Digital Converter (AUXADC) module is used to identify the plugged peripheral and perform temperature/voltage measurement. There are 16 input channels that allow diverse applications, such as temperature/voltage measurement and light sensing.

5.14.2 Features

The module contains:

- Immediate analog-digital conversion
- Background detection and interrupt

5.14.2.1 Immediate Mode

In the immediate mode, AUXADC samples the value once only when the flag in the AUXADC_CON1 register is set. For example, if the flag IMM0 in AUXADC_CON1 is set, the module will sample the data for channel 0. The IMM flags have to be cleared and set again to initialize another sampling. The value sampled for channel 0 is stored in the register AUXADC_DAT0. If the AUTOSET(x) flag in the register AUXADC_CON0 is set, the auto-sampling function will be enabled in channel(x). The module will sample the data for channel(x) whenever the corresponding data register is read.

If multiple channels are selected at the same time, the task will be performed sequentially on every selected channel from channel 15 to channel 0.

5.14.2.2 Background Detection

If background detection is enabled, AUXADC will automatically compare the selected channel data with the user-defined value. If the results are continuously greater/less than the given value, AUXADC will issue an interrupt to inform system user.

5.14.3 Block Diagram

Software (SW) controls AUXADC through the APB bus. Once the hardware receives the command, it will trigger AUXADC's channel sampling automatically. SW polls the status register or waits for interrupts from the CPU.

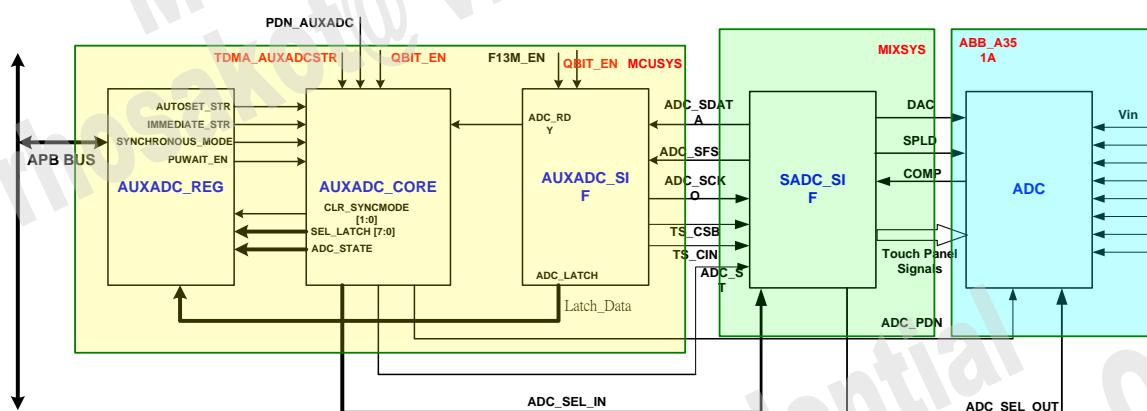


Figure 5-113 AXUADC Block Diagram

5.14.4 AUXADC AC Timing

5.14.4.1 Introduction

The auxiliary ADC includes the following functional blocks:

1. Analog multiplexer: Selects signal from one of the auxiliary input channels. There are 16- input channels of AUXADC. Some are for an internal voltage measure and some for an external voltage measure. Environmental messages to be monitored, e.g. temperature, while it should be transferred to the voltage domain.
2. 12-bit A/D converter: Converts the multiplexed input signal to 12-bit digital data.

5.14.4.1.1 Features

Refer to Table 5-65 for brief descriptions of AUXADC input channels.

Table 5-65 Definitions of AUXADC Channels

AUXADC Channel ID	Description
Channel 0	External use (AUXIN0)
Channel 1	External use (AUXIN1)
Channel 2~9	Internal use
Channel 10~11	Thermal sensor input
Channel 12	External use (AUXIN2)
Channel 13~15	Internal use

5.14.4.1.2 Block Diagram

Figure 5-114 is the block diagram of AUXADC in MT2712.

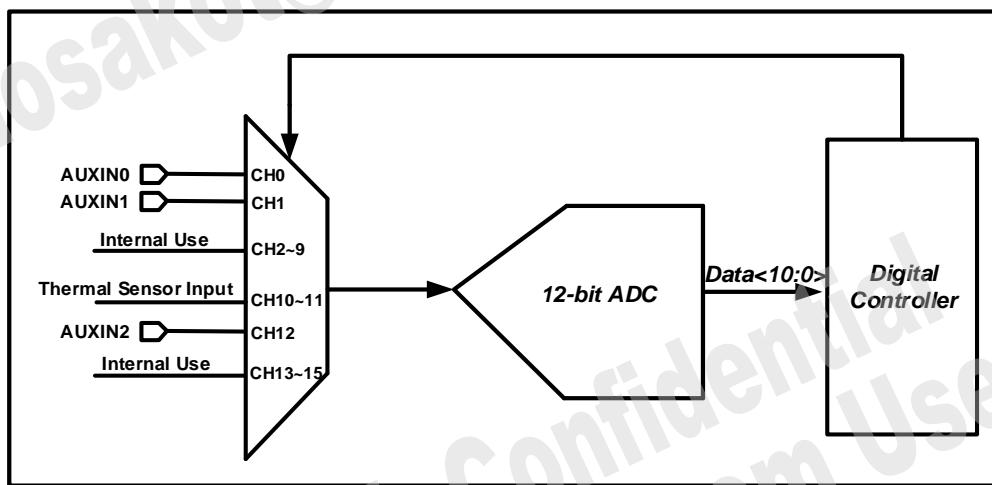


Figure 5-114 AUXADC Block Diagram

5.14.4.1.3 Functional Specifications

Refer to Table 5-66 for the functional specifications of auxiliary ADC.

Table 5-66 AUXADC Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
N	Resolution	-	-	12	Bit	-
F _C	Clock rate	-	-	4	MHz	-
F _S	Sampling rate@N-Bit	-	-	250	ksps	-
V _{IN}	Input swing	0	-	1.5	V	-
DNL	Differential nonlinearity	-	+1.0/-1.0	-	LSB	Sim data
INL	Integral nonlinearity	-	+2.0/-2.0	-	LSB	Sim data
SINAD	Signal to noise and distortion ratio (1 kHz full swing input)	56	62	-	dB	-
C _{IN}	Input capacitance selected channel	-	4.32	-	pF	Sim data
	Input capacitance unselected channel	-	0.05	-	pF	Sim data
R _{IN}	Input resistance selected channel	-	360	-	Ω	Sim data
	Input resistance unselected channel	-	425	-	MΩ	Sim data
I	Current consumption Power-up Power-down	-	480 1	650	- μA μA	-

5.14.4.2 Temperature Sensor

5.14.4.2.1 Introduction

In order to monitor the temperature of CPUs several temperature sensors are provided. These temperature sensors are made of substrate like in BJT & CMOS process. The voltage output of temperature sensor is measured by AUXADC.

5.14.4.2.2 Functional Specifications

Refer to the Table 5-67 for the functional specifications of temperature sensor.

Table 5-67 Temperature Sensor Specifications

Parameter	Min.	Typ.	Max.	Unit
Temperature range	-40		125	°C
Thermal sensor Accuracy @CP calibration	121.5	125	128.5	°C

5.14.5 Theory of Operations

Successive-approximation-register (SAR) ADC provides low power consumption, cost-effective and medium resolution. The AUXADC module has the SAR ADC architecture.

Following is an example of 12-bit conversion. V_{REF} is the reference voltage of AUXADC.

AUXADC implements a binary search algorithm. An initial register VDA value, the mid-value between ($2^{12}-1$) and 0, is compared to the input voltage V_{IN} . The value represents $V_{REF}/2$. If V_{IN} is bigger than V_{DA} , the output of comparison will be 1, and the MSB bit will be 1. Otherwise, the MSB bit will be 0. Subsequently, bit 11 will be set to 1, and another comparison will be done. Bit 10 to bit 0 will be executed as the previous action. Then, the 12-bit digital value will be available.

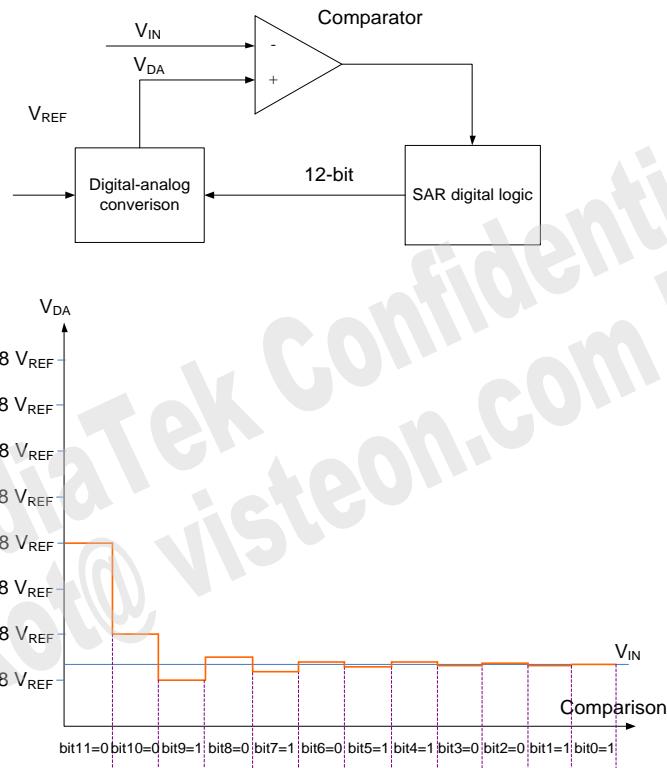


Figure 5-115 Theory of AUXADC Operation

5.14.6 Register Definition

For register details, please refer to Chapter 3.12 of “MT2712 IVI Application Processor Registers”.

5.14.7 Programming Guide

5.14.7.1 Immediate Mode

Table 5-68 Immediate Mode Programming Sequence

Immediate Mode						
Description	R/W	Address	Bit	MACRO	Value	Note
Open AUXADC Clock	W	PERICFG_BASE + 0x0010	[28]	AUXADC_PDN	1'b1	Set 1 to enable AUXADC clock
Set Immediate Mode	W	AUXADC_BASE + 0x0004	[15:0]	AUXADC_CON1	USER_DEFINED	Set 1 to sample corresponding channel once
Polling Ready	R	AUXADC_BASE + 0x0014 + n*4	[12]	RDYn	1'b1	Sample data is ready when this bit changes to 1.
Read Result	R	AUXADC_BASE + 0x0014 + n*4	[11:0]	DATn	-	Sample result

5.14.7.2 Background Detection

Table 5-69 Background Detection Programming Sequence

Background Detection						
Description	R/W	Address	Bit	MACRO	Value	Note
Set Threshold Voltage	W	AUXADC_BASE + 0x0084	[11:0]	VOL	USER_DEFINED	Set Threshold Voltage
Set Compare direction	W	AUXADC_BASE + 0x0084	[12]	INV	USER_DEFINED	0 : Lower 1 : Higher
Set Detection Channel	W	AUXADC_BASE + 0x0088	[3:0]	CHSEL	USER_DEFINED	Set channel to be sampled in background
Set Detection Period	W	AUXADC_BASE + 0x008C	[13:0]	BG_DET_PERIOD	USER_DEFINED	Background sample period: When this value is not 0, the background detection will be activated automatically and other ADC sampling functions will be stopped. The counter counts by 32K clock. When counter value is greater than DET_PERIOD, the detection will be activated.
Set Detection Debounce	W	AUXADC_BASE + 0x0090	[13:0]	BG_DET_TIME	USER_DEFINED	Background de-bounce time: When the number of the detected channel is higher or lower than the pre-defined voltage and exceeds "debounce_time", the interrupt will be issued.

5.15 I2C/SCCB Controller

5.15.1 Introduction

Inter-IC (I2C)/Serial Camera Control Bus (SCCB) controller is a two-wire serial interface. The two signals are Serial Clock Line (SCL) and Serial Data Line (SDA). SCL is a clock signal that is driven by the master. SDA is a bi-

directional data signal that can be driven by either the master or the slave. This generic controller supports the master role and conforms to the I2C specifications.

5.15.2 Features

The I2C/SCCB controller supports the following features:

- I2C compliant master mode operation
- Adjustable clock speed for LS (Standard mode)/FS (Fast mode)/FS+ (Fast mode Plus) mode operation
- Support 7-bit or 10-bit addressing
- Support high-speed mode
- Support slave clock extension
- START/STOP/REPEATED START conditions
- Manual transfer mode
- Multi-write per transfer
- Multi-read per transfer
- Multi-transfer per transaction
- Combined format transfer with length change capability
- Active drive/wired-and I/O configuration

Limitation: For High Speed mode, IO voltage should be 1.8 V and AC timing specification is not guaranteed.

5.15.3 Block Diagram

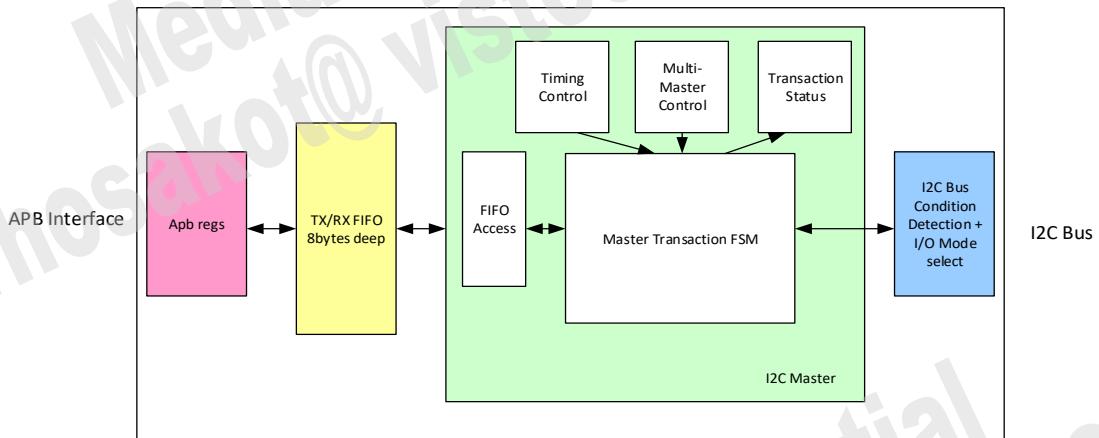


Figure 5-116 I2C Block Diagram

5.15.4 I2C AC Timing

5.15.4.1 I2C Timing Characteristics

I2C Controller supports Standard mode (ST(I2C)) (100 kHz), Fast-mode (FS(I2C)) (400 kHz) and Fast-mode Plus (Fm+(I2C)) (1 MHz). Fast-mode Plus(Fm+(I2C)) devices offer an increase in I2C-bus transfer speeds and the total bus capacitance. Fm+(I2C) devices can transfer information at a bit-rate of up to 1-MHz, yet they remain fully downward compatible with fast or standard-mode devices for bi-directional communication in a mixed-speed bus system. The same serial bus protocol and data format is maintained with the fast or standard-mode system. Figure 5-117 describes the definition of timing for the F/S mode devices on the I2C-Bus.

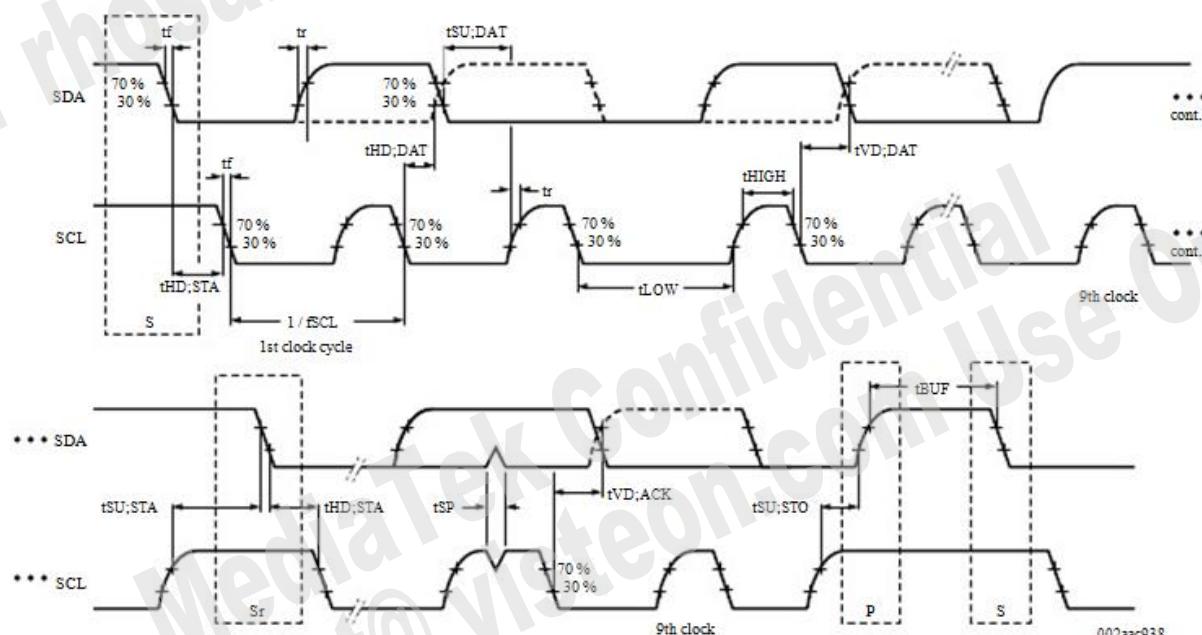


Figure 5-117 Definition of Timing for the F/S Mode Devices on the I2C-Bus

5.15.4.2 I2C Electrical Characteristics

I2C electrical characteristic specifications within current chip refer to the Table 5-70. The I2C electrical characteristic specifications followed by the I2C standard specifications refer to the Table 5-71 as shown below.

Table 5-70 I2C Electrical Characteristic Specifications (Current Chip)

Symbol	Parameter	Standard-mode		Fast-mode		Fast-mode Plus		Unit	Note
		Min.	Max.	Min.	Max.	Min.	Max.		
fSCL	SCL clock frequency	0	100	0	400	0	1000	kHz	-
tHD;STA	Hold time(repeated)STAR T condition	1080	-	720	-	300	-	ns	reg adjustable
tLOW	LOW period of the SCL clock	2740	-	960	-	420	-	ns	reg adjustable

Symbol	Parameter	Standard-mode		Fast-mode		Fast-mode Plus		Unit	Note
		Min.	Max.	Min.	Max.	Min.	Max.		
tHIGH	HIGH period of the SCL clock	2740	-	960	-	420	-	ns	reg adjustable
tSU;STA	Set-up time for a repeated START condition	1780	-	720	-	300	-	ns	reg adjustable
tHD;DAT	Data hold time	1780	-	720	-	300	-	ns	output,reg adjustable
tHD;DAT	Data hold time	0	-	0	-	0	-	ns	input requirement
tSU;DAT	Data set-up time	2480	-	720	-	300	-	ns	output,reg adjustable
tSU;DAT	Data set-up time	0	-	0	-	0	-	ns	input requirement
tr	Rise time of both SDA and SCL signals	-	1000	20	300	-	120	ns	-
tf	Fall time of both SDA and SCL signals	-	300	20x (VDD/5.5V)	300	20x (VDD/5.5V)	120	ns	VDD is IO voltage
tSU;STO	Set-up time for STOP condition	2480	-	720	-	300	-	ns	-
tVD;DAT	Data valid time	2780	-	1020	-	420	-	ns	output,reg adjustable
tVD;DAT	Data valid time	0	-	0	-	0	-	ns	input requirement
tVD;ACK	Data valid acknowledge time	2780	-	1020	-	420	-	ns	output,reg adjustable
tVD;ACK	Data valid acknowledge time	0	-	0	-	0	-	ns	input requirement
C _b	Capacitive load of each bus line	-	100	-	100	-	100	pF	-

Note: Tr/Tf is up to RC parameter and current AC parameter uses max tr/tf value of standard spec as reference.

When using different RC parameter, the AC timing will have corresponding change.

Table 5-71 I2C Electronic Characteristic Specifications (Standard Protocol)

Symbol	Parameter	Standard-mode		Fast-mode		Fast-mode Plus		Unit	Note
		Min.	Max.	Min.	Max.	Min.	Max.		
fSCL	SCL clock frequency	0	100	0	400	0	1000	kHz	-
tHD;STA	Hold time(repeated)START condition	4.0	-	0.6	-	0.26	-	us	-
tLOW	LOW period of the SCL clock	4.7	-	1.3	-	0.5	-	us	-
tHIGH	HIGH period of the SCL clock	4.0	-	0.6	-	0.26	-	us	-
tSU;STA	Set-up time for a repeated START condition	4.7	-	0.6	-	0.26	-	us	-

Symbol	Parameter	Standard-mode		Fast-mode		Fast-mode Plus		Unit	Note
		Min.	Max.	Min.	Max.	Min.	Max.		
tHD;DAT	Data hold time	0	-	0	-	0	-	us	I2C-bus devices
tSU;DAT	Data set-up time	250	-	100	-	50	-	ns	-
tr	Rise time of both SDA and SCL signals	-	1000	20	300	-	120	ns	-
tf	Fall time of both SDA and SCL signals	-	300	20x (VDD/5.5V)	300	20x (VDD/5.5V)	120	ns	VDD is IO voltage
tSU;STO	Set-up time for STOP condition	4.0	-	0.6	-	0.26	-	ns	-
tVD;DAT	Data valid time	-	3.45	-	0.9	-	0.45	us	-
tVD;ACK	Data valid acknowledge time	-	3.45	-	0.9	-	0.45	us	-
C _b	Capacitive load of each bus line	-	400	-	400	-	550	pF	-

5.15.5 Manual Transfer Mode

The controller offers the manual mode. When the manual mode is selected, in addition to the slave address register, the controller has a built-in 8-byte deep First In First Out (FIFO) which allows Microcontroller Unit (MCU) to prepare up to 8 bytes of data for a write transfer, or read up to 8 bytes of data for a read transfer.

5.15.6 Transfer Format Support

This controller is designed to be as generic as possible in order to support a wide range of devices that may utilize different combinations of transfer formats. Following are the transfer format types supported through different software configurations:

Wording convention note

- Transfer = Anything encapsulated within a Start and Stop or Repeated Start.
- Transfer length = Number of bytes within the transfer
- Transaction = This is the top unit. Everything combined equals 1 transaction.
- Transaction length = Number of transfers to be conducted.



Master to slave dir



Slave to master dir

Single-byte access

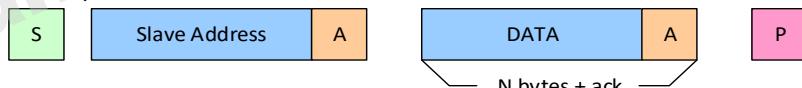
Single Byte Write



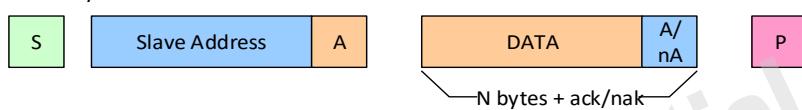
Single Byte Read

**Multi-byte access**

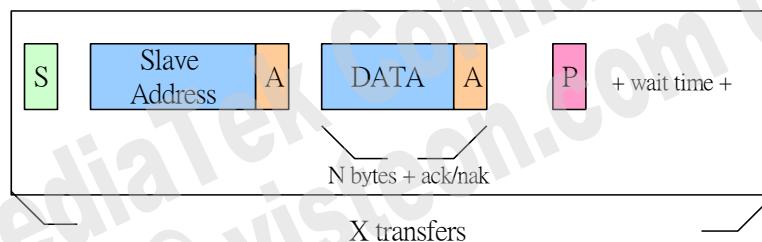
Multi Byte Write



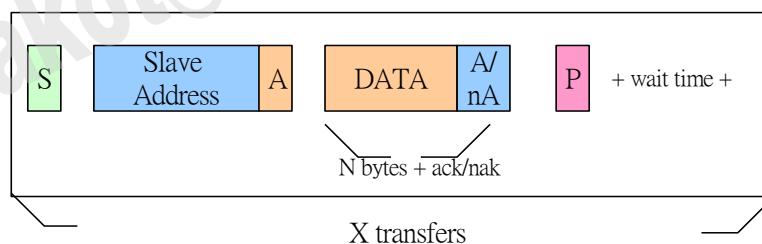
Multi Byte Read

**Multi-byte transfer + multi-transfer (same direction)**

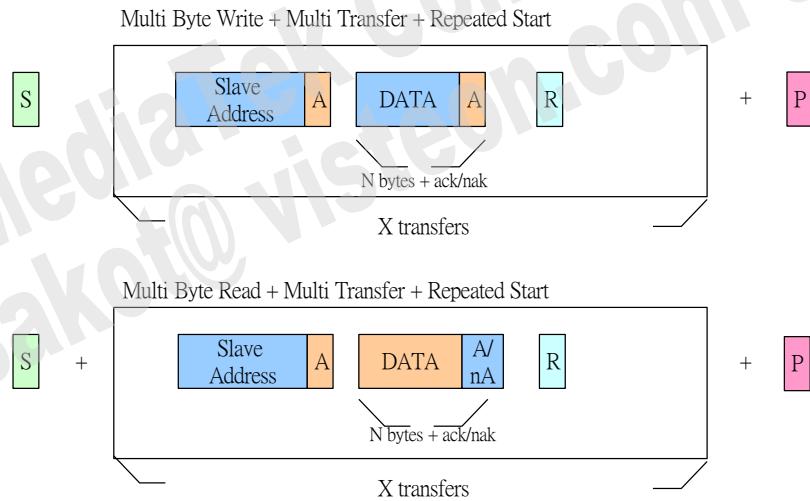
Multi Byte Write + Multi Transfer



Multi Byte Read + Multi Transfer



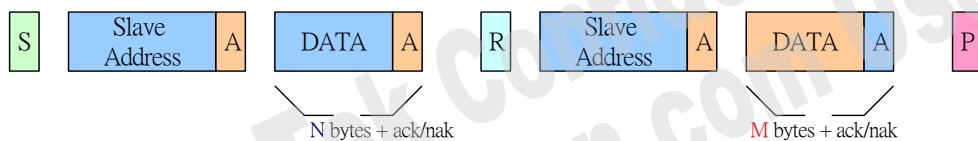
Multi-byte transfer + multi-transfer with Repeated Start (same direction)



Combined write/read with Repeated Start (direction change)

Note: Only support the write before read sequence. Read before write sequence is not supported.

Combined Multi Byte Write + Multi Byte Read



5.15.7 Register Definition

For register details, please refer to Chapter 3.13 of “MT2712 IVI Application Processor Registers”.

5.15.8 Programming Guide

Steps to conduct an I2C transmission:

1. Set the timing and protocol to conduct the I2C transmission. Detailed setup parameters can be found in Figure 5-118 and Figure 5-119.
2. Set the register start to start the transaction
3. Fill in and take out the data of FIFOs according to the FIFO status
4. After the transmission is done, the interrupt will be asserted if the interrupt is unmasked.

Note: The bus clock (axi clock) is used for I2C's base_clock.

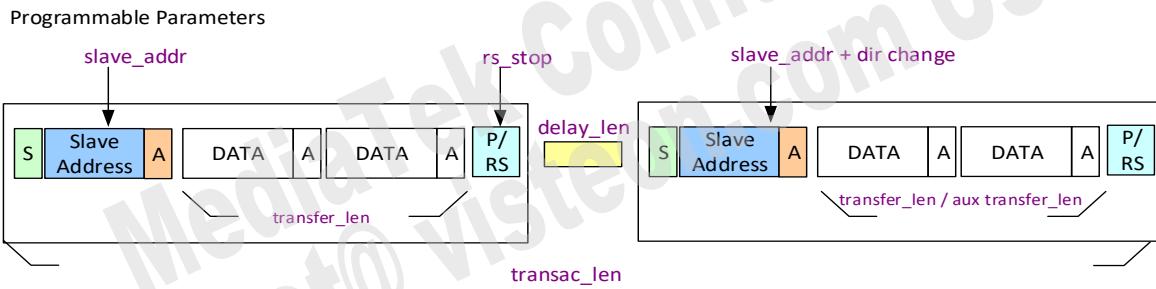


Figure 5-118 Common Transfer Programmable Parameters

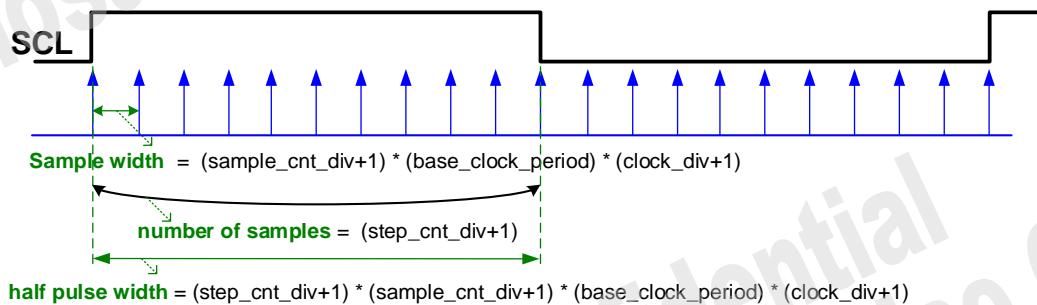


Figure 5-119 Output Waveform Timing Programmable Parameters

5.16 Pulse-Width Modulation

5.16.1 Introduction

Eight generic pulse-width modulators are implemented to generate pulse sequences with programmable frequency and duration for LCD backlight, charging or other purposes. Before enabling Pulse Width Modulation (PWM), the pulse sequences must be prepared in the memory or registers. Then PWM will read the pulse sequences to generate random waveform to meet all kinds of applications (see Figure 5-120).

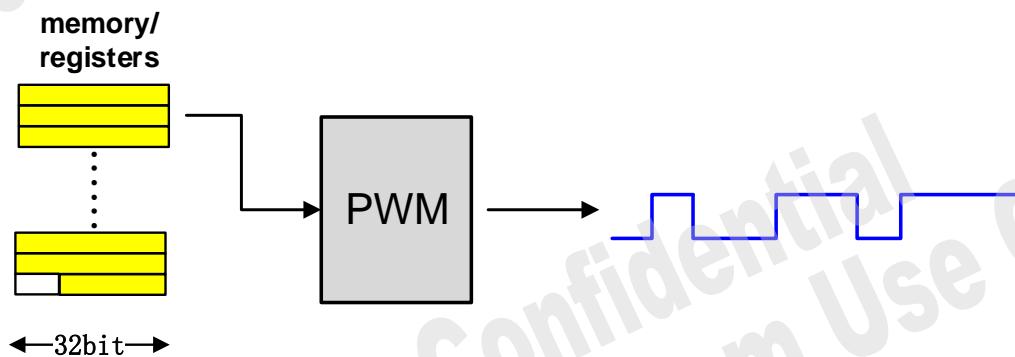


Figure 5-120 PWM Generation Procedure

5.16.2 Specification

- MT2712 has eight dedicated PWM outputs for other function applications.
- PWM IO level depends on its IO Power Domain (1.8 V or 3.3 V)
- PWM output frequency range (0.04 Hz~78 MHz)
- PWM output duty (0%~100%)

5.16.3 Features

- Old mode
- FIFO mode
- Periodical memory
- Random mode
- Sequential output mode
- 3DLCM mode

5.16.4 Block Diagram

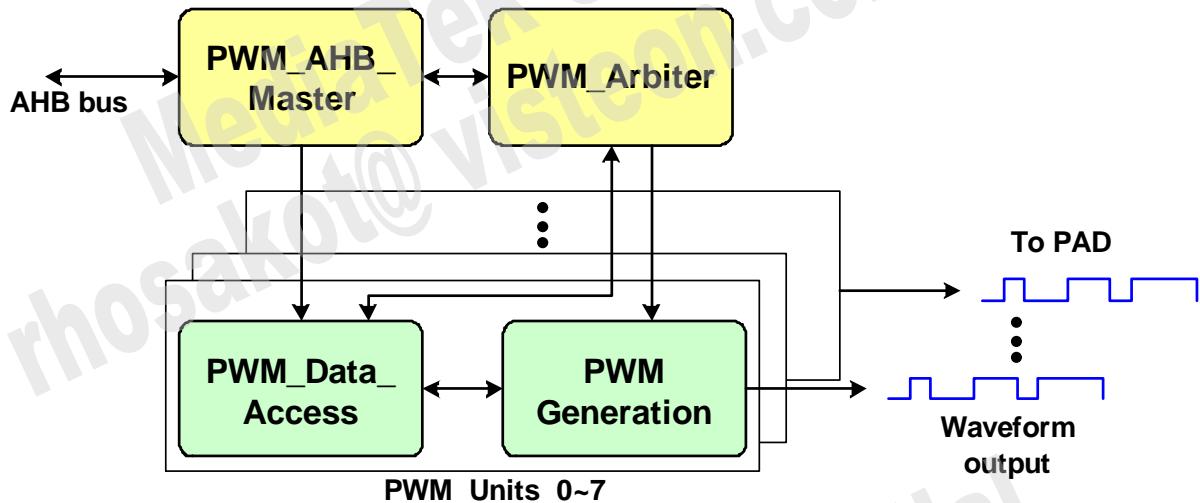


Figure 5-121 PWM Block Diagram

5.16.5 PWM AC Timing

5.16.5.1 Timing Characteristics of PWM Parameter

This section describes the timing parameters of the PWM module. Table 5-72 depicts the characteristics of the PWM out timing lines, and Figure 5-122 illustrates the PWM module timing characteristics.

Table 5-72 Characteristics of the PWM Timing

Parameter	Description	Min.	Max.	Unit
T_{HIGH}	High time of PWM_n_out	6.4	-	ns
T_{LOW}	Low time of PWM_n_out	6.4	-	ns

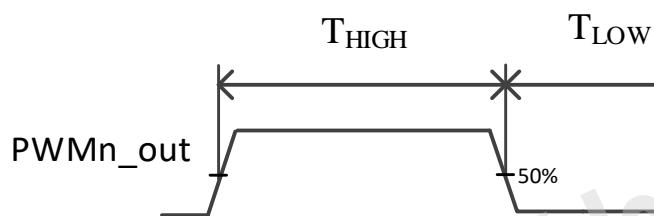
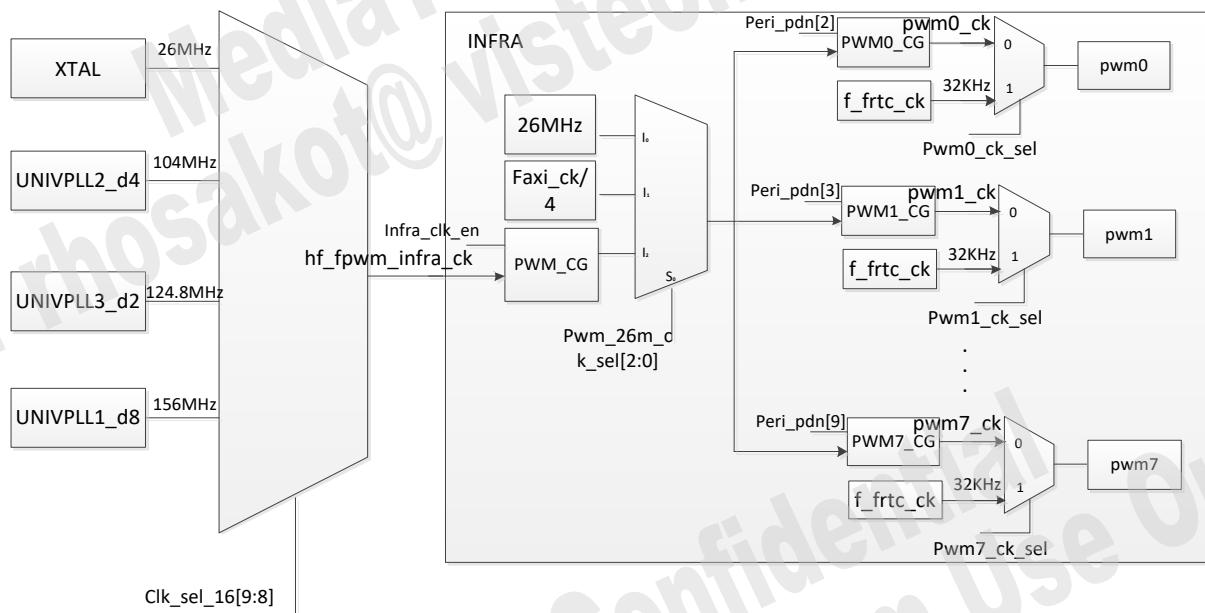


Figure 5-122 PWM Timing

5.16.6 Clock Relationship



MT2712 PWM module clock

Figure 5-123 PWM Module Clock Structure

5.16.7 Register Definition

For register details, please refer to Chapter 3.14 of “MT2712 IVI Application Processor Registers”.

5.16.8 Programming Guide

5.16.8.1 Old Mode

Table 5-73 Old Mode Setting Procedures

PWM Setting Sequence for Old Mode							
Step	Description	R/W	Address	Bit	MACRO	Value	Note
1	Set PWM_ENABLE	W	PWM_BASE + 0x0000	[N]	PWM_ENA BLE	1'b0	Disable PWM[N]
2	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[15]	OLD_PWM _MODE	1'b1	-
3	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[8]	GUARD_VA LUE	USER_DEFINED	-
4	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[7]	IDLE_VALU E	USER_DEFINED	-
5	Set PWM_WAVE_NUM	W	PWM_BASE + 0x038+ PWM_NUM*0x40	[15:0]	PWM_WA VE_NUM	USER_DEFINED	If WAVE_NUM=0, the waveform generation will not stop until it is disabled.
6	Set PWM_GDURATION	W	PWM_BASE + 0x01C+ PWM_NUM*0x40	[15:0]	PWM_GDU RATION	USER_DEFINED	-
7	Set PWM_DATA_WIDTH	W	PWM_BASE + 0x3C+ PWM_NUM*0x40	[12:0]	PWM_DAT A_WIDTH	USER_DEFINED	-
8	Set PWM_THRESH	W	PWM_BASE + 0x40+ PWM_NUM*0x40	[12:0]	PWM_THR ESH	USER_DEFINED	-
9	Set PWM_ENABLE	W	PWM_BASE + 0x0000	[N]	PWM_ENA BLE	1'b1	Enable PWM[N]

5.16.8.2 FIFO Mode

Table 5-74 FIFO Mode Setting Procedures

PWM Setting Sequence for FIFO Mode							
Step	Description	R/W	Address	Bit	MACRO	Value	Note
1	Set PWM_ENABLE	W	PWM_BASE + 0x0000	[N]	PWM_ENA BLE	1'b0	Disable PWM[N]

PWM Setting Sequence for FIFO Mode							
Step	Description	R/W	Address	Bit	MACRO	Value	Note
2	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[15]	OLD_PWM_MODE	1'b0	-
3	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[6]	MODE	1'b0	-
4	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[5]	SRCSEL	1'b0	-
5	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[8]	GUARD_VALUE	USER_DEFINED	-
6	Set IDLE_VALUE	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[7]	IDLE_VALUE	USER_DEFINED	-
7	Set PWM_WAVE_NUM	W	PWM_BASE + 0x038+ PWM_NUM*0x40	[15:0]	PWM_WAVE_NUM	USER_DEFINED	If WAVE_NUM=0, the waveform generation will not stop until it is disabled.
8	Set PWM_GDURATION	W	PWM_BASE + 0x01C+ PWM_NUM*0x40	[15:0]	PWM_GDURATION	USER_DEFINED	-
9	Set PWM_HDURATION	W	PWM_BASE + 0x014+ PWM_NUM*0x40	[15:0]	PWM_HDURATION	USER_DEFINED	-
10	Set PWM_LDURATION	W	PWM_BASE + 0x018+ PWM_NUM*0x40	[15:0]	PWM_LDURATION	USER_DEFINED	-
11	Set PWM_SEND_DATA0	W	PWM_BASE + 0x030+ PWM_NUM*0x40	[31:0]	PWM_SEND_DATA0	USER_DEFINED	This value should be written only in Periodical FIFO Mode. In other modes, this buffer is for internal memory access.
12	Set PWM_SEND_DATA1	W	PWM_BASE + 0x034+ PWM_NUM*0x40	[31:0]	PWM_SEND_DATA1	USER_DEFINED	This value should be written only in Periodical FIFO Mode. In other modes, this buffer is for internal memory access.
13	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[14:9]	STOP_BITPOS	USER_DEFINED	In Periodical Mode, it is used to indicate the stop bit position for source data In FIFO Mode, it is used to indicate the stop bit position of total 64-bits.

PWM Setting Sequence for FIFO Mode							
Step	Description	R/W	Address	Bit	MACRO	Value	Note
							In Memory Mode, it is used to indicate the stop bit position of the last 32-bits.
14	Set PWM_ENABLE	W	PWM_BASE + 0x0000	[N]	PWM_ENA BLE	1'b1	Enable PWM[N]

5.16.8.3 Memory Mode

Table 5-75 Memory Mode Setting Procedures

PWM Setting Sequence for Memory Mode							
Step	Description	R/W	Address	Bit	MACRO	Value	Note
1	Set PWM_ENABLE	W	PWM_BASE + 0x0000	[N]	PWM_ENA BLE	1'b0	Disable PWM[N]
2	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[15]	OLD_PWM _MODE	1'b0	-
3	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[6]	MODE	1'b1	-
4	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[5]	SRCSEL	1'b1	-
5	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[8]	GUARD_VA LUE	USER_DEFINED	-
6	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[7]	IDLE_VALU E	USER_DEFINED	-
7	Set PWM_WAVE_N UM	W	PWM_BASE + 0x038+ PWM_NUM*0x40	[15:0]	PWM_WA VE_NUM	USER_DEFINED	If WAVE_NUM=0, the waveform generation will not stop until it is disabled.
8	Set PWM_GDURATI ON	W	PWM_BASE + 0x01C+ PWM_NUM*0x40	[15:0]	PWM_GDU RATION	USER_DEFINED	-
9	Set PWM_HDURATI ON	W	PWM_BASE + 0x014+ PWM_NUM*0x40	[15:0]	PWM_HDU RATION	USER_DEFINED	-
10	Set PWM_LDURATI ON	W	PWM_BASE + 0x018+ PWM_NUM*0x40	[15:0]	PWM_LDU RATION	USER_DEFINED	-

PWM Setting Sequence for Memory Mode							
Step	Description	R/W	Address	Bit	MACRO	Value	Note
11	Set PWM_BUFO_BA SE_ADDR	W	PWM_BASE + 0x020+ PWM_NUM*0x40	[31:0]	PWM_BUF O_BASE_A DDR	USER_DEFINED	Base address of memory buffer0 for PWM's waveform data.
12	Set PWM_BUFO_SIZ E	W	PWM_BASE + 0x024+ PWM_NUM*0x40	[2:0]	PWM_BUF O_BASE_A DDR_EXTE ND	USER_DEFINED	It indicates the length of waveform data in memory buffer0. PWM should generate it.
13	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[14:9]	STOP_BITP OS	USER_DEFINED	In Periodical Mode, it is used to indicate the stop bit position for source data. In FIFO Mode, it is used to indicate the stop bit position of total 64-bits. In Memory Mode, it is used to indicate the stop bit position of the last 32-bits.
14	Set PWM_ENABLE	W	PWM_BASE + 0x0000	[N]	PWM_ENA BLE	1'b1	Enable PWM[N]

5.16.8.4 Random Mode

Table 5-76 Random Mode Setting Procedures

PWM Setting Sequence for Random Mode							
Step	Description	R/W	Address	Bit	MACRO	Value	Note
1	Set PWM_ENABLE	W	PWM_BASE + 0x0000	[N]	PWM_ENA BLE	1'b0	Disable PWM[N]
2	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[15]	OLD_PWM _MODE	1'b0	-
3	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[6]	MODE	1'b0	-
4	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[5]	SRCSEL	1'b1	-
5	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[8]	GUARD_VA LUE	USER_DEFINED	-
6	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[7]	IDLE_VALU E	USER_DEFINED	-

PWM Setting Sequence for Random Mode							
Step	Description	R/W	Address	Bit	MACRO	Value	Note
7	Set PWM_GDURATI ON	W	PWM_BASE + 0x01C+ PWM_NUM*0x40	[15:0]	PWM_GDU RATION	USER_DEFINED	-
8	Set PWM_HDURATI ON	W	PWM_BASE + 0x014+ PWM_NUM*0x40	[15:0]	PWM_HDU RATION	USER_DEFINED	-
9	Set PWM_LDURATI ON	W	PWM_BASE + 0x018+ PWM_NUM*0x40	[15:0]	PWM_LDU RATION	USER_DEFINED	-
10	Set PWM_BUFO_BA SE_ADDR	W	PWM_BASE + 0x020+ PWM_NUM*0x40	[31:0]	PWM_BUFO_BA SE_ADDR	USER_DEFINED	Base address of memory buffer0 for PWM's waveform data
11	Set PWM_BUFO_SIZ E	W	PWM_BASE + 0x024+ PWM_NUM*0x40	[2:0]	PWM_BUFO_BA SE_ADDR_EXTE ND	USER_DEFINED	It indicates the length of waveform data in memory buffer0. PWM should generate it.
12	Set PWM_BUF1_BA SE_ADDR	W	PWM_BASE + 0x028+ PWM_NUM*0x40	[31:0]	PWM_BUF1_BA SE_ADDR	USER_DEFINED	Base address of memory buffer1 for PWM's waveform data.
13	Set PWM_BUF1_SIZ E	W	PWM_BASE + 0x02C+ PWM_NUM*0x40	[2:0]	PWM_BUF1_BA SE_ADDR_EXTE ND	USER_DEFINED	It indicates the length of waveform data in memory buffer1. PWM should generate it.
14	Set PWM_VALID	W	PWM_BASE + 0x048+ PWM_NUM*0x40	[1:0]	BUF1_VALI D/ BUFO_VALI D	2'b11	Memory1/0 is not empty. When finishing writing data to memory1, write 1 to inform PWM that the data in memory1 are ready.
15	Set PWM_CON	W	PWM_BASE + 0x010+ PWM_NUM*0x40	[14:9]	STOP_BITP OS	USER_DEFINED	In Periodical Mode, it is used to indicate the stop bit position for source data. In FIFO Mode, it is used to indicate the stop bit position of total 64-bits. In Memory Mode, it is used to indicate the stop bit position of the last 32-bits.
16	Set PWM_ENABLE	W	PWM_BASE + 0x0000	[N]	PWM_ENA BLE	1'b1	Enable PWM[N].

5.17 System Timer

5.17.1 Introduction

System Timer (sys_timer) is a 64-bit, non-stop, always-on up-counter which is used as universal timer in system. The counter value of sys_timer is transferred to Application Processing Microcontroller Unit (APMCU), and other micro-processors to provide a uniform system timestamp among operating systems (OSs) (Android Linux, RTOS, etc.).

5.17.2 Features

The System Timer includes the following features:

- A 64-bit, always-on up-counter (this counter is by default enabled to tick with 13 MHz clock period).
- Clock divider to allow timer to tick with 26 MHz/13 MHz/6.5 MHz clock period.
- Hardware (HW) counter increases by an offset compensation when switching to 32 kHz clock source
- 12 x 32-bit counter timeout value (read as 32-bit down counter)
- Security access permission control for each control registers (with one-time lock bit)

5.17.3 Block Diagram

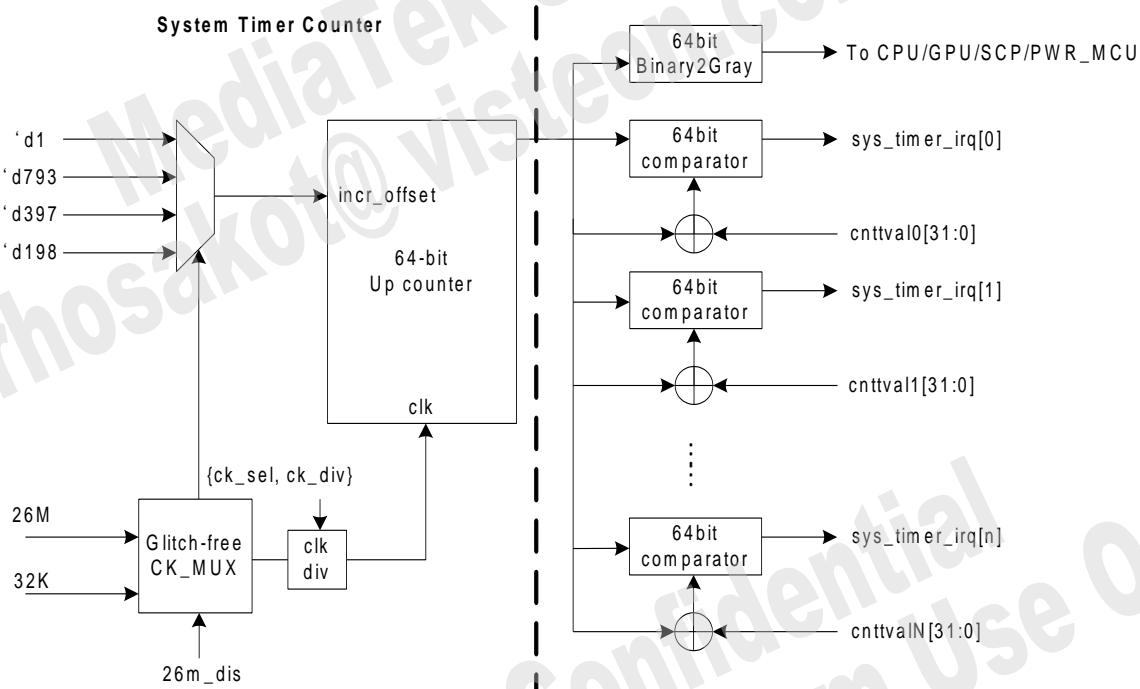


Figure 5-124 sys_timer Block Diagram

sys_timer consists of one 64-bit up counter, one glitch-free clock mux, one clock divider, and multiple 64-bit comparators. The 64-bit up counter is by default enabled, and it will start ticking with 13 MHz clock period after reset is released. The timer tick can be programmed as 26 MHz, 13 MHz, or 6.5 MHz, and can be switched to 32 kHz by power manager when 26 MHz clock source is unavailable.

In 32 kHz mode, the counter's increment offset values change with clock divider settings to compensate the difference in the clock rate. The 64-bit counter value is exported to other sub-system like CPU, GPU, SCP, and so on. To avoid the multi-bit clock domain crossing problem, the counter value is converted into gray-code before output, and a gray-to-binary converter is required to convert the counter value back at receiving side. Aside from exporting the 64-bit counter value to different sub-system, sys_timer also provides multiple comparators which allow programmers to set up 32-bit counter timeout values that can trigger interrupts after timeout. When programmers write a 32-bit offset value into the CNTVAL[n] register, the 32-bit offset value is added to the current 64-bit counter as the expected timeout value. The behavior of timer is described in Figure 5-125. Reading CNTVAL[n] represents the difference between the expected timeout value and current 64-bit counter value. Therefore, the CNTVAL[n] can be seen as a 32-bit down-counter (with 64-bit up-counter counting) which triggers sys_timer_irq[n] when CNTVAL reaches zero.

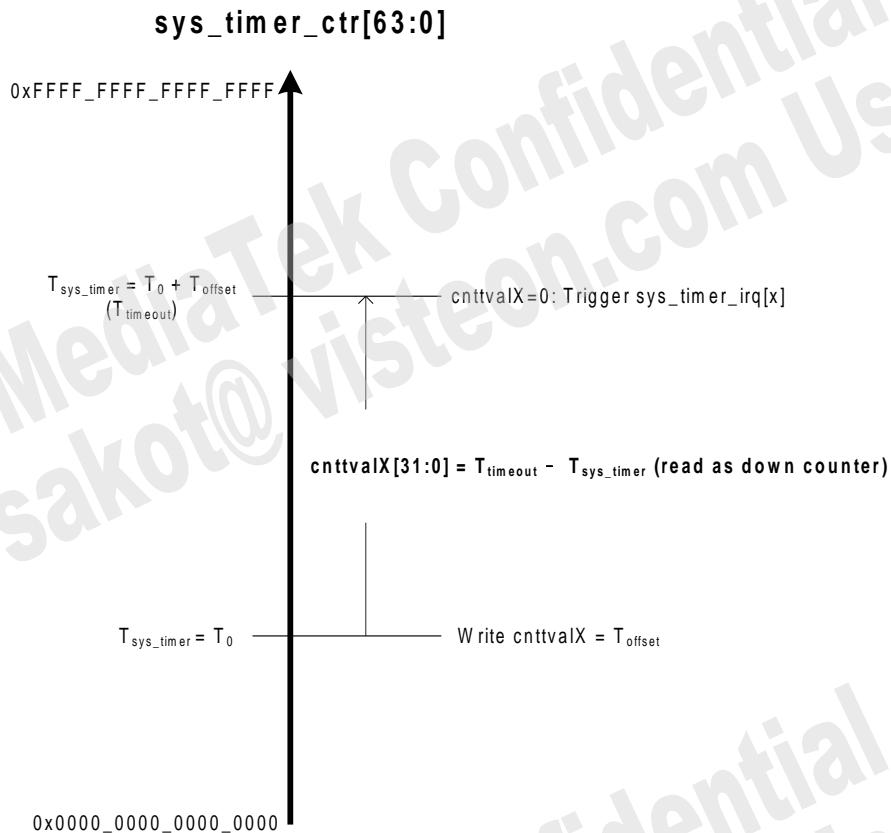


Figure 5-125 Behavior of sys_timer Counter Timeout Value

The security access control is controlled by three registers – CNTWACR (write access control), CNTRACR (read access control), and CNTACR_LOCK (program lock bit). These three registers can only be programmed via secure access, which means that they can only be configured in a secure world or by secure firmware. CNTWACR and

CNTRACR define the non-secure write/read permission to each registers in sys_timer (default: CNTWACR = 0xFFFF_FFF0, CNTRACR = 0xFFFF_FFFF, the bit field definition is described in register definition table).
CNTACR_LOCK is a one-time program register which can lock the CNTWACR and CNTRACR after the configuration is fixed.

5.17.4 Register Definition

For register details, please refer to Chapter 3.15 of “MT2712 I^VI Application Processor Registers”.

5.17.5 Programming Guide

This section describes the following operating sequence:

Note: All the operation mode programming sequences take sys_timer0 as an example.

Table 5-77 sys_timer Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set up timer						
1	sys_timer base address + 0x040	CNTTVAL0_CON	CNTTVAL0_EN	RW	0'h1	Enable timer0
2	sys_timer base address + 0x044	CNTTVAL0	CNTTVAL0	RW	-	Set timeout value
3	sys_timer base address + 0x040	CNTTVAL0_CON	-	RW	0'h3	Enable interrupt
Wait for sys_timer0 issuing interrupt						
Update timer and clear interrupt						
4	sys_timer base address + 0x044	CNTTVAL0	CNTTVAL0	RW	-	Set timeout value
5	sys_timer base address + 0x040	CNTTVAL0_CON	-	RW	0'h13	Clear interrupt
Disable timer						
6	sys_timer base address + 0x040	CNTTVAL0_CON	-	RW	0'h11	Clear interrupt and disable interrupt
7	sys_timer base address + 0x044	CNTTVAL0_CON	-	RW	0'h0	Disable timer

5.18 General-Purpose Timer

5.18.1 Introduction

The Application Processor X General-Purpose Timer (APXGPT) module includes five 32-bit timers and one 64-bit timer. Each timer has four operation modes, namely, ONE-SHOT, REPEAT, KEEP-GO and FREERUN. And they can operate on one of the two clock sources, Real-Time Clock (RTC) (32.768 kHz) and system clock (13 MHz).

5.18.2 Features

The four operation modes for GPT are ONE-SHOT, REPEAT, KEEP-GO and FREERUN. For the details, please refer to Table 5-78.

Table 5-78 Operation Modes of GPT

Mode	Auto Stop	Interrupt Supported	Count Behavior	When GPTn_COUNT Equals GPTn_COMPARE	Example: Compare is set to 2 (Underline means interrupt asserted)
ONE-SHOT	Yes	Yes	Count stops when GPTn_COUNT equals GPTn_COMPARE	EN is reset to 0	0,1, <u>2</u> ,2,2,2,2,2,2,2,2,...
REPEAT	No	Yes	Count is reset to 0 when GPTn_COUNT equals GPTn_COMPARE	Count is reset to 0	0,1, <u>2</u> ,0,1, <u>2</u> ,0,1, <u>2</u> ,0,1, <u>2</u> ...
KEEP-GO	No	Yes	Count is reset to 0 when count overflows	-	0,1, <u>2</u> ,3,4,5,6,7,8,9,10,...
FREERUN	No	No	Count is reset to 0 when count overflows	-	0,1,2,3,4,5,6,7,8,9,10,...

Note:

GPTn_COUNT (apxgpt Base address+(0x0018+0x10*(n-1))), GPTn_COMPARE (apxgpt Base address + (0x001C+0x10*(n-1))) (n=1,2,3,4,5,6).

Each timer's operation is independent and can be programmed to select the clock source of RTC clock (32.768 kHz) or system clock (13 MHz). After the clock source is determined, the division ratio of the selected clock can be programmed. The division ratio can be fine-granulated as 1 to 13 and coarse-granulated as 16, 32 and 64.

5.18.3 Block Diagram

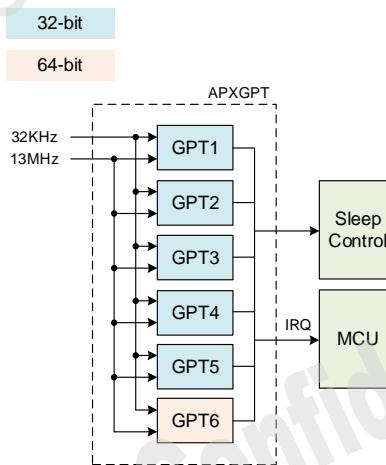


Figure 5-126 APXGPT Block Diagram

To program and use GPT, please note:

The counter value can be read at any time, even when the clock source is RTC clock.

- The compare value can be programmed at any time.

For the GPT6 64-bit timer, the read operation of the 64-bit timer value will be separated into two Advanced Peripherals Bus (APB) reads since an APB read is of 32-bit width. To perform the read of 64-bit timer value, the lower word should be read first before the higher word. The read operation of the lower word will freeze the “read value” of the higher word, but will not freeze the timer from counting. This ensures that the separated read operation acquires the correct timer value. If two tasks, e.g. task A and task B, both perform the read of 64-bit timer value, task A will first read the lower word of the value, and then task B will read the lower word of the value. Either of the two tasks reads the higher word of the timer value, and the obtained value will be the time when task B reads the lower word of the timer value. To guarantee task A reads the correct 64-bit timer value, some software procedures are required, e.g. semaphore.

5.18.4 Register Definition

For register details, please refer to Chapter 3.16 of “MT2712 IVI Application Processor Registers”.

5.18.5 Programming Guide

This section describes the following operating sequences. The programming sequences of the operation modes take GPT1 as an example.

5.18.5.1 APXGPT ONE-SHOT Mode

Table 5-79 APXGPT ONE-SHOT Mode Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set APXGPT control register						
1	APXGPT base address + 0x014	GPT1_CLK	CLK1	RW	Please refer to Description	Select GPT1 clock source, 00: System clock (13 MHz) 01: RTC clock (32.768 kHz)
2	APXGPT base address + 0x014	GPT1_CLK	CLKDIV1	RW	Please refer to Description	Set GPT1 clock divider 0000: Clock source divided by 1 0001: Clock source divided by 2 0010: Clock source divided by 3 0011: Clock source divided by 4 0100: Clock source divided by 5 0101: Clock source divided by 6 0110: Clock source divided by 7 0111: Clock source divided by 8 1000: Clock source divided by 9 1001: Clock source divided by 10 1010: Clock source divided by 11 1011: Clock source divided by 12 1100: Clock source divided by 13 1101: Clock source divided by 16 1110: Clock source divided by 32 1111: Clock source divided by 64

Step	Address	Register Name	Local Address	R/W	Value	Description
3	APXGPT base address + 0x010	GPT1_CON	MODE1	RW	2'b00	Set GPT1 control register for ONE-SHOT mode. 00: ONE-SHOT mode 01: REPEAT mode 10: KEEP-GO mode 11: FREE RUN mode
4	APXGPT base address + 0x01C	GPT1_COMPARE	COMPARE1	RW	Please refer to Description	Set compare value of GPT1
5	APXGPT base address + 0x000	GPT_IRQEN	IRQEN1	RW	1'b1	Set IRQEN1= "1" to enable GPT1 interrupt
Enable GPT1 and wait GPT1 Interrupt						
6	APXGPT base address + 0x010	GPT1_CON	EN1	RW	1'b1	Set EN1= "1" to enable GPT1
7	APXGPT base address + 0x004	GPT_IRQSTA	IRQSTA1	RO	Please refer to Description	Wait IRQSTA1= "1" 0: No associated interrupt is generated 1: Associated interrupt is pending and waiting for service
Clear GPT1 interrupt						
8	APXGPT base address + 0x008	GPT_IRQACK	IRQACK1	WO	1'b1	Set IRQACK1= "1" to clear GPT1 interrupt

5.18.5.2 APXGPT REPEAT Mode

Table 5-80 APXGPT REPEAT Mode Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set APXGPT control register						
1	APXGPT base address + 0x014	GPT1_CLK	CLK1	RW	Please refer to Description	Select GPT1 clock source, 00: System clock (13 MHz) 01: RTC clock (32.768 kHz)
2	APXGPT base address + 0x014	GPT1_CLK	CLKDIV1	RW	Please refer to Description	Set GPT1 clock divider 0000: Clock source divided by 1 0001: Clock source divided by 2 0010: Clock source divided by 3 0011: Clock source divided by 4 0100: Clock source divided by 5 0101: Clock source divided by 6 0110: Clock source divided by 7 0111: Clock source divided by 8 1000: Clock source divided by 9 1001: Clock source divided by 10 1010: Clock source divided by 11 1011: Clock source divided by 12 1100: Clock source divided by 13 1101: Clock source divided by 16 1110: Clock source divided by 32 1111: Clock source divided by 64

Step	Address	Register Name	Local Address	R/W	Value	Description
3	APXGPT base address + 0x010	GPT1_CON	MODE1	RW	2'b01	Set GPT1 control register for REPEAT mode. 00: ONE-SHOT mode 01: REPEAT mode 10: KEEP-GO mode 11: FREERUN mode
4	APXGPT base address + 0x01C	GPT1_COMPARE	COMPARE1	RW	Please refer to Description	Set compare value of GPT1
5	APXGPT base address + 0x000	GPT_IRQEN	IRQEN1	RW	1'b1	Set IRQEN1= "1" to enable GPT1 interrupt
Enable GPT1 and wait GPT1 Interrupt						
6	APXGPT base address + 0x010	GPT1_CON	EN1	RW	1'b1	Set EN1= "1" to enable GPT1
7	APXGPT base address + 0x004	GPT_IRQSTA	IRQSTA1	RO	Please refer to Description	Wait IRQSTA1= "1" 0: No associated interrupt is generated 1: Associated interrupt is pending and waiting for service
Clear GPT1 interrupt						
8	APXGPT base address + 0x008	GPT_IRQACK	IRQACK1	WO	1'b1	Set IRQACK1= "1" to clear GPT1 interrupt

5.18.5.3 APXGPT KEEP-GO Mode

Table 5-81 APXGPT KEEP-GO Mode Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set APXGPT control register						
1	APXGPT base address + 0x014	GPT1_CLK	CLK1	RW	Please refer to Description	Select GPT1 clock source, 00: System clock (13 MHz) 01: RTC clock (32.768 kHz)
2	APXGPT base address + 0x014	GPT1_CLK	CLKDIV1	RW	Please refer to Description	Set GPT1 clock divider 0000: Clock source divided by 1 0001: Clock source divided by 2 0010: Clock source divided by 3 0011: Clock source divided by 4 0100: Clock source divided by 5 0101: Clock source divided by 6 0110: Clock source divided by 7 0111: Clock source divided by 8 1000: Clock source divided by 9 1001: Clock source divided by 10 1010: Clock source divided by 11 1011: Clock source divided by 12 1100: Clock source divided by 13 1101: Clock source divided by 16 1110: Clock source divided by 32 1111: Clock source divided by 64

Step	Address	Register Name	Local Address	R/W	Value	Description
3	APXGPT base address + 0x010	GPT1_CON	MODE1	RW	2'b10	Set GPT1 control register for KEEP-GO mode. 00: ONE-SHOT mode 01: REPEAT mode 10: KEEP-GO mode 11: FREERUN mode
4	APXGPT base address + 0x01C	GPT1_COMPARE	COMPARE1	RW	Please refer to Description	Set compare value of GPT1
5	APXGPT base address + 0x000	GPT_IRQEN	IRQEN1	RW	1'b1	Set IRQEN1= "1" to enable GPT1 interrupt
Enable GPT1						
6	APXGPT base address + 0x010	GPT1_CON	EN1	RW	1'b1	Set EN1= "1" to enable GPT1
7	APXGPT base address + 0x004	GPT_IRQSTA	IRQSTA1	RO	Please refer to Description	Wait IRQSTA1= "1" 0: No associated interrupt is generated 1: Associated interrupt is pending and waiting for service
Read GPT1 counter value and clear GPT1 interrupt						
8	APXGPT base address + 0x018	GPT1_COUNT	COUNTER1	RO	Please refer to Description	Timer counter of GPT1
9	APXGPT base address + 0x008	GPT_IRQACK	IRQACK1	WO	1'b1	Set IRQACK1= "1" to clear GPT1 interrupt

5.18.5.4 APXGPT FREERUN Mode

Table 5-82 APXGPT FREERUN Mode Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set APXGPT control register						
1	APXGPT base address + 0x014	GPT1_CLK	CLK1	RW	Please refer to Description	Select GPT1 clock source, 00: System clock (13 MHz) 01: RTC clock (32.768 kHz)
2	APXGPT base address + 0x014	GPT1_CLK	CLKDIV1	RW	Please refer to Description	Set GPT1 clock divider 0000: Clock source divided by 1 0001: Clock source divided by 2 0010: Clock source divided by 3 0011: Clock source divided by 4 0100: Clock source divided by 5 0101: Clock source divided by 6 0110: Clock source divided by 7 0111: Clock source divided by 8 1000: Clock source divided by 9 1001: Clock source divided by 10 1010: Clock source divided by 11 1011: Clock source divided by 12 1100: Clock source divided by 13 1101: Clock source divided by 16 1110: Clock source divided by 32 1111: Clock source divided by 64

Step	Address	Register Name	Local Address	R/W	Value	Description
3	APXGPT base address + 0x010	GPT1_CON	MODE1	RW	2'b11	Set GPT1 control register for FREERUN mode. 00: ONE-SHOT mode 01: REPEAT mode 10: KEEP-GO mode 11: FREERUN mode
Enable GPT1						
4	APXGPT base address + 0x010	GPT1_CON	EN1	RW	1'b1	Set EN1= "1" to enable GPT1
Read GPT1 counter value						
5	APXGPT base address + 0x018	GPT1_COUNT	COUNTER1	RO	Please refer to Description	Timer counter of GPT1

5.18.5.5 GPT6 64-bit Counter Read

Table 5-83 GPT6 64-bit Counter Read Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Read GPT6 counter low 32-bit						
1	APXGPT base address + 0x068	GPT6_COUNTL	-	RO	Please refer to Description	Lower word of timer count of GPT6
Read GPT6 counter high 32-bit						
2	APXGPT base address + 0x078	GPT6_COUNTH	-	RO	Please refer to Description	Higher word of timer count of GPT6

5.19 General-Purpose Timer (GPT2)

5.19.1 Introduction

The Application Processor X General-Purpose Timer (APXGPT) module includes five 32-bit timers and one 64-bit timer. Each timer has four operation modes, namely, ONE-SHOT, REPEAT, KEEP-GO and FREERUN, and can operate on one of the two clock sources, Real-Time Clock (RTC) (32.768 kHz) and system clock (13 MHz).

5.19.2 Features

The four operation modes for GPT are ONE-SHOT, REPEAT, KEEP-GO and FREERUN. For the details, please refer to Table 5-84.

Table 5-84 Operation Mode of GPT

Mode	Auto Stop	Interrupt Supported	Count Behavior	When GPTn_COUNT Equals GPTn_COMPARE	Example: Compare is set to 2 (Underline means interrupt asserted)
ONE-SHOT	Yes	Yes	Count stops when GPTn_COUNT equals GPTn_COMPARE	EN is reset to 0	0,1, <u>2</u> ,2,2,2,2,2,2,2,2,...
REPEAT	No	Yes	Count is reset to 0 when GPTn_COUNT equals GPTn_COMPARE	Count is reset to 0	0,1, <u>2</u> ,0,1, <u>2</u> ,0,1, <u>2</u> ,0,1, <u>2</u> ...
KEEP-GO	No	Yes	Count is reset to 0 when count overflows	-	0,1, <u>2</u> ,3,4,5,6,7,8,9,10,...
FREERUN	No	No	Count is reset to 0 when count overflows	-	0,1,2,3,4,5,6,7,8,9,10,...

Note:

GPTn_COUNT (apxgpt Base address+(0x0018+0x10*(n-1))), GPTn_COMPARE (apxgpt Base address + (0x001C+0x10*(n-1))) (n=1,2,3,4,5,6).

Each timer's operation is independent and can be programmed to select the clock source, RTC clock (32.768 kHz) or system clock (13 MHz). After the clock source is determined, the division ratio of the selected clock can be programmed. The division ratio can be fine-granulated as 1 to 13 and coarse-granulated as 16, 32 and 64.

5.19.3 Block Diagram

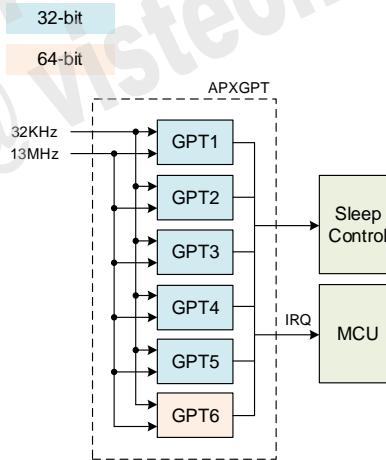


Figure 5-127 APXGPT Block Diagram

To program and use GPT, please note:

The counter value can be read at any time, even when the clock source is RTC clock.

- The compare value can be programmed at any time.

For the GPT6 64-bit timer, the read operation of the 64-bit timer value will be separated into two Advanced Peripherals Bus (APB) reads since an APB read is of 32-bit width. To perform the read of 64-bit timer value, the lower word should be read first, before the higher word. The read operation of the lower word will freeze the “read value” of the higher word, but will not freeze the timer from counting. This ensures that the separated read operation acquires the correct timer value. If two tasks, e.g. task A and task B, both perform the read of 64-bit timer value, task A will first read the lower word of the value, and then task B will read the lower word of the value. Either of the two tasks reads the higher word of the timer value, and the obtained value will be the time when task B reads the lower word of the timer value. To guarantee task A reads the correct 64-bit timer value, some software procedures are required, e.g. semaphore.

5.19.4 Register Definition

For register details, please refer to Chapter 3.16 of “MT2712 IVI Application Processor Registers”.

5.19.5 Programming Guide

This section describes the following operating sequences. The programming sequences of all four operation modes take GPT1 as an example.

5.19.5.1 APXGPT ONE-SHOT Mode

Table 5-85 APXGPT ONE-SHOT Mode Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set APXGPT control register						
1	APXGPT base address + 0x014	GPT1_CLK	CLK1	RW	Please refer to Description	Select GPT1 clock source, 00: System clock (13 MHz) 01: RTC clock (32.768 kHz)
2	APXGPT base address + 0x014	GPT1_CLK	CLKDIV1	RW	Please refer to Description	Set GPT1 clock divider 0000: Clock source divided by 1 0001: Clock source divided by 2 0010: Clock source divided by 3 0011: Clock source divided by 4 0100: Clock source divided by 5 0101: Clock source divided by 6 0110: Clock source divided by 7 0111: Clock source divided by 8 1000: Clock source divided by 9 1001: Clock source divided by 10 1010: Clock source divided by 11 1011: Clock source divided by 12 1100: Clock source divided by 13 1101: Clock source divided by 16 1110: Clock source divided by 32 1111: Clock source divided by 64
3	APXGPT base address + 0x010	GPT1_CON	MODE1	RW	2'b00	Set GPT1 control register for ONE-SHOT mode. 00: ONE-SHOT mode 01: REPEAT mode 10: KEEP-GO mode 11: FREERUN mode

Step	Address	Register Name	Local Address	R/W	Value	Description
4	APXGPT base address + 0x01C	GPT1_COMPARE	COMPARE1	RW	Please refer to Description	Set compare value of GPT1
5	APXGPT base address + 0x000	GPT_IRQEN	IRQEN1	RW	1'b1	Set IRQEN1= "1" to enable GPT1 interrupt
Enable GPT1 and wait GPT1 Interrupt						
6	APXGPT base address + 0x010	GPT1_CON	EN1	RW	1'b1	Set EN1= "1" to enable GPT1
7	APXGPT base address + 0x004	GPT_IRQSTA	IRQSTA1	RO	Please refer to Description	Wait IRQSTA1= "1" 0: No associated interrupt is generated 1: Associated interrupt is pending and waiting for service
Clear GPT1 interrupt						
8	APXGPT base address + 0x008	GPT_IRQACK	IRQACK1	WO	1'b1	Set IRQACK1= "1" to clear GPT1 interrupt

5.19.5.2 APXGPT REPEAT Mode

Table 5-86 APXGPT REPEAT Mode Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set APXGPT control register						
1	APXGPT base address + 0x014	GPT1_CLK	CLK1	RW	Please refer to Description	Select GPT1 clock source, 00: System clock (13 MHz) 01: RTC clock (32.768 kHz)
2	APXGPT base address + 0x014	GPT1_CLK	CLKDIV1	RW	Please refer to Description	Set GPT1 clock divider 0000: Clock source divided by 1 0001: Clock source divided by 2 0010: Clock source divided by 3 0011: Clock source divided by 4 0100: Clock source divided by 5 0101: Clock source divided by 6 0110: Clock source divided by 7 0111: Clock source divided by 8 1000: Clock source divided by 9 1001: Clock source divided by 10 1010: Clock source divided by 11 1011: Clock source divided by 12 1100: Clock source divided by 13 1101: Clock source divided by 16 1110: Clock source divided by 32 1111: Clock source divided by 64
3	APXGPT base address + 0x010	GPT1_CON	MODE1	RW	2'b01	Set GPT1 control register for REPEAT mode. 00: ONE-SHOT mode 01: REPEAT mode 10: KEEP-GO mode 11: FREERUN mode
4	APXGPT base address + 0x01C	GPT1_COMPARE	COMPARE1	RW	Please refer to Description	Set compare value of GPT1

Step	Address	Register Name	Local Address	R/W	Value	Description
5	APXGPT base address + 0x000	GPT_IRQEN	IRQEN1	RW	1'b1	Set IRQEN1= "1" to enable GPT1 interrupt
Enable GPT1 and wait GPT1 Interrupt						
6	APXGPT base address + 0x010	GPT1_CON	EN1	RW	1'b1	Set EN1= "1" to enable GPT1
7	APXGPT base address + 0x004	GPT_IRQSTA	IRQSTA1	RO	Please refer to Description	Wait IRQSTA1= "1" 0: No associated interrupt is generated 1: Associated interrupt is pending and waiting for service
Clear GPT1 interrupt						
8	APXGPT base address + 0x008	GPT_IRQACK	IRQACK1	WO	1'b1	Set IRQACK1= "1" to clear GPT1 interrupt

5.19.5.3 APXGPT KEEP-GO Mode

Table 5-87 APXGPT KEEP-GO Mode Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set APXGPT control register						
1	APXGPT base address + 0x014	GPT1_CLK	CLK1	RW	Please refer to Description	Select GPT1 clock source, 00: System clock (13 MHz) 01: RTC clock (32.768 kHz)
2	APXGPT base address + 0x014	GPT1_CLK	CLKDIV1	RW	Please refer to Description	Set GPT1 clock divider 0000: Clock source divided by 1 0001: Clock source divided by 2 0010: Clock source divided by 3 0011: Clock source divided by 4 0100: Clock source divided by 5 0101: Clock source divided by 6 0110: Clock source divided by 7 0111: Clock source divided by 8 1000: Clock source divided by 9 1001: Clock source divided by 10 1010: Clock source divided by 11 1011: Clock source divided by 12 1100: Clock source divided by 13 1101: Clock source divided by 16 1110: Clock source divided by 32 1111: Clock source divided by 64
3	APXGPT base address + 0x010	GPT1_CON	MODE1	RW	2'b10	Set GPT1 control register for KEEP-GO mode. 00: ONE-SHOT mode 01: REPEAT mode 10: KEEP-GO mode 11: FREERUN mode
4	APXGPT base address + 0x01C	GPT1_COMPARE	COMPARE1	RW	Please refer to Description	Set compare value of GPT1
5	APXGPT base address + 0x000	GPT_IRQEN	IRQEN1	RW	1'b1	Set IRQEN1= "1" to enable GPT1 interrupt
Enable GPT1						
6	APXGPT base address + 0x010	GPT1_CON	EN1	RW	1'b1	Set EN1= "1" to enable GPT1

Step	Address	Register Name	Local Address	R/W	Value	Description
7	APXGPT base address + 0x004	GPT_IRQSTA	IRQSTA1	RO	Please refer to Description	Wait IRQSTA1= "1" 0: No associated interrupt is generated 1: Associated interrupt is pending and waiting for service
Read GPT1 counter value and clear GPT1 interrupt						
8	APXGPT base address + 0x018	GPT1_COUNT	COUNTER1	RO	Please refer to Description	Timer counter of GPT1
9	APXGPT base address + 0x008	GPT_IRQACK	IRQACK1	WO	1'b1	Set IRQACK1= "1" to clear GPT1 interrupt

5.19.5.4 APXGPT FREERUN Mode

Table 5-88 APXGPT FREERUN Mode Setting Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Set APXGPT control register						
1	APXGPT base address + 0x014	GPT1_CLK	CLK1	RW	Please refer to Description	Select GPT1 clock source, 00: System clock (13 MHz) 01: RTC clock (32.768 kHz)
2	APXGPT base address + 0x014	GPT1_CLK	CLKDIV1	RW	Please refer to Description	Set GPT1 clock divider 0000: Clock source divided by 1 0001: Clock source divided by 2 0010: Clock source divided by 3 0011: Clock source divided by 4 0100: Clock source divided by 5 0101: Clock source divided by 6 0110: Clock source divided by 7 0111: Clock source divided by 8 1000: Clock source divided by 9 1001: Clock source divided by 10 1010: Clock source divided by 11 1011: Clock source divided by 12 1100: Clock source divided by 13 1101: Clock source divided by 16 1110: Clock source divided by 32 1111: Clock source divided by 64
3	APXGPT base address + 0x010	GPT1_CON	MODE1	RW	2'b11	Set GPT1 control register for FREERUN mode. 00: ONE-SHOT mode 01: REPEAT mode 10: KEEP-GO mode 11: FREERUN mode
Enable GPT1						
4	APXGPT base address + 0x010	GPT1_CON	EN1	RW	1'b1	Set EN1= "1" to enable GPT1
Read GPT1 counter value						
5	APXGPT base address + 0x018	GPT1_COUNT	COUNTER1	RO	Please refer to Description	Timer counter of GPT1

5.19.5.5 GPT6 64-bit Counter Read

Table 5-89 GPT6 64-bit Counter Read Flow

Step	Address	Register Name	Local Address	R/W	Value	Description
Read GPT6 counter low 32-bit						
1	APXGPT base address + 0x068	GPT6_COUNTL	-	RO	Please refer to Description	Lower word of timer count of GPT6
Read GPT6 counter high 32-bit						
2	APXGPT base address + 0x078	GPT6_COUNTH	-	RO	Please refer to Description	Higher word of timer count of GPT6

5.20 Thermal Controller

5.20.1 Introduction

In latest platform, thermal management is very fundamental. The thermal management controls the platform's computing performance to meet the requirement and keep the Raven within the temperature constraints. Operation at an over-high temperature for a long time will run the risk of reducing Raven reliability.

In MT2712, several temperature sensors are embedded in possible hot spots on the die. The thermal controller module executes a periodic measurement for each hot spot. The temperature readings are readable for software.

In order to minimize software's effort to monitor temperature, the thermal controller will generate interrupts to CPU to inform it of any abnormal condition.

5.20.2 Features

The features of Thermal Controller module are:

- Support up to 4 thermal sensors
- Periodic temperature measurement
- Temperature monitoring
- Different types of low pass filter for thermal sensor readings

5.20.3 Block Diagram

There are microprocessors with maximum frequencies over 1 GHz in MT2712, and the transistor count is also large. The power consumed by microprocessors occupies a high percentage of whole chip's power consumption. Besides the microprocessors, EMI will also be a source of power consumption because it provides high DRAM data bandwidth to other modules in MT2712.

Thermal controller is implemented for software to operate MT2712 with a pre-defined temperature range. Or it will have function failure and issues of reliability. According to the temperature measurement, the system performance can be adjusted and the system design for power dissipation can also be monitored.

The hottest location in MT2712 may be different in different applications. Besides, when the thermal controller informs the software of an abnormal condition, the following power reduction action should be efficient and with low latency.

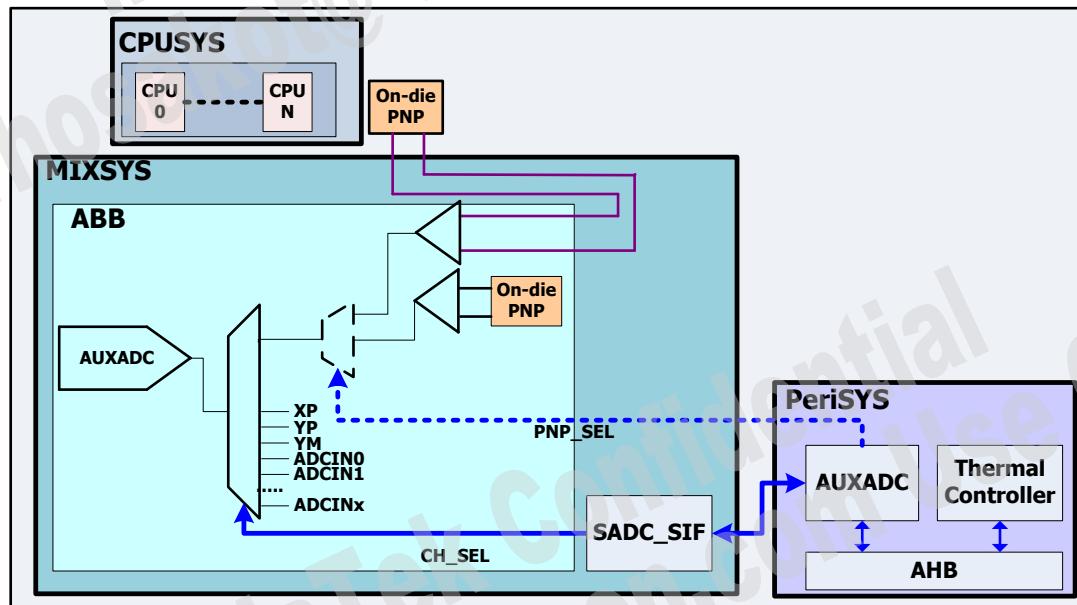


Figure 5-128 System Level Thermal Controller Block Diagram

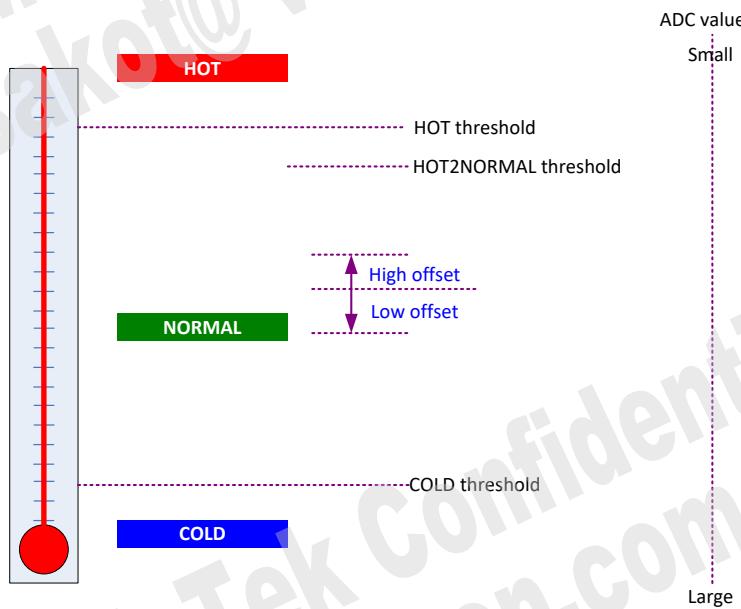


Figure 5-129 System Temperature Measurement

5.20.4 Register Definition

For register details, please refer to Chapter 3.17 of “MT2712 IVI Application Processor Registers”.

5.20.5 Programming Guide

5.20.5.1 Programming Flow

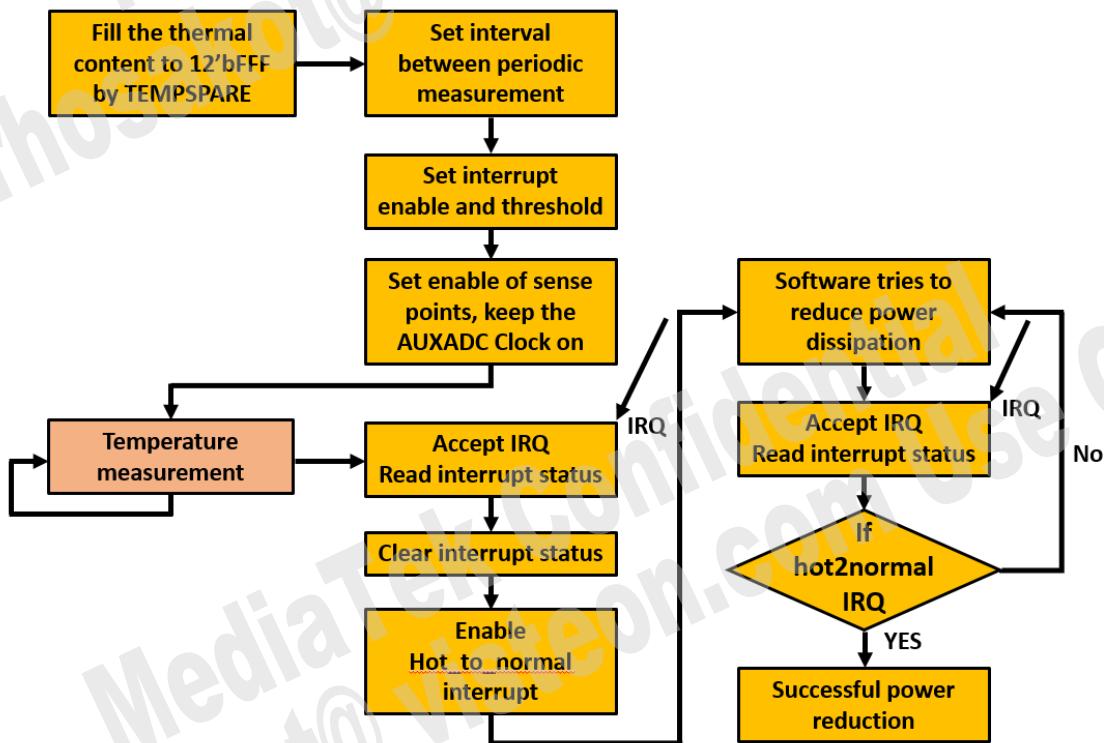


Figure 5-130 Programming Flow

1. Fill the thermal content to 12'bFFF, by accessing TEMPSPARE

```

vWriteREG(TEMPMONCTL1, 'h0);
vWriteREG(TEMPMONCTL2, 'h0);
vWriteREG(TEMPAHBPOLL, 'h0); // Polling interval to check if temp sensor is ready
vWriteREG(TEMPAHBTO, 'hFF); // If exceeds this polling time, IRQ would be inserted
vWriteREG(TEMPSPARE0, 'h1FFF); // Fill the TEMPSPARE0 register as 'h1FFF
vWriteREG(TEMPPNPMUXADDR, 32'hTS_CON1); // The auxadc mux address to select Thermal channel, and
please refer to mixsys.doc
vWriteREG(TEMPADCENADDR, 32'TEMPSPARE1); // The auxadc enable address to trigger Thermal Sensor,
please refer to auxadc.doc
vWriteREG(TEMPADCVLIDADDR, 32'hTEMPSPARE1); // The auxadc status address to check if Thermal Sensor
reading is valid, please refer to auxadc.doc

```

vWriteREG(TEMPADCVOLTADDR, 32'hTEMPSPAREO); // The auxadc temperature address for the value read back from temp sensor, please refer to auxadc.doc

vWriteREG(TEMPPRDCTRL, 'h0); // Use TEMPSPAREO as valid address

vWriteREG(TEMPADCVALIDMASK, 'h2c); // Set auxadc valid polarity to 0

vWriteREG(TEMPPMONCTL0, 'h0F); // Enable all sense points including the debug one, Wait until the content of TEMPIIMMD are filled by 'hFFF

2. Set interval between two neighboring periodic temperature measurement, if the MODULE clock is 66 MHz

vWriteREG(TEMPPMONCTL1, 'h3FF); // Counting unit is 1024*15.15 ns = 15.5 us

vWriteREG(TEMPPMONCTL2, 'h3FF); // Sensing interval is 1024*15.5us = 15.87 ms

vWriteREG(TEMPAHBPOLL, 'h0F); // Polling interval to check if temp sense is ready

vWriteREG(TEMPAHBTO, 'hFF); // If this polling time is exceeded, IRQ would be inserted

vWriteREG(TEMPADCPNP0, 32'h0); // The auxadc PNP data to select Thermal Sensor, and please refer to mixsys.doc

vWriteREG(TEMPADCPNP1, 32'h1); // The auxadc PNP data to select Thermal Sensor, and please refer to mixsys.doc

vWriteREG(TEMPADCPNP2, 32'h2); // The auxadc PNP data to select Thermal Sensor, and please refer to mixsys.doc

vWriteREG(TEMPADCEN, 32'h800); // The auxadc enables data(CH-11) to trigger Thermal Sensor, and please refer to auxadc.doc

vWriteREG(TEMPADCMUX, 32'h800); // The auxadc enables data (CH-11) to clear trigger Thermal Sensor, and please refer to auxadc.doc

vWriteREG(TEMPPNPMUXADDR, 32'hTS_CON1); // The auxadc mux address to select Thermal channel, and please refer to mixsys.doc

vWriteREG(TEMPADCENADDR, 32'hAUXADC_CON1_SET); // The auxadc enables address to trigger Thermal Sensor, and please refer to auxadc.doc

vWriteREG(TEMPADCMUXADDR, 32'hAUXADC_CON1_CLR); // The auxadc enables address to trigger Thermal Sensor, and please refer to auxadc.doc

vWriteREG(TEMPADCVALIDADDR, 32'hAUXADC_CON3); // The auxadc status address to check if Thermal Sensor reading is valid, and please refer to auxadc.doc

vWriteREG(TEMPADCVOLTADDR, 32'hAUXADC_DAT11); // The auxadc temperature address for the value read back from temp sensor, and please refer to auxadc.doc

vWriteREG(TEMPPRDCTRL, 'h0); // Use AUXADC_DAT11 as valid address

vWriteREG(TEMPADCVALIDMASK, 'h2c); // Set auxadc valid polarity to 0

vWriteREG(TEMPWRITECTRL, 'h3); // Enable write transaction type

3. Set monitoring threshold and SPM wake up event

vWriteREG(TEMPHTHRE, 'hxxx); // Set hot threshold

vWriteREG(TEMPCTHRE, 'hxxx); // Set cold threshold

vWriteREG(TEMPH2NTHRE, 'hxxx); // Set hot to normal threshold

vWriteREG(TEMPPROTCTL, 'h20xxx); // Set hot to wakeup event control

vWriteREG(TEMPPROTT, 'hxxx); // Set hot to HOT wakeup event

```
vWriteREG(TEMPMONINT, 'h8000001F); // Enable interrupt  
  
4. Set sense points enable  
vWriteREG(TEMPMONCTL0, 'h07); // Enable all three sense points  
  
5. Accept IRQ  
vReadREG(TEMPMONINTSTS); // Read interrupt and clear interrupt status  
  
6. Read temperature readings (optional)  
vReadREG(TEMPMSR0); // Read temperature reading of sense point 0  
vReadREG(TEMPMSR1); // Read temperature reading of sense point 1  
vReadREG(TEMPMSR2); // Read temperature reading of sense point 2  
  
7. Release pause of periodic temperature measurement  
vWriteREG(TEMPMSRCTL1, vReadREG(TEMPMSRCTL1) & 0xFFE);
```

5.20.5.2 Immediate Temperature Measurement

After each immediate temperature measurement is done, software must disable the immediate mode.

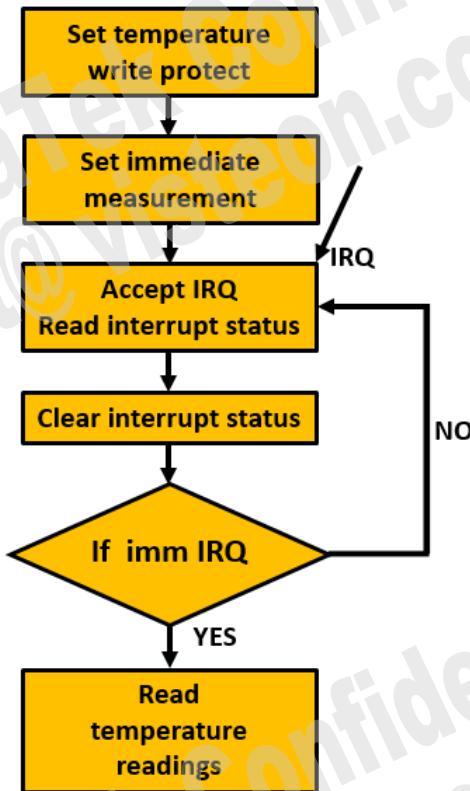


Figure 5-131 Immediate Measurement Programming Flow

5.21 Infrared Receiver

5.21.1 Introduction

The Infrared Receiver (IRRX) module can receive the Infra-Red signal and support NEC protocol, RC5 protocol, and RC6 protocol. This IR receiver can decode various IR transmission protocols:

- Pulse-width coding, such as the NEC IR transmission protocol. See Figure 5-132.
- Bi-phase coding, e.g. RC5, RC6, and RCMM. See Figure 5-133. The signal can be decoded by a constant period sample pulse.

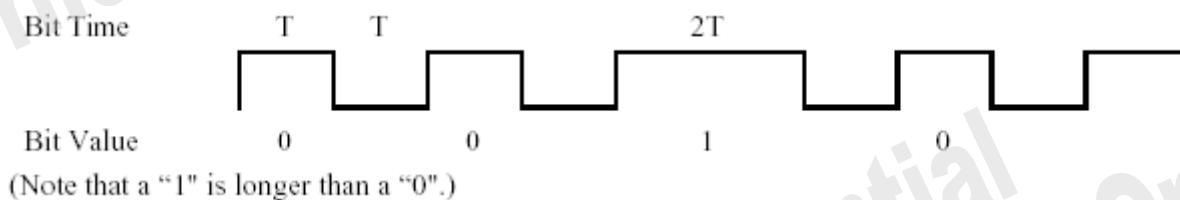


Figure 5-132 Pulse-width Coding

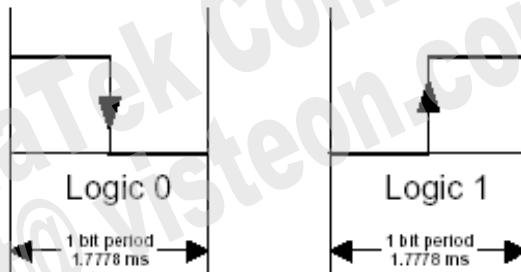


Figure 5-133 Bi-phase Coding

5.21.2 Features

The IRRX module supports the following features:

- NEC protocol
- RC5/RC6 protocol
- RCMM protocol
- Software decoder

5.21.3 Block Diagram

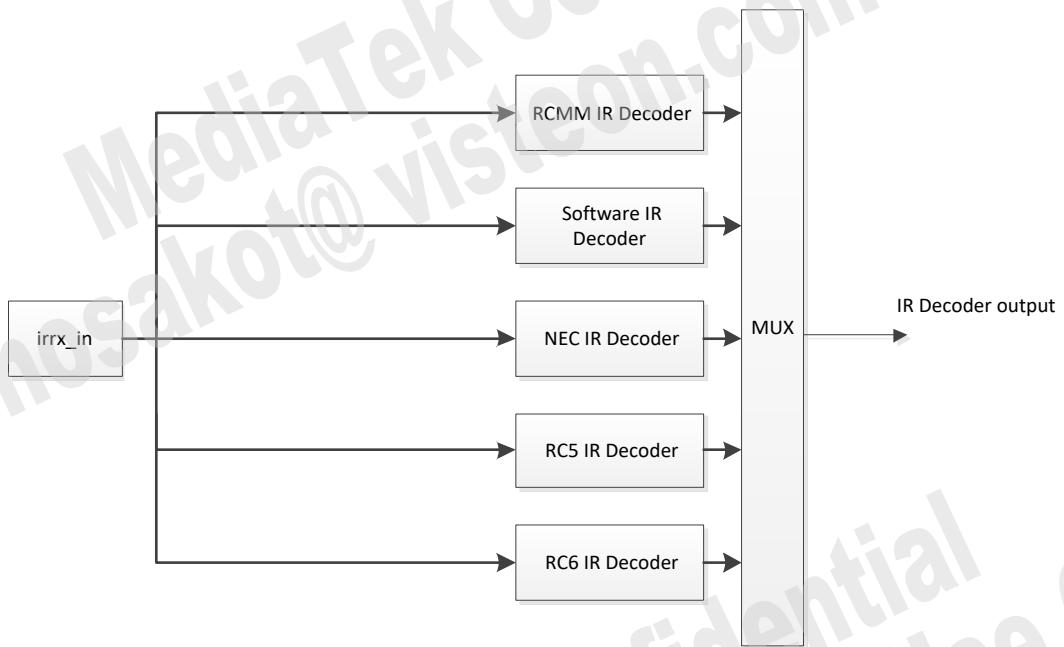


Figure 5-134 Infrared Receiver Block Diagram

5.21.4 IRRX AC Timing

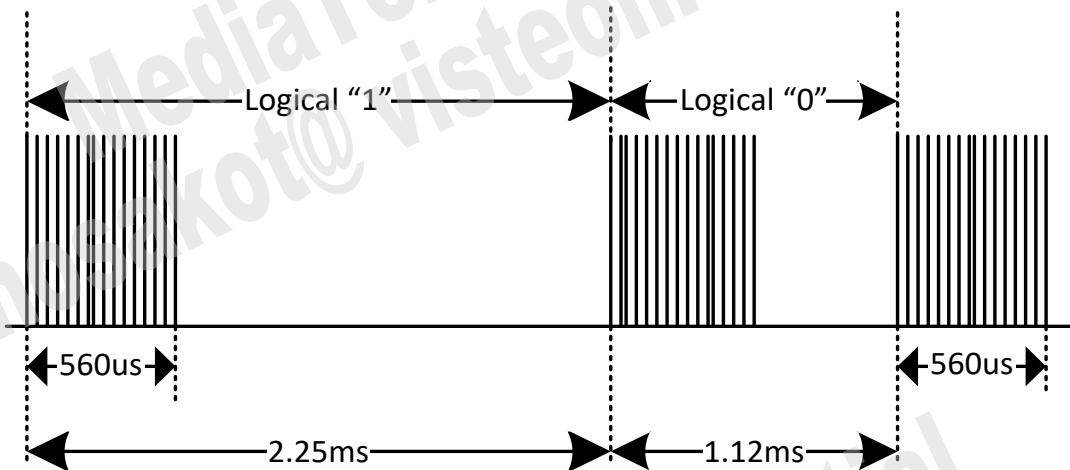


Figure 5-135 NEC Protocol

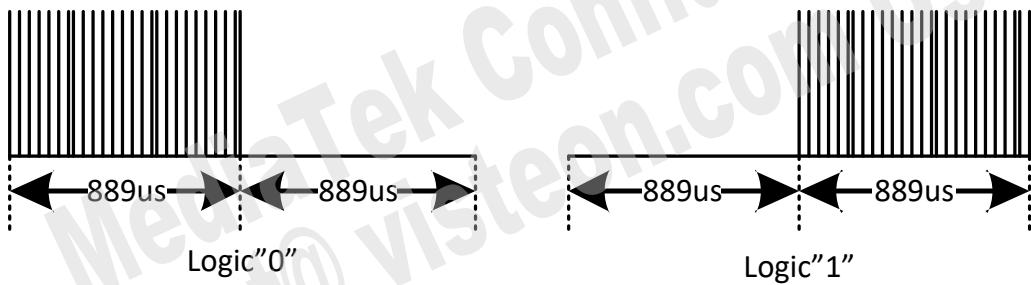


Figure 5-136 RC5 Protocol

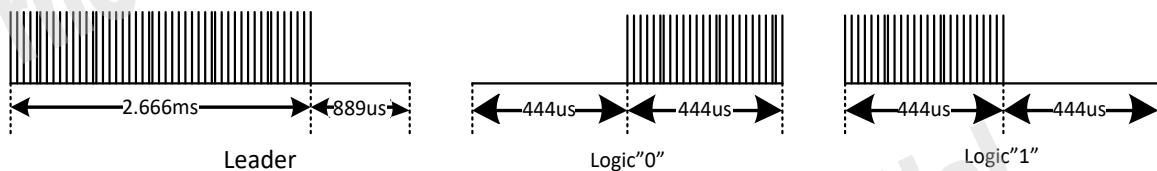


Figure 5-137 RC6 Protocol



Figure 5-138 RCMM Protocol

5.21.5 Register Definition

For register details, please refer to Chapter 3.18 of “MT2712 IVI Application Processor Registers”.

5.21.6 Programming Guide

Table 5-90 IRRX Decode Programming Flow

Step	Address	Register Name	Local Address	R/W	Default Value	Description
Set IRRX control register						
1	IRRX base address + 0x0c	PDREG_IRCFGH	-	RW	20'hF0000	Configure IRRX
2	IRRX base address + 0x10	PDREG_IRCFGL	-	RW	8'hff	Configure IRRX
3	IRRX base address + 0x14	PDREG_IRTHD	-		12'h600	Configure IRRX
Enable IRRX and wait IRRX interrupt						

Step	Address	Register Name	Local Address	R/W	Default Value	Description
4	IRRX base address + 0x0c	PDREG_IRCFGH	IREN	RW	1'b0	Set IREN = "1" to enable IRRX
5	IRRX base address + 0x84	PDREG_INTEN	-	RW	1'b0	Set INT_EN = "1" to enable IRRX interrupt
6						Wait IRRX IRQ
Read IRRX decode value and clear IRRX interrupt						
7	IRRX base address + 0x00	PDREG_IRH	-	RO	-	Read IRRX decode value
8	IRRX base address + 0x04	PDREG_IRM	-	RO	-	Read IRRX decode value
9	IRRX base address + 0x08	PDREG_IRL	-	RO	-	Read IRRX decode value
10	IRRX base address + 0x88	PDREG_IR_INTCLR	-	WO	-	Clear IRRX interrupt

5.22 Ethernet QOS

5.22.1 Introduction

The Ethernet QOS (ETHER_QOS) enables a host to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2008.

5.22.2 Features

The ETHER_QOS supports the following features:

- MAC TX/RX
- Audio Video Bridge (AVB) features
 - IEEE802.1QAV
 - Quality of Service (QOS) with credit-base shaper
 - IEEE1588 & IEEE802.1AS
 - Precision Time Protocol (PTP) offload
 - Both 1-step & 2-step
 - BUS interface
 - Advanced eXtensible Interface (AXI) master
 - Advanced Peripherals Bus (APB) slave
 - Physical (PHY) interface
 - MII/RMII/RGMII

5.22.3 Block Diagram

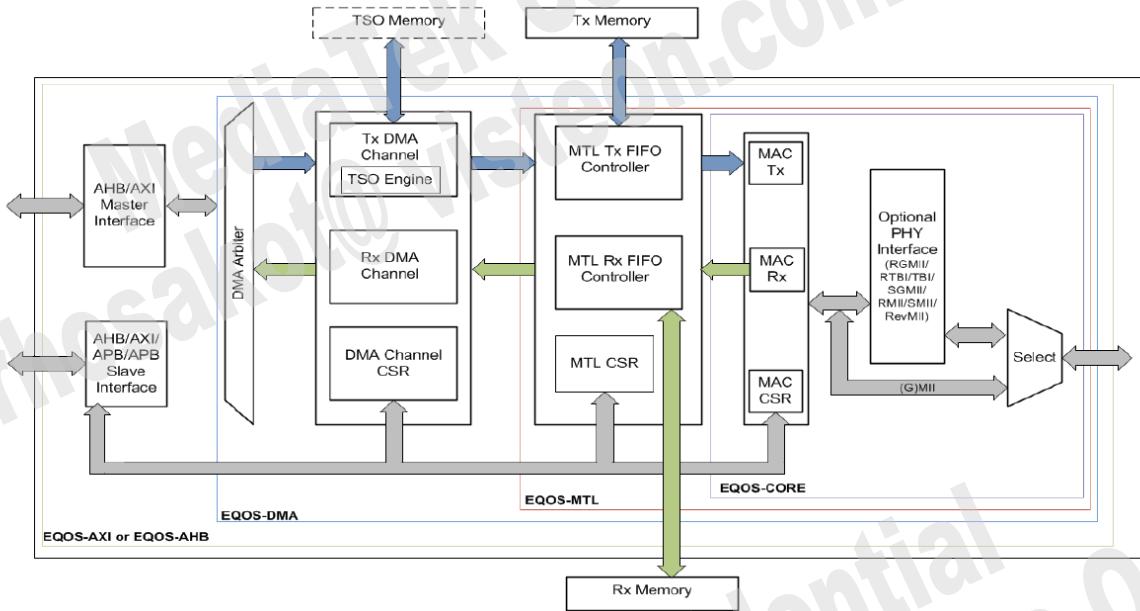


Figure 5-139 Ethernet with AVB MAC Controller Block Diagram

5.22.4 ETHER_QOS AC Timing

5.22.4.1 MII

Table 5-91 MII Interface Timing Requirement

Parameter	Symbol	Min.	Typ.	Max.	Unit
TXC/RXC period	TXC/RXC	-	40	-	ns
TXC rising edge to TXD/TXEN/TXER	Tdelay	10	-	15	ns
RXD/RXDV/RXER setup time to RXC rising edge	TsetupR	1	-	-	ns
RXD/RXDV/RXER hold time to RXC rising edge	TholdR	1	-	-	ns
TXC/RXC duty cycle	-	35	-	65	%
TXC/RXC rising time (from 0.8 V to 2 V)	-	1	-	5	ns
TXC/RXC falling time (from 2 V to 0.8V)	-	1	-	5	ns

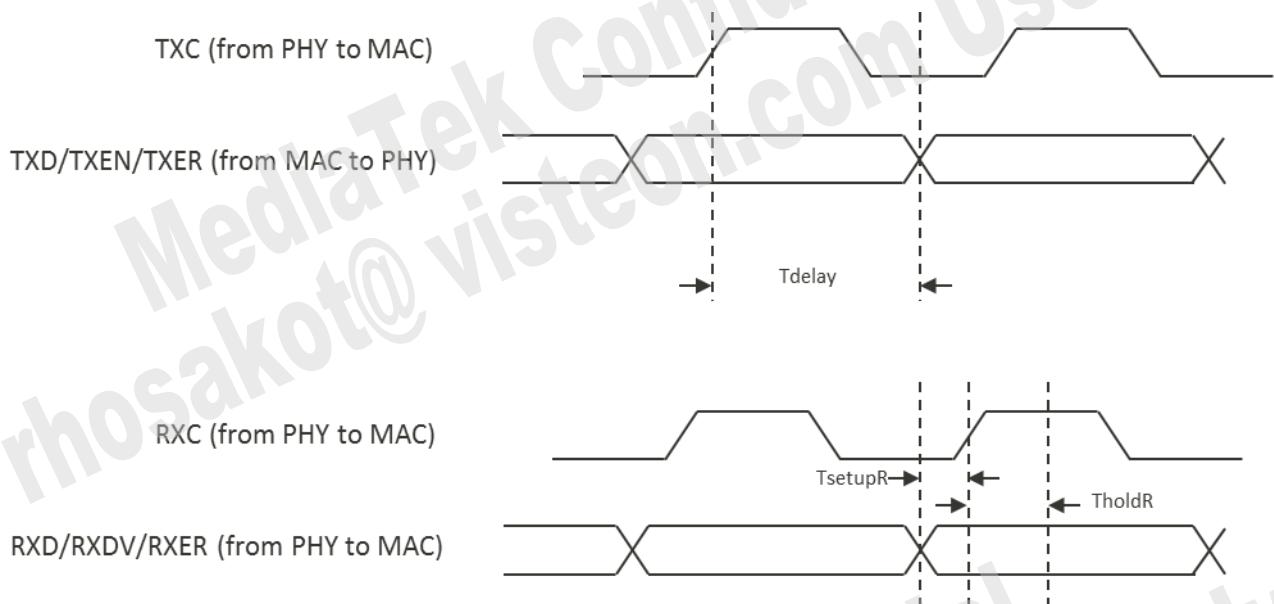
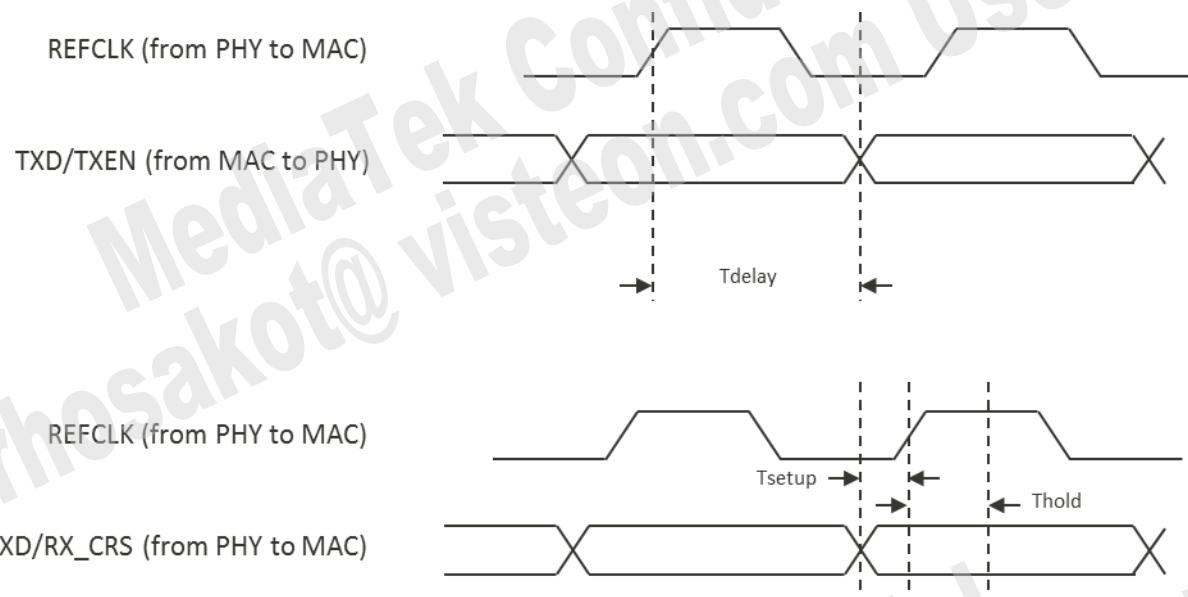


Figure 5-140 MII Interface Timing Diagram

5.22.4.2 RMII

Table 5-92 RMII Interface Timing Requirement

Parameter	Symbol	Min.	Typ.	Max.	Unit
REFCLK period	Tperiod	15	20	25	ns
REFCLK rising edge to TXD/TXEN	Tdelay	10	-	14	ns
RXD/RX_CRS setup time to REFCLK rising edge	Tsetup	1	-	-	ns
RXD/RX_CRS hold time to REFCLK rising edge	Thold	1	-	-	ns
REFCLK duty cycle	-	45	50	55	%
REFCLK rising time (from 0.8 V to 2 V)	-	1	-	5	ns
REFCLK falling time (from 2 V to 0.8 V)	-	1	-	5	ns

**Figure 5-141 RMII Interface Timing Diagram**

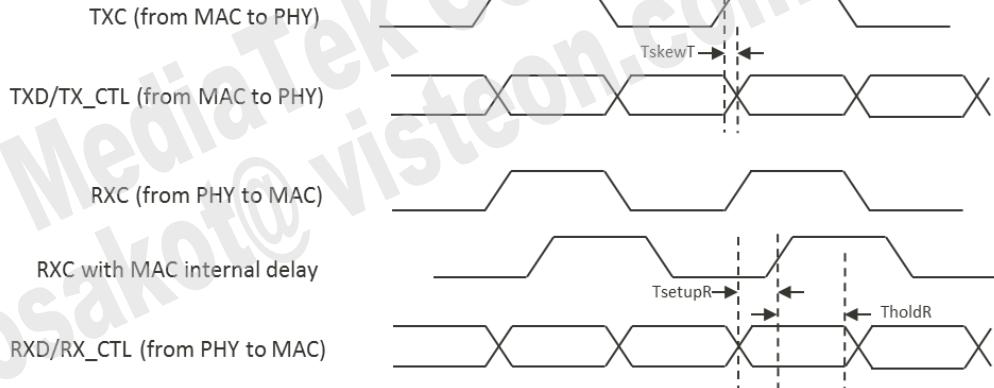
5.22.4.3 RGMII

VCC33IO_GBE should be supplied with 1.8V when 1G Ethernet RGMII is used.

Table 5-93 RGMII Interface Timing Requirement

Parameter	Symbol	Min.	Typ.	Max.	Unit
TXCLK/RXCLK period	T_{period}	7.2	8.0	8.8	ns
Data to Clock output Skew (at MAC)	T_{skewT}	-500	0	500	ps
Data to Clock input Skew (at PHY)	T_{skewR}	0.7	2	2.8	ns
TXCLK/RXCLK duty cycle	T_{dutyh}/T_{period}	45	50	55	%
Data to Clock output Setup	T_{setupT}	1.2	2	-	ns
Data to Clock output Hold	T_{holdT}	1.2	2	-	ns
Data to Clock input Setup	T_{setupR}	1	2	-	ns
Data to Clock input Hold	T_{holdR}	1	2	-	ns
TXCLK/RXCLK rising time (20~80%)	-	-	-	0.75	ns
TXCLK/RXCLK falling time (20~80%)	-	-	-	0.75	ns

RGMII



RGMII-ID

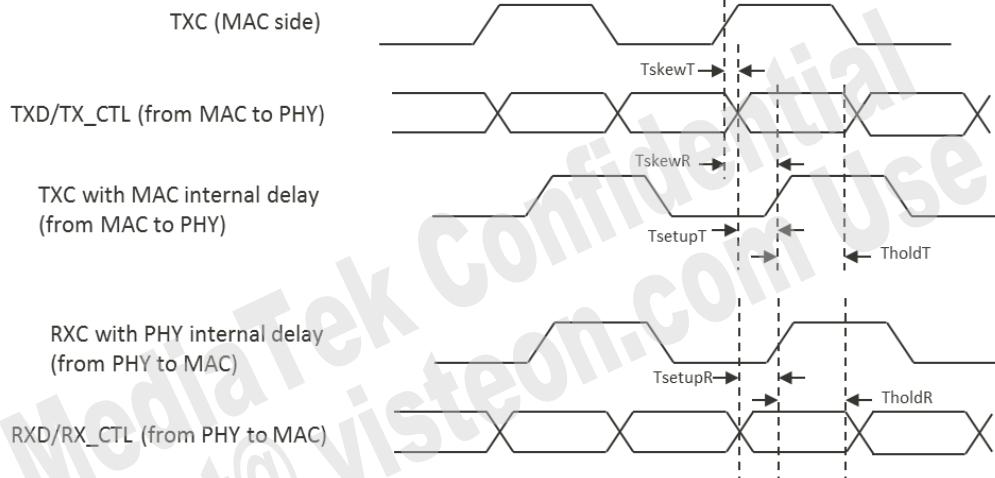


Figure 5-142 RGMII Interface Timing Diagram

5.22.4.3.1 Internal Delay Setting of RGMII Interface

1. TX side: In the following settings, the RGMII output of MT2712 ensures original RGMII and RGMII-ID Timing.

(1) Original RGMII Timing ($T_{skewT} = 0 \pm 500\text{ps}$)

Register	Address	Bit	Setting
PERI_ETH_PHY_DLY_SEL	0x10003428	Bit[6]	0: Non Inverse
		Bit[5]	0: Bypass
		Bit[4:0]	X: Disabled
PERI_REV_REG	0x10003800	Bit[1]	X: Disabled

(2) RGMII-ID Timing ($T_{setupT}/T_{holdT} = \text{Typ } 2.0, \text{ Min } 1.2\text{ns}$)

Register	Address	Bit	Setting
PERI_ETH_PHY_DLY_SEL	0x10003428	Bit[6]	0: Non Inverse 1: GTXC DLY macro
		Bit[5]	0x09: Guarantee value
		Bit[4:0]	
PERI_REV_REG	0x10003800	Bit[1]	1: Fine tuned

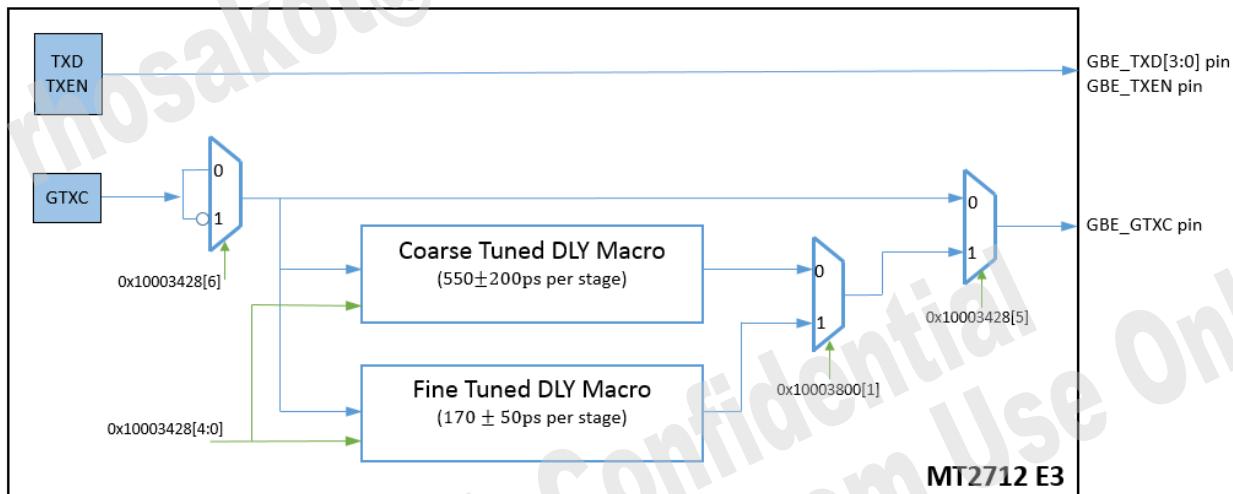


Figure 5-143 Block Diagram of Delay Macro

- a. Designed with the same delay time for TXC and TXD [3: 0]
- 2. RX side: In the following settings, the RGMII input of MT2712 ensures original RGMII and RGMII-ID Timing.
 - (1) Original RGMII Timing (TsetupR/THoldR = Min 1.0ns)

Register	Address	Bit	Setting
PERI_ETH_PHY_DLY_SEL	0x10003428	Bit[13]	0: Non Inverse
		Bit[12]	0: Bypass
		Bit[11:7]	X: Disabled
PERI_REV_REG	0x10003800	Bit[2]	X: Disabled

(2) RGMII Internal Delay Timing

Register	Address	Bit	Setting
PERI_ETH_PHY_DLY_SEL	0x10003428	Bit[13]	0: Non Inverse
		Bit[12]	1: RXC DLY macro
		Bit[11:7]	0x09: Guarantee value
PERI_REV_REG	0x10003800	Bit[2]	1: Fine tuned

5.22.4.4 MDIO

Table 5-94 MDIO Interface Timing Requirement

Parameter	Symbol	Min.	Max.	Unit
MDC falling edge to MDIO output invalid (minimum propagation delay)	S1	0	-	ns
MDC falling edge to MDIO output valid (maximum propagation delay)	S2	-	3	ns
MDIO (input) to MDC rising edge setup	S3	20	-	ns
MDIO (input) to MDC rising edge hold	S4	0	-	ns
MDC pulse width high	S5	45%	55%	MDC period
MDC pulse width low	S6	45%	55%	MDC period

*The MDC frequency should be equal to or less than 2.5MHz to be compliant with IEEE802.3 MII specification. MT2712 provides 1.58MHz MDC as default.

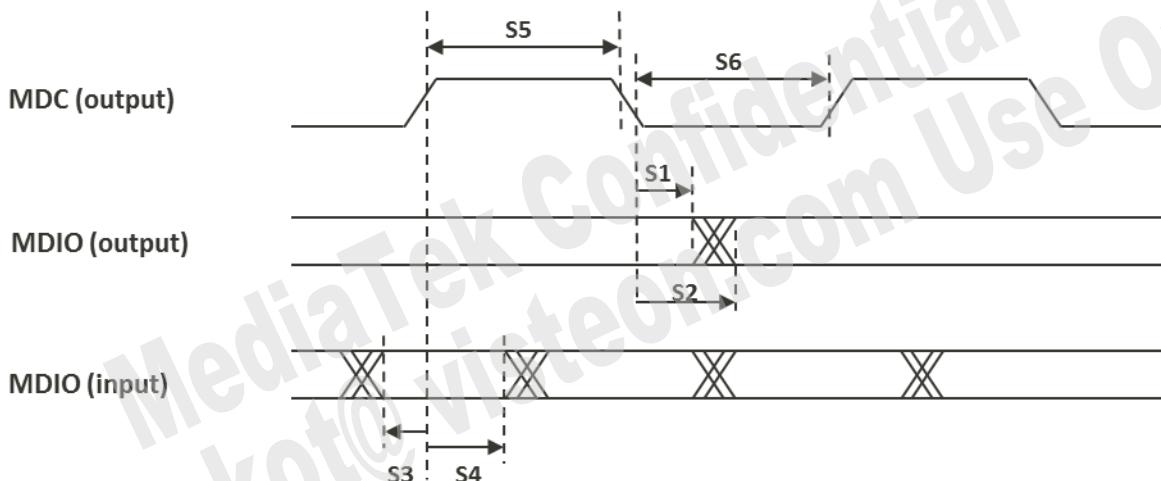


Figure 5-144 MDIO Interface Timing Diagram

5.22.5 Clocks

Table 5-95 Clock Requirements

Clock Name	Frequency (MHz)	DCM	Description
clk_tx_i	125/25/2.5	Free run	From PHY
clk_rx_i	125/25/2.5	Free run	From PHY
clk_csr_i	66.5	Free run	APB clock
clk_rmii_i	50	Free run	From PHY or CKHEN
clk_125	125	Free run	From CKGEN for RGMII
clk_ptp_ref_i	62.5	Free run	From CKGEN for PTP
ack	133	Free run	From CKGEN for BUS/DMA/MTL

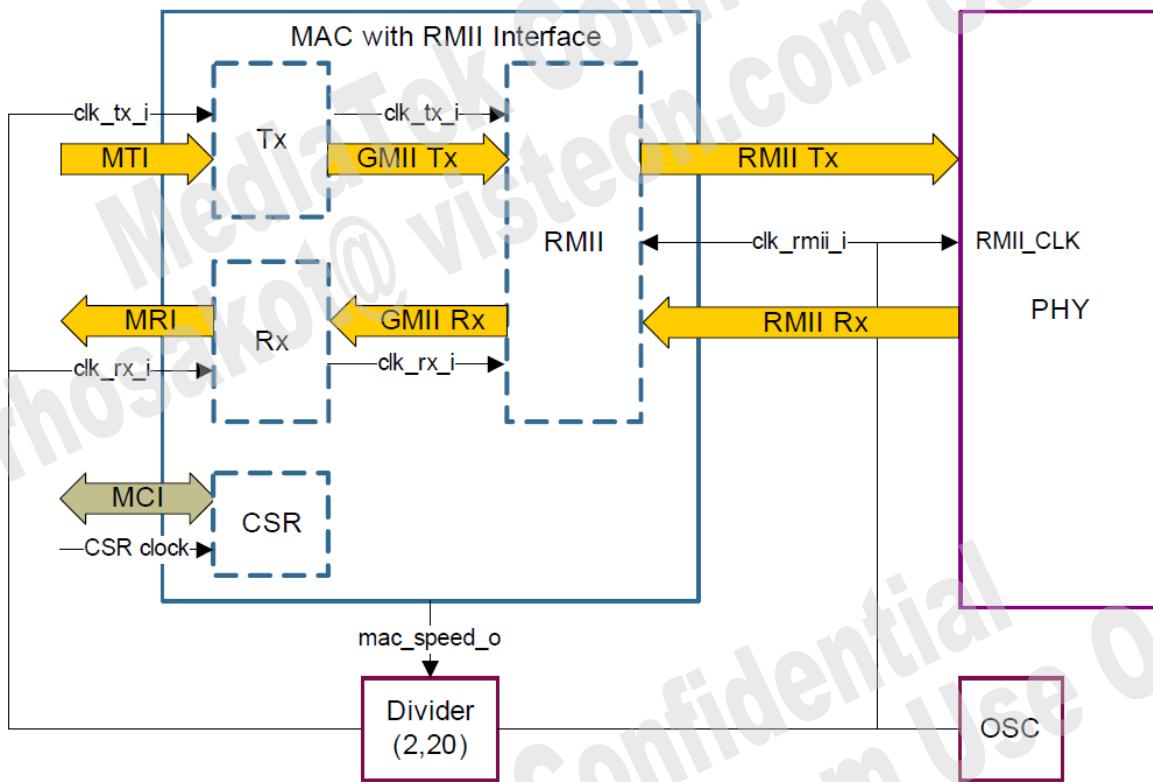


Figure 5-145 RMII Clock Structure

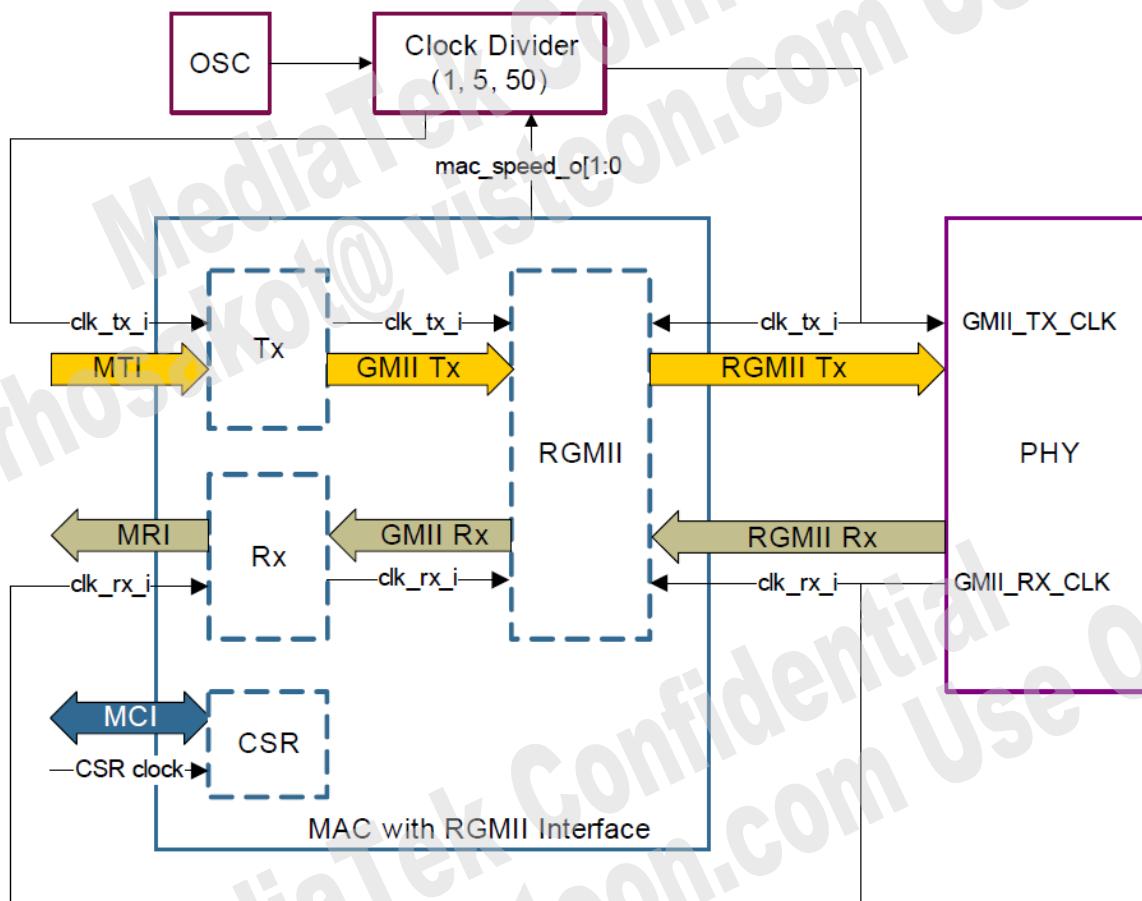


Figure 5-146 RGMII Clock Structure

5.22.6 Register Definition

For register details, please refer to Chapter 3.19 of “MT2712 IVI Application Processor Registers”.

5.22.7 Programming guide

5.22.7.1 Initializing DMA

Complete the following steps to initialize the DMA:

1. Provide a software reset. This resets all of the MAC internal registers and logic. (Bit-0 of DMA_Mode).
2. Wait for the completion of the reset process (poll Bit 0 of the DMA_Mode, which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the DMA_SysBus_Mode Register:
 - a. AAL
 - b. Fixed burst or undefined burst
 - c. Burst mode values in case of AHB bus interface, OSR_LMT in case of AXI bus interface.
 - d. If fixed length value is enabled, select the maximum burst length possible on the AXI Bus (Bits [7:1])

4. Create a descriptor list for transmit and receive. In addition, ensure that the receive descriptors are owned by DMA (set Bit 31 of descriptor TDES3/RDES3).
5. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TxDesc_Ring_Length (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RxDesc_Ring_Length (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). The ring length programmed must be at least 4.
6. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TxDesc_List_Address (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1), DMA_CH(#i)_RxDesc_List_Address (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). Also, program transmit and receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TxDesc_Tail_Pointer (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RxDesc_Tail_Pointer (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)).
7. Program the settings of the following registers for the parameters like maximum burst-length (PBL) initiated by DMA, descriptor skip lengths, OSP in case of TxDMA, RBSZ in case of RxDMA, and so on:
 - DMA_CH(#i)_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
 - DMA_CH(#i)_TX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
 - DMA_CH(#i)_RX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)
8. Enable the interrupts by programming the DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
9. Start the Receive and Transmit DMAs by setting SR (Bit 0) of the DMA_CH(#i)_RX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1) and ST (Bit 0) of the DMA_CH(#i)_TX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register
10. Repeat steps 4 to 9 for all the Tx DMA and Rx DMA channels selected in the hardware.

5.22.7.2 Initializing MTL Registers

Complete the following steps to initialize the MTL Registers:

1. Program the Tx Scheduling (SCHALG) and Receive Arbitration Algorithm (RAA) fields in MTL_Operation_Mode to initialize the MTL operation in case of multiple Tx and Rx queues.
2. Program the Receive Queue to DMA mapping in MTL_RxQ_DMA_Map0 and MTL_RxQ_DMA_Map1 registers.
3. Program the following fields to initialize the mode of operation in the MTL_TxQ0_Operation_Mode register.
 - a. Transmit Store and Forward (TSF) or Transmit Threshold Control (TTC) in case of threshold mode
 - b. Transmit Queue Enable (TXQEN) to value 2'b10 to enable Transmit Queue0
 - c. Transmit Queue Size (TQS)
4. Program the following fields to initialize the mode of operation in the MTL_RxQ0_Operation_Mode register:
 - a. Receive Store and Forward (RSF) or RTC in case of Threshold mode
 - b. Flow Control Activation and De-activation thresholds for MTL Receive FIFO (RFA and RFD)
 - c. Error Packet and undersized good Packet forwarding enable (FEP and FUP)

- d. Receive Queue Size (RQS)
5. Repeat previous two steps for all MTL Tx and Rx queues selected in the configuration.

5.22.7.3 Initializing MAC

The following MAC Initialization operations can be performed after DMA initialization. If the MAC initialization is completed before the DMA is configured, enable the MAC receiver (last step in the following sequence) only after the DMA is active. Otherwise, received frames fill the Rx FIFO and overflow.

1. Provide the MAC address registers: MAC_Address0_High and MAC_Address0_Low. If more than one MAC address is enabled in users' configuration (during configuration in the coreConsultant), program the MAC addresses appropriately.
2. Program the following fields to set the appropriate filters for the incoming frames in the MAC_Packet_Filter register:
 - a. Receive all
 - b. Promiscuous mode
 - c. Hash or Perfect Filter
 - d. Unicast, multicast, broadcast, and control frames filter settings
3. Program the following fields for proper flow control in the MAC_Q0_Tx_Flow_Ctrl register:
 - a. Pause time and other pause frame control bits
 - b. Transmit flow control bits
 - c. Flow control busy
4. Program the MAC_Interrupt_Enable register, as required.
5. Program the appropriate fields in the MAC_Configuration register. For example, the Inter-packet gap while transmission and jabber are disabled.
6. Set Bit 0 and 1 in MAC_Configuration registers to start the MAC transmitter and receiver.

5.22.7.4 Performing Normal Receive and Transmit Operation

For normal operation, complete the following steps:

1. For normal transmit and receive interrupts, read the interrupt status. Then poll the descriptors, and read the status of the descriptor owned by the Host (either transmit or receive).
2. Set appropriate values for the descriptors to ensure that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If the descriptors are not owned by the DMA (or no descriptor is available), the DMA goes into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and writing the descriptor tail pointer to Tx/Rx tail pointer register (DMA_CH[n]_TxDesc_Tail_Pointer and DMA_CH[n]_RxDesc_Tail_Pointer).
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (DMA_CH[n]_Current_App_TxDesc and DMA_CH[n]_Current_App_RxDesc).

5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (Register DMA_CH[n].Current_App_TxBuffer and DMA_CH[n].Current_App_RxBuffer).

5.22.7.5 Stopping and Starting Transmission

Complete the following steps to pause the transmission for some time:

1. Disable the Transmit DMA (if applicable) by clearing Bit 0 (ST) of DMA_CH(#i).TX_Control (for $i = 0; i <= \text{DWC_EQOS_NUM_DMA_TX_CH}-1$) Register.
2. Wait for any previous frame transmissions to complete. Users can check this by reading the appropriate bits of MTL_TxQ0_Debug Register (TRCSTS is not 01 and TXQSTS=0).
3. Disable the MAC transmitter and MAC receiver by clearing Bit (RE) and Bit 1(TE) of the MAC_Configuration Register.
4. Disable the Receive DMA (if applicable), after making sure that the data in the Rx FIFO are transferred to the system memory (by reading the appropriate bits of MTL_TxQ0_Debug Register, PRXQ=0 and RXQSTS=00).
5. Make sure that both Tx Queue and Rx Queue are empty (TXQSTS is 0 in MTL_TxQ0_Debug Register and RXQSTS is 0 in MTL_RxQ0_Debug Register).
6. To restart the operation, start the DMAs first, and then enable the MAC Transmitter and Receiver.

5.22.7.6 Programming Guidelines for Multi-Channel Multi-Queuing

5.22.7.6.1 Transmit

1. Program the Transmit queue size in the TQS field of MTL_TxQ[n].Operation_Mode register. Based on the value programmed in TQS field, the size of the queue is determined. In the Transmit operation, the number of channels is equal to the number of the queues. Therefore, the Channel to Queue mapping is fixed.
2. For a queue to be used, the queue needs to be enabled in TXQEN in the corresponding MTL_TxQ[n].Operation_Mode Register. In DMA configurations, ST bit of DMA_CH[n].Tx_Control Register and corresponding TXQEN in MTL_TxQ[n].Operation Mode Register needs to be enabled.
3. The scheduling method needs to be programmed in SCHALG of MTL_Operation_Mode register.
4. Program the MTL_TxQ[n].Quantum_Weight for generic or DCB queue as per the selected algorithm. In case of CBS algorithm in AVB queues, the MTL_TxQ[n].ETS_Control, MTL_TxQ[n].SendSlopeCredit, MTL_TxQ[n].HiCredit and MTL_TxQ[n].LoCredit registers also need to be programmed as required.
5. If DCB is enabled and PFC function is required, program MAC_TxQ_Prtv_Map0 Register to assign a fixed priority to the queue. This priority assigned is used for determining if the corresponding queue should stop transmitting packet based on the received PFC packet.

5.22.7.7 Programming Guidelines for IEEE 1588 Timestamping

5.22.7.7.1 Initialization Guideline for System Time Generation

Users can enable the timestamp feature by setting Bit 0 of the MAC_Timestamp_Control Register. However, it is essential that the timestamp counter should be initialized after this bit is set. Complete the following steps during DWC_ether_qos initialization:

1. Mask the Timestamp Trigger interrupt by clearing the Bit 16 of MAC_Interrupt_Enable Register.
2. Set Bit 0 of MAC_Timestamp_Control Register to enable timestamping.
3. Program MAC_Sub_Second_Increment Register based on the PTP clock frequency.
4. If users are using the Fine Correction approach, program MAC_Timestamp_Addend and set Bit 5 of MAC_Timestamp_Control Register.
5. Poll the MAC_Timestamp_Control Register until Bit 5 is cleared.
6. Program Bit 1 of MAC_Timestamp_Control Register to select the Fine Update method (if required).
7. Program MAC_System_Time_Seconds_Update Register and MAC_System_Time_Nanoseconds_Update Register with the appropriate time value.
8. Set Bit 2 in MAC_Timestamp_Control Register. The timestamp counter starts operation as soon as it is initialized with the value written in the Timestamp Update registers.
 - a. To enable one-step timestamping, program Bit 27 of the TDES3 Context Descriptor.
 - b. Program registers MAC_Timestamp_Ingress_Asym_Corr and MAC_Timestamp_Egress_Asym_Corr to update the correction field in PDelay_Req PTP messages.
9. Enable the MAC receiver and transmitter for proper timestamping.

System Time Correction

To synchronize or update the system time in one process (coarse correction method), complete the following steps:

1. Set the offset (positive or negative) in the Timestamp Update registers (MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update).
2. Set Bit 3 (TSUPDT) of the MAC_Timestamp_Control Register. The value in the Timestamp Update registers is added to or subtracted from the system time when the TSUPDT bit is cleared.

To synchronize or update the system time to reduce system-time jitter (fine correction method), complete the following steps:

1. With the help of the algorithm explained in “System Time Register Module”, calculate the rate by which users want to make the system time increment slower or faster.
2. Update the MAC_Timestamp_Addend with the new value and set Bit 5 of the MAC_Timestamp_Control Register.
3. Wait for the time during which users want the new value of the Addend register to be active. Users can do this by enabling the Timestamp Trigger interrupt after the system time reaches the target value.

4. Program the required target time in MAC_PPS[n]_Target_Time_Seconds Register and MAC_PPS[n]_Target_Time_Nanoseconds Register.
5. Enable the Timestamp interrupt in Bit 12 of MAC_Interrupt_Enable register.
6. Set Bit 4 in Register MAC_Timestamp_Control.
7. When this trigger causes an interrupt, read MAC_Interrupt_Status Register.
8. Reprogram MAC_Timestamp_Addend Register with the old value and set Bit 5 again.

6 Multimedia System

6.1 Audio System

6.1.1 Introduction

The audio system provides the ability to exchange audio data. The interfaces are listed as follows:

- Inter-IC Sound (I2S) input/output interface
- Master 2-ch I2S output ×4 (the fourth I2SO only has data pin output, and shares clock pin with the first I2SO). I2S output can be configured to multi-channel (7.1 or 5.1+stereo)
- Master/Slave 2-ch I2S input ×3
- Master/Slave Pulse-Code Modulation (PCM) interface with Sample Rate Converter (SRC) ×1 (can be configured to I2S mode)
- Master Time-Division Multiplexing (TDM) output interface ×2
- Master/Slave TDM input interface ×1 (shares pin with I2S input)
- PCM/I2S merged interface for MTK connectivity IC ×1
- Internal Audio Analog-Digital Converter (ADC) ×2
- 2-ch memory base SRC ×5
- 8-ch I2S-IQ interface ×3 (slave mode only, can be configured to I2S/TDM/PCM/DSP/RJ/LJ/EIAJ format)
- Support audio safety function (I2S/TDM Cyclic Redundancy Check (CRC))

6.1.2 Features

- Audio Internal ADC
 - Support recording at 8, 16, 32, 48 KHz sample rate for Analog Microphone (MIC)
 - Support stereo recording
- I2S
 - Support 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 88.2, 96, 176.4, 192, 352.8 and 384 KHz sample rate
 - Support master/slave input mode
 - Support master output mode and can support multi-channel output
 - Support the function of I2S input and I2S output sharing clock
 - Support 16/24/32-bit stereo data
 - Support EIAJ/RJ/LJ/I2S format

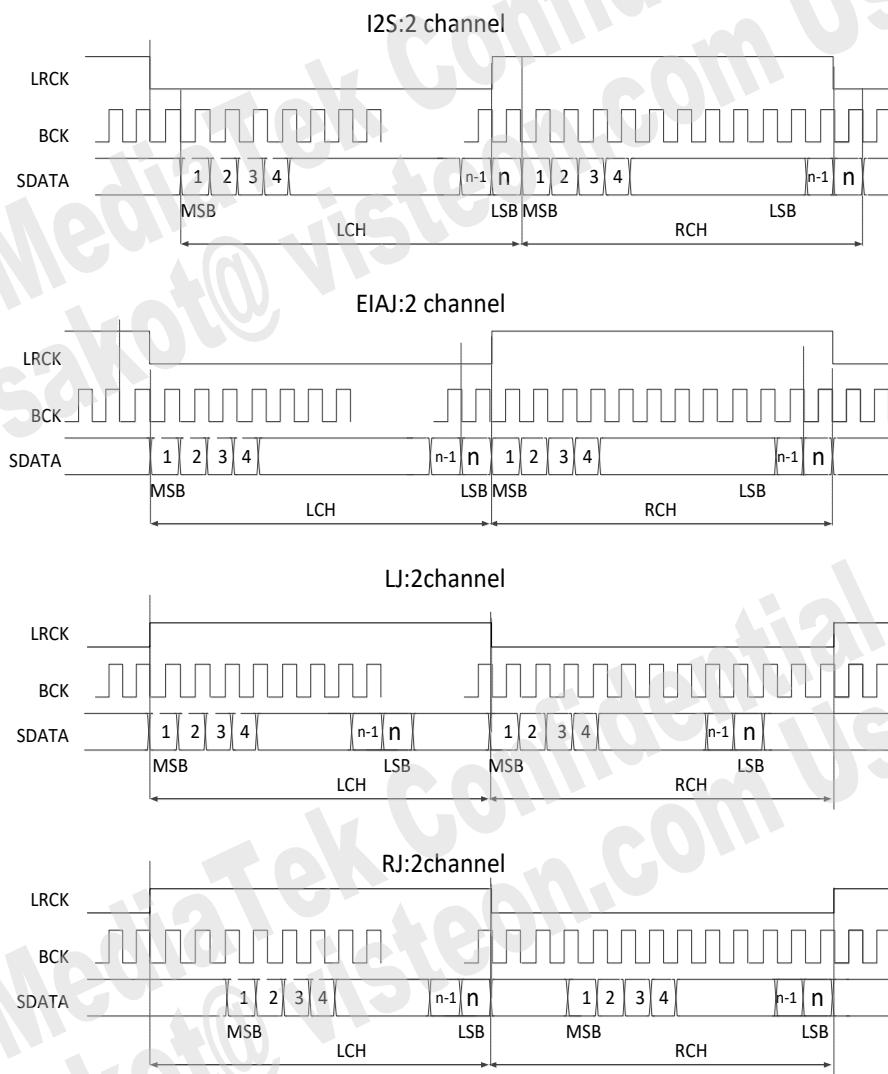


Figure 6-1 I2S Protocol

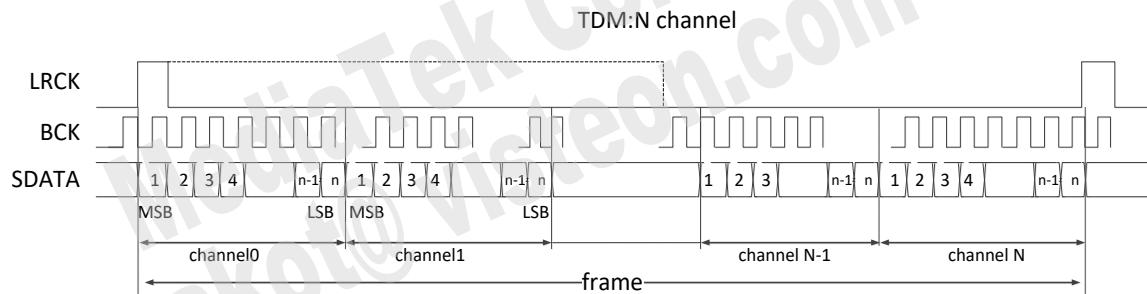
- TDM
 - Master operation supported for TDMOUT and master/slave operation supported for TDMIN.
 - Support TDM/I2S protocol
 - Word length is programmable for 16/32 bits
 - LRCK and Bit Clock (BCK) can be inverted before being output to the slave device
 - Input data (up to 16 channels) can swap to each other
 - Support 8/16/24/32-bit stereo data (only 16/32-bit for slave mode)
 - Support 2/4/8/12/16 channels (only 2/4/8 channels for slave mode)
 - Changeable LRCK width
 - The BCK speed faster than 24.576 MHz is not recommend.
 - Supported sample rates are as follows:

Table 6-1 TDM Sample Rates

TDM OUT (v: support x: not support)

APLL1(KHz)		196608.00									
APLL2(KHz)		180633.60									
TDM IN Scenarios	BCLK clock rate table										
-	Num of CH	16	16	12	12	8	8	4	4	2	2
-	Sample bitwidth	16	32	16	32	16	32	16	32	16	32
Sample rate (KHz)	8.00	v	v	v	v	v	v	v	v	v	v
	16.00	v	v	v	v	v	v	v	v	v	v
	24.00	v	v	x	x	v	v	v	v	v	v
	32.00	v	x	v	v	v	v	v	v	v	v
	44.10	v	x	x	x	v	v	v	v	v	v
	48.00	v	x	x	x	v	v	v	v	v	v
	88.20	x	x	x	x	v	x	v	v	v	v
	96.00	x	x	x	x	v	x	v	v	v	v
	192.00	x	x	x	x	x	x	v	x	v	v
TDM OUT Scenarios	BCLK clock rate table										
-	Num of CH	16	16	12	12	8	8	4	4	2	2
-	Sample bitwidth	16	32	16	32	16	32	16	32	16	32
Sample rate (KHz)	8.00	v	v	v	v	v	v	v	v	v	v
	16.00	v	v	v	v	v	v	v	v	v	v
	24.00	v	v	x	x	v	v	v	v	v	v
	32.00	v	v	v	v	v	v	v	v	v	v
	44.10	v	v	x	x	v	v	v	v	v	v
	48.00	v	v	x	x	v	v	v	v	v	v
	88.20	v	x	x	x	v	v	v	v	v	v
	96.00	v	x	x	x	v	v	v	v	v	v
	192.00	x	x	x	x	v	x	v	v	v	v

TDM one SDATA mode



TDM four SDATA modes

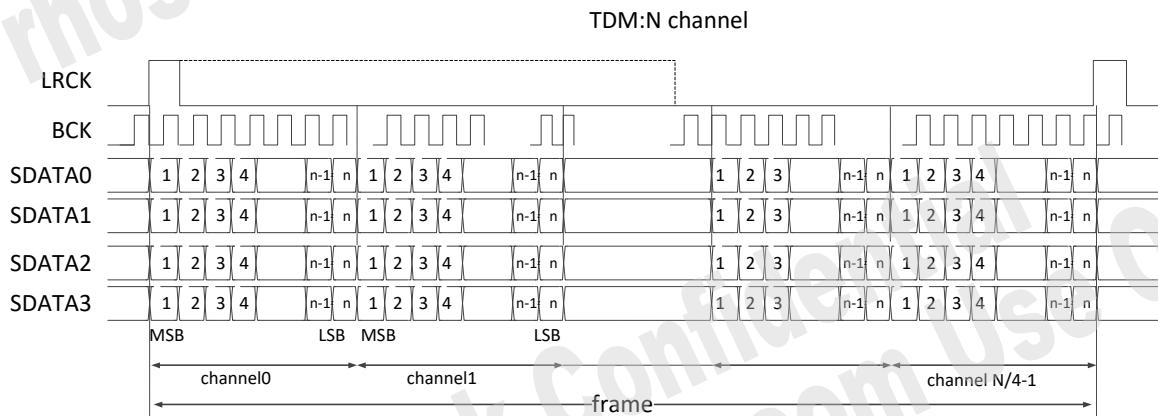
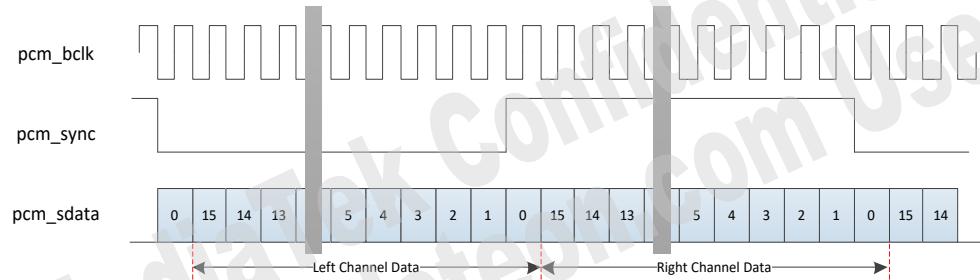


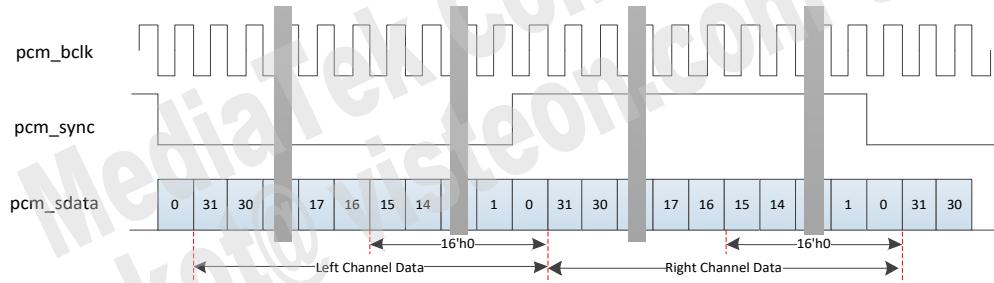
Figure 6-2 TDM Protocol

- PCM: The PCM interface can operate as an interface for transfer of I2S/EIAJ/mode A/mode B format by specifying the data format control register.
 - Support master/slave mode with SRC
 - Support FS: 8/16/32/48 KHz
 - Support I2S/EIAJ/mode A/mode B format
 - Support 16/24 bits precision in PCM data path
 - Support mode A/mode B sync period 32/64 BCK cycles
 - I2S/EIAJ format LRCK width support 32/64 BCK cycles
 - Support 32/64 BCK cycles per frame sync (word length = 16/32 BCK cycles)

16-bit word length in 16-bit channel length of I2S format



16-bit word length in 32-bit channel length of I2S format



24-bit word length in 32-bit channel length of I2S format

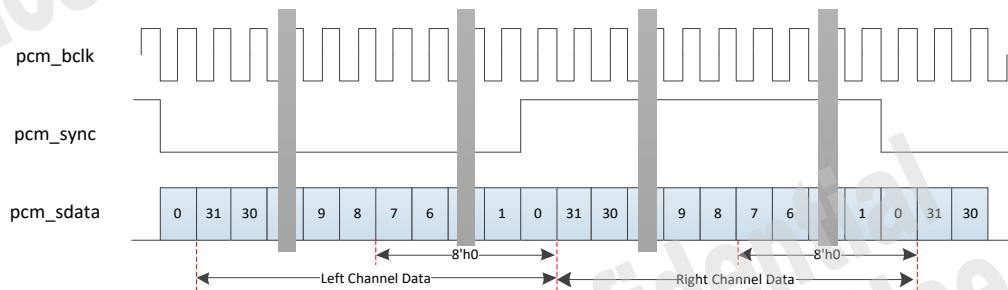
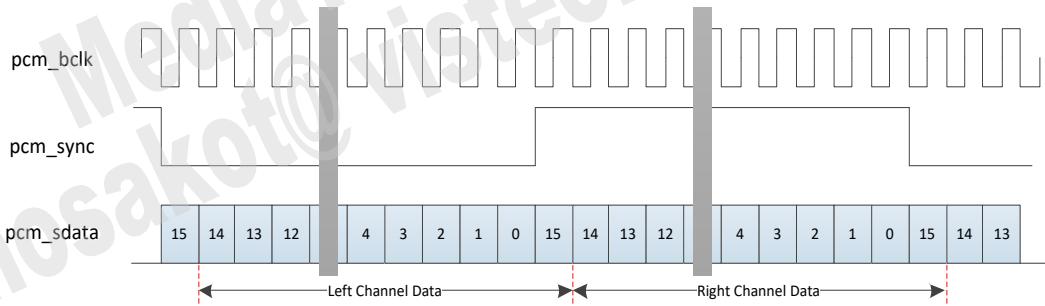
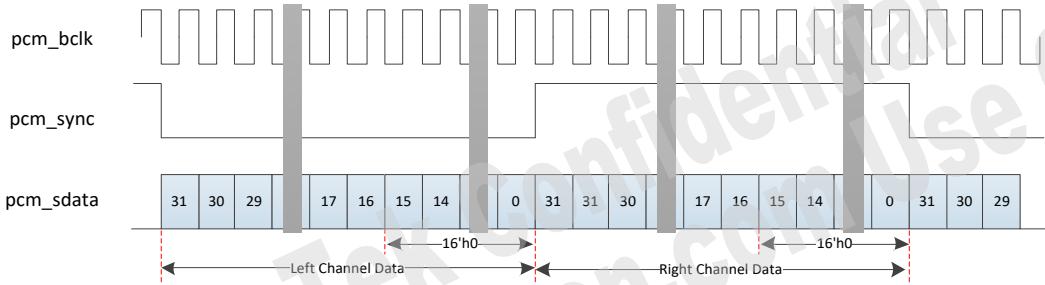


Figure 6-3 Waveform of I2S Format in PCM Interface

16-bit word length in 16-bit channel length of EIAJ format



16-bit word length in 32-bit channel length of EIAJ format



24-bit word length in 32-bit channel length of EIAJ format

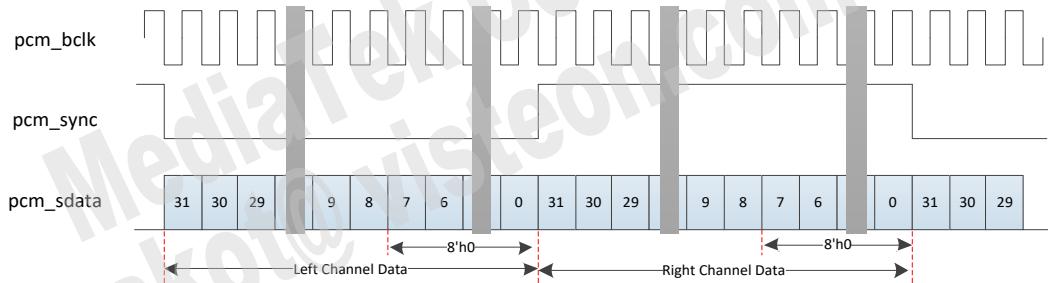
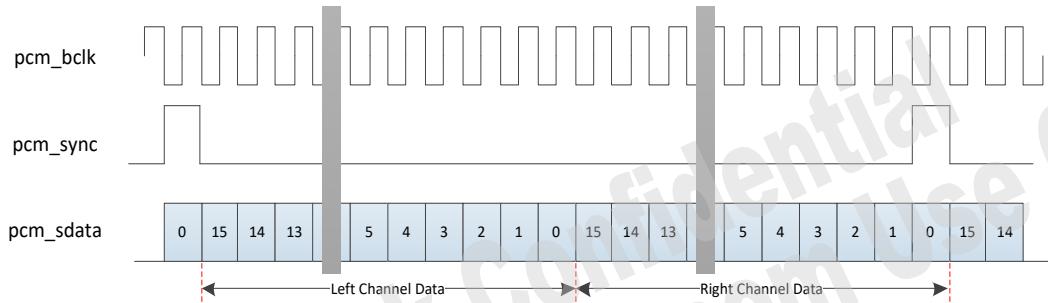
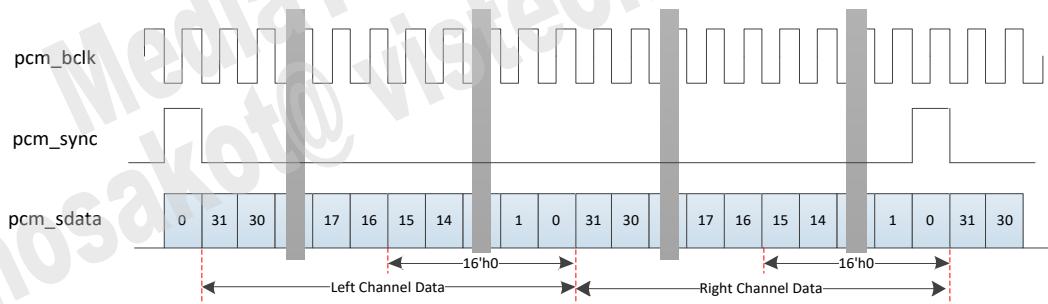


Figure 6-4 Waveform of EIAJ Format in PCM Interface

16-bit word length in 16-bit channel length of mode A format



16-bit word length in 32-bit channel length of mode A format



24-bit word length in 32-bit channel length of mode A format

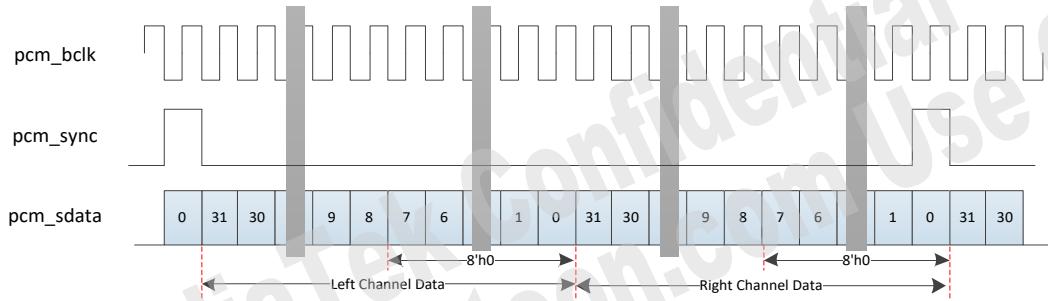
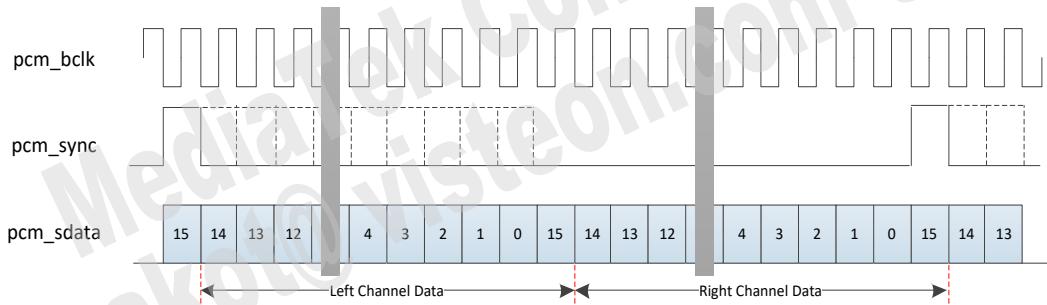
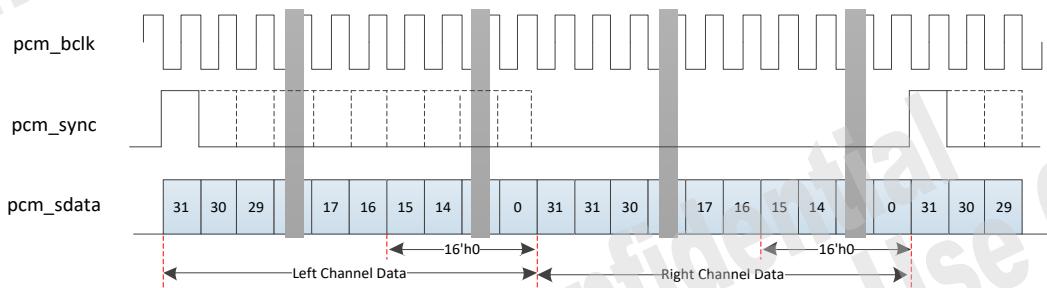


Figure 6-5 Waveform of Mode A Format in PCM Interface

16-bit word length in 16-bit channel length of mode B format



16-bit word length in 32-bit channel length of mode B format



24-bit word length in 32-bit channel length of mode B format

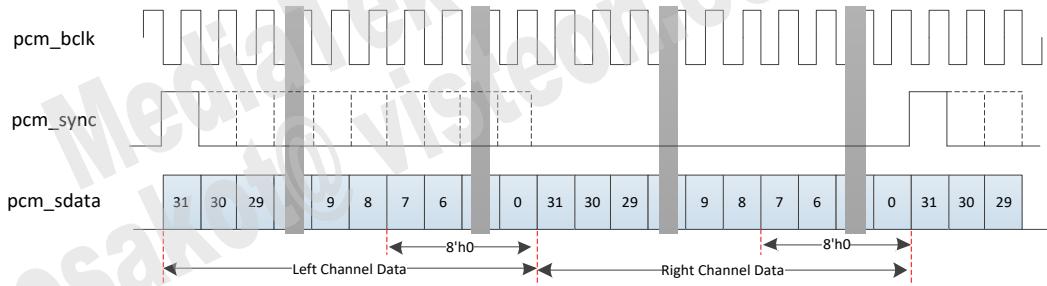


Figure 6-6 Waveform of Mode B Format in PCM Interface

- PCM/I2S merged interface
 - 4-pin interface for concurrently supporting I2S and PCM
 - PCM supports 8/16 KHz sample rate
 - I2S supports 32, 44.1, and 48 KHz sample rate
- ASRC
 - Input/output interface is memory. Use input buffer point and output buffer point to control system running
 - Support one stereo channel-set per Asynchronous Sample Rate Converter (ASRC) HW
 - Each channel-set can configure its input frequency and output frequency separately
 - The total system can have up to four 24-bit representable Fs simultaneously

- $1/16x < \text{Conversion_Ratio} < 8x$ (for anti-alias filter with good performance, better $1/4x < \text{Conversion_Ratio} < 8x$)
- Data come from and write to Dynamic Random-Access Memory (DRAM)
- 16/32-bit input
- 16/32-bit output
- THD+N
- About 150 dB (on average)
- Hardware gain function with higher resolution to enhance the audio quality and flexibility of interconnection
- Flexible interconnection system to make data exchange between interfaces without intervention of Central Processing Unit (CPU)
- I2S-IQ
 - Only support slave operation mode
 - Support TDM/I2S/PCM/DSP/LJ/RJ/EIAJ protocols
 - Word length is programmable for 8~32 bits
 - LRCK and BCK can be inverted before being input to I2S-IQ
 - Support 2/4/8 channels
 - Sample rate is decided by master device
 - I2S-IQ supports protocols shown in Figure 6-7 to Figure 6-14

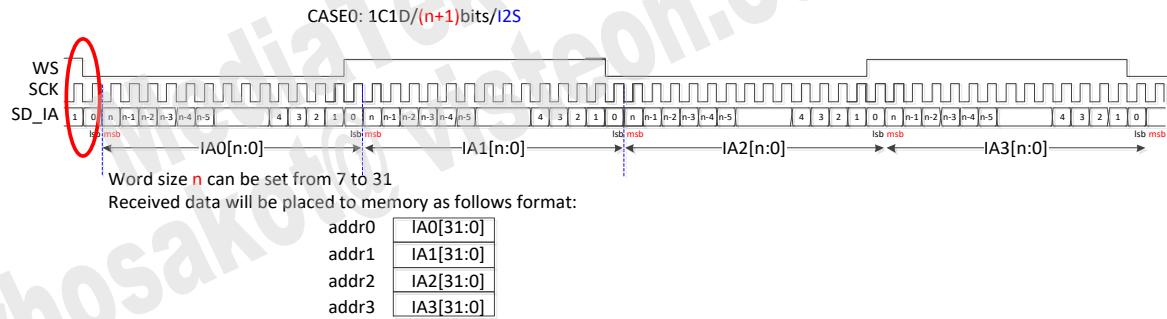


Figure 6-7 Waveform of CASE0 Format in I2S-IQ Interface

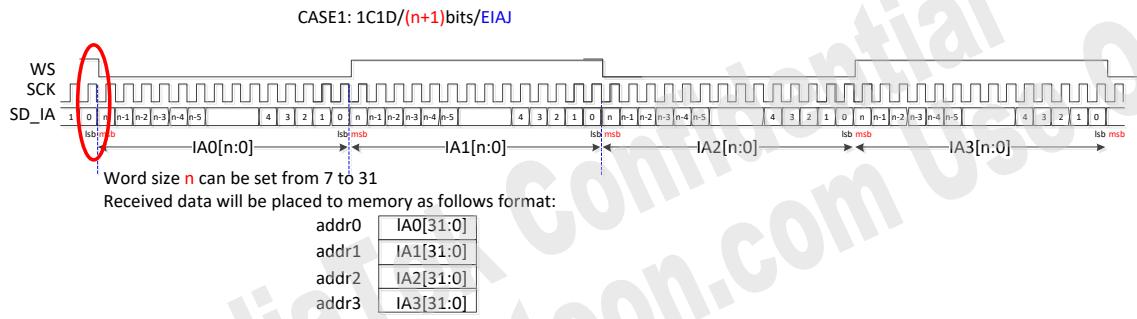


Figure 6-8 Waveform of CASE1 Format in I2S-IQ Interface

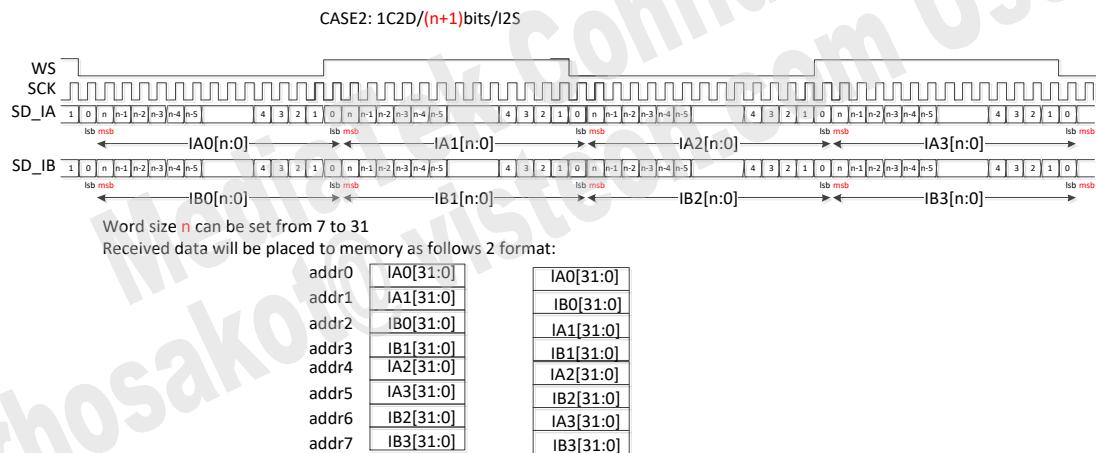


Figure 6-9 Waveform of CASE2 Format in I2S-IQ Interface

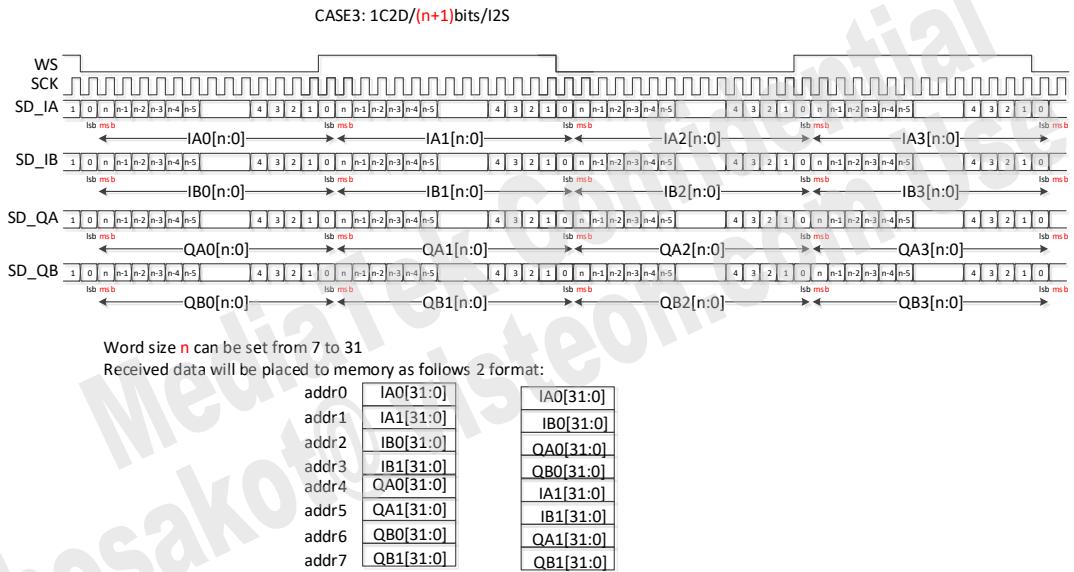


Figure 6-10 Waveform of CASE3 Format in I2S-IQ Interface

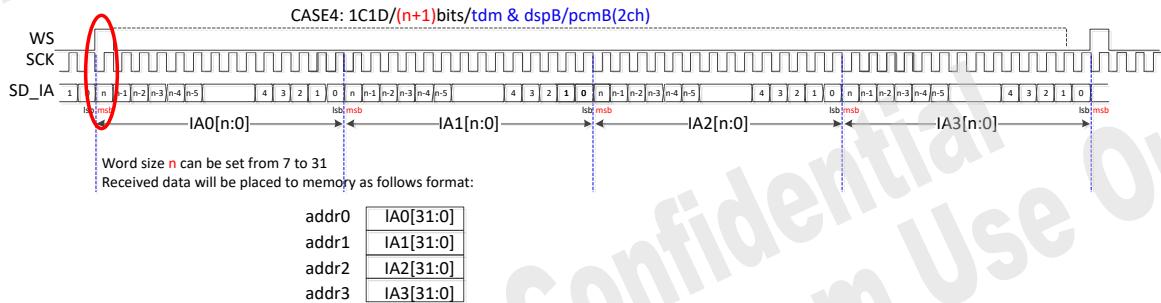


Figure 6-11 Waveform of CASE4 Format in I2S-IQ Interface

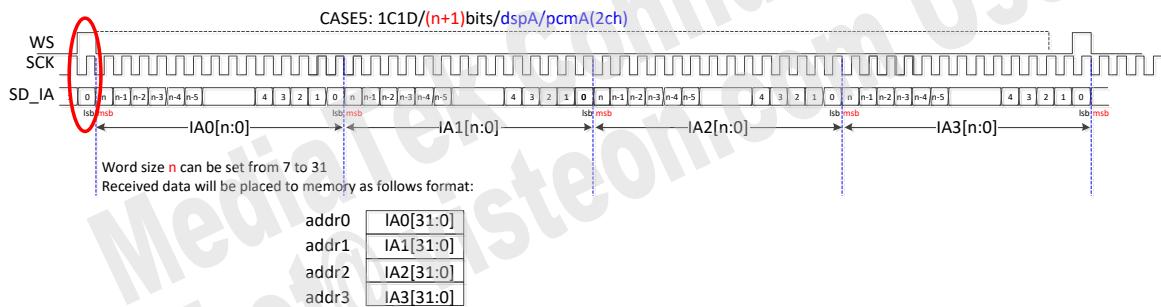


Figure 6-12 Waveform of CASE5 Format in I2S-IQ Interface

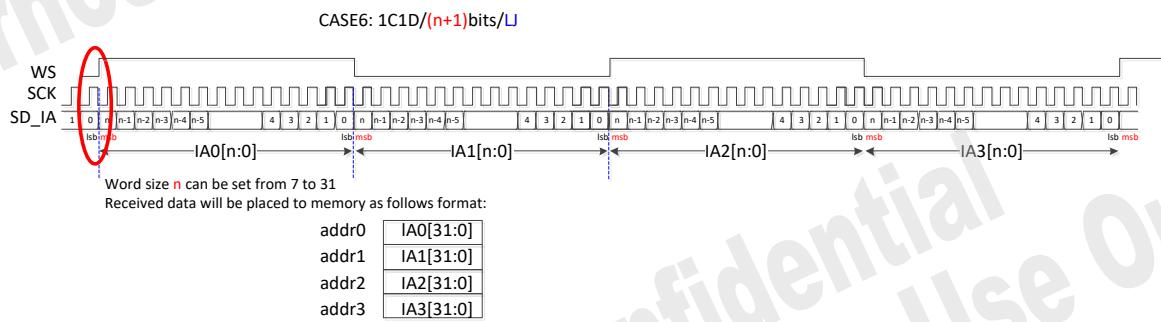


Figure 6-13 Waveform of CASE6 Format in I2S-IQ Interface

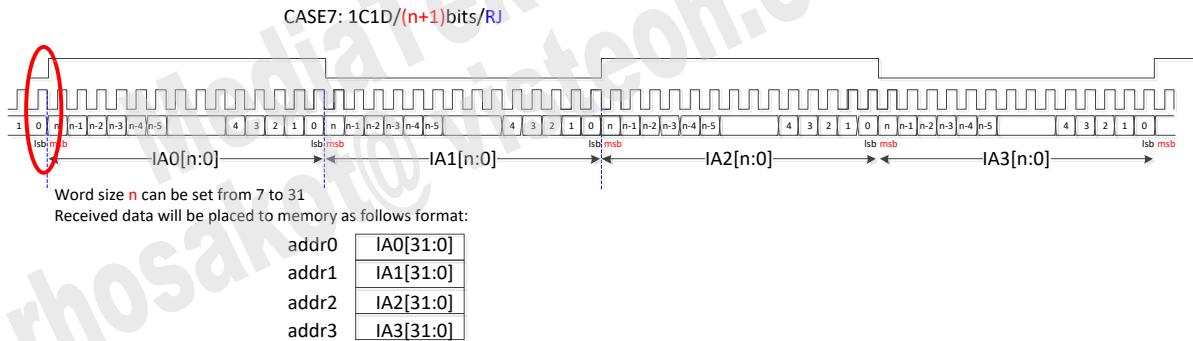


Figure 6-14 Waveform of CASE7 Format in I2S-IQ Interface

- Audio safety function (I2S/TDM CRC Check)

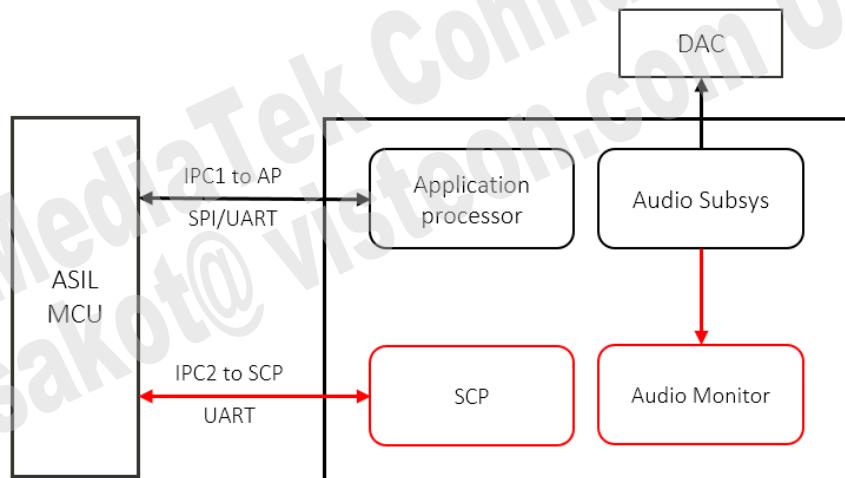


Figure 6-15 Audio Safety Feature Diagram

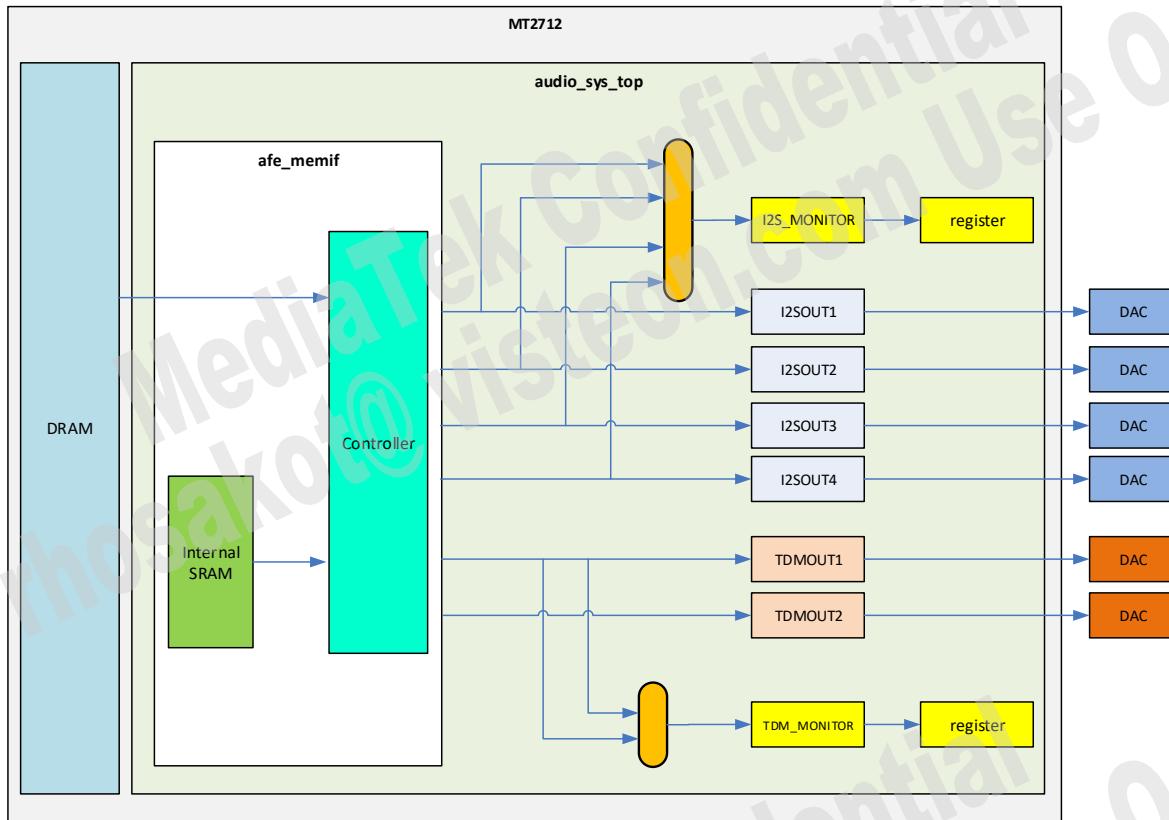


Figure 6-16 Audio Safety Feature Diagram

- Audio safety function is shown as Figure 6-15 and Figure 6-16 (i2s_monitor and tdm_monitor)
- Audio monitor calculates CRC of the output PCM data of the specific audio output channel, i.e., in one of the I2S ports or one of the TDM output ports. CRC is calculated using 32-bit polynomial with an initial CRC Sum of 32'hFFFFFFF:

$$\text{crc} = 1+x^1+x^2+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16}+x^{22}+x^{23}+x^{26}+x^{32}.$$

- The I2S or TDM output ports for warning sound playback shall be configured to a special monitor-mode.
- The audio PCM data shall be padding to 32-bit with 16/24 MSB bits for audio sample and two LSB bits for enabling the CRC on the audio sample and restarting flag (as shown in Figure 6-17)
- When the warning sound is going to play out, ASIL MCU first resets CRC restart counter, then reads back the CRC result and CRC restart counter at a proper time to compare the CRC result with pre-defined signature.

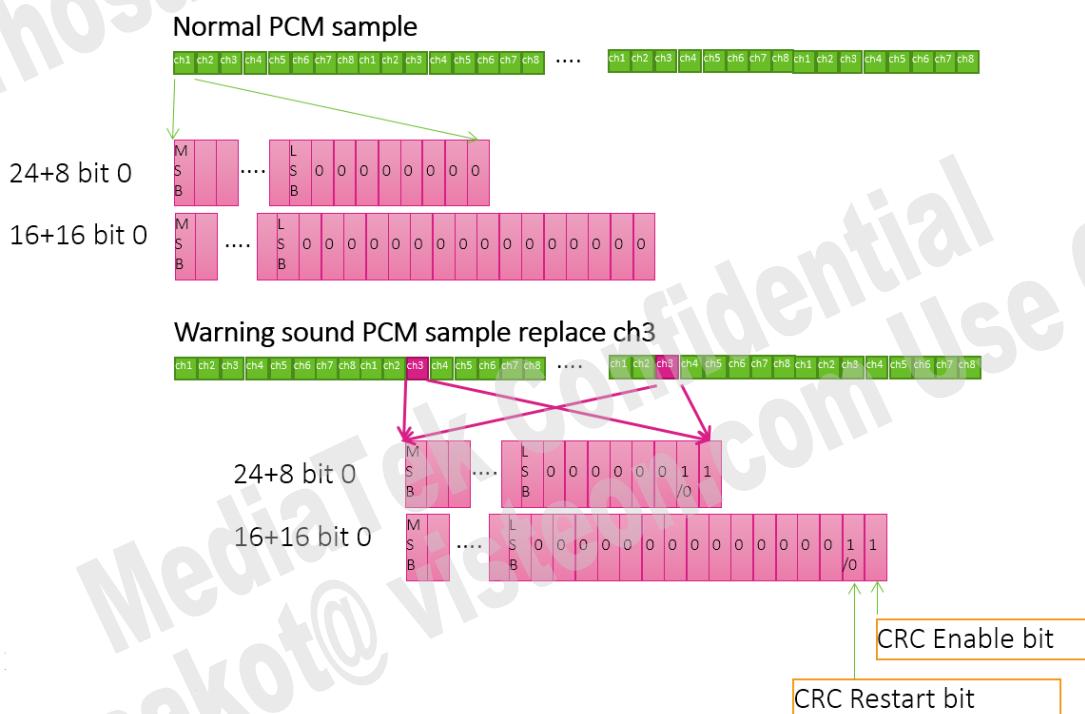


Figure 6-17 PCM Format of Audio Warning Sound

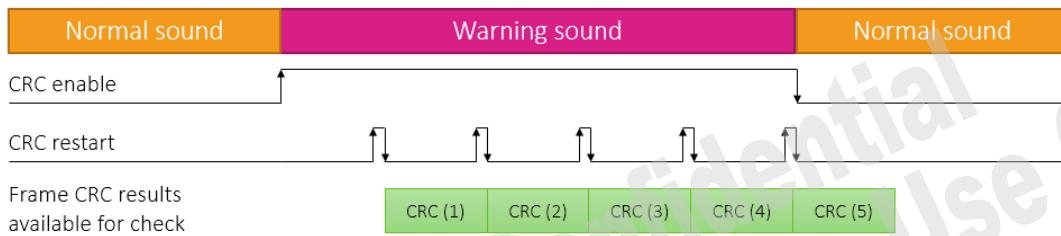


Figure 6-18 CRC of Audio Warning Sound

- Once HW CRC is enabled, HW will update CRC/counter automatically.
- When the rising edge of CRC enable comes, counter will be reset by HW.

- When the rising edge of CRC restart comes, CRC result & counter will be updated by HW.

6.1.3 Block Diagram

Figure 6-19 shows the flexible interconnection between audio interfaces.

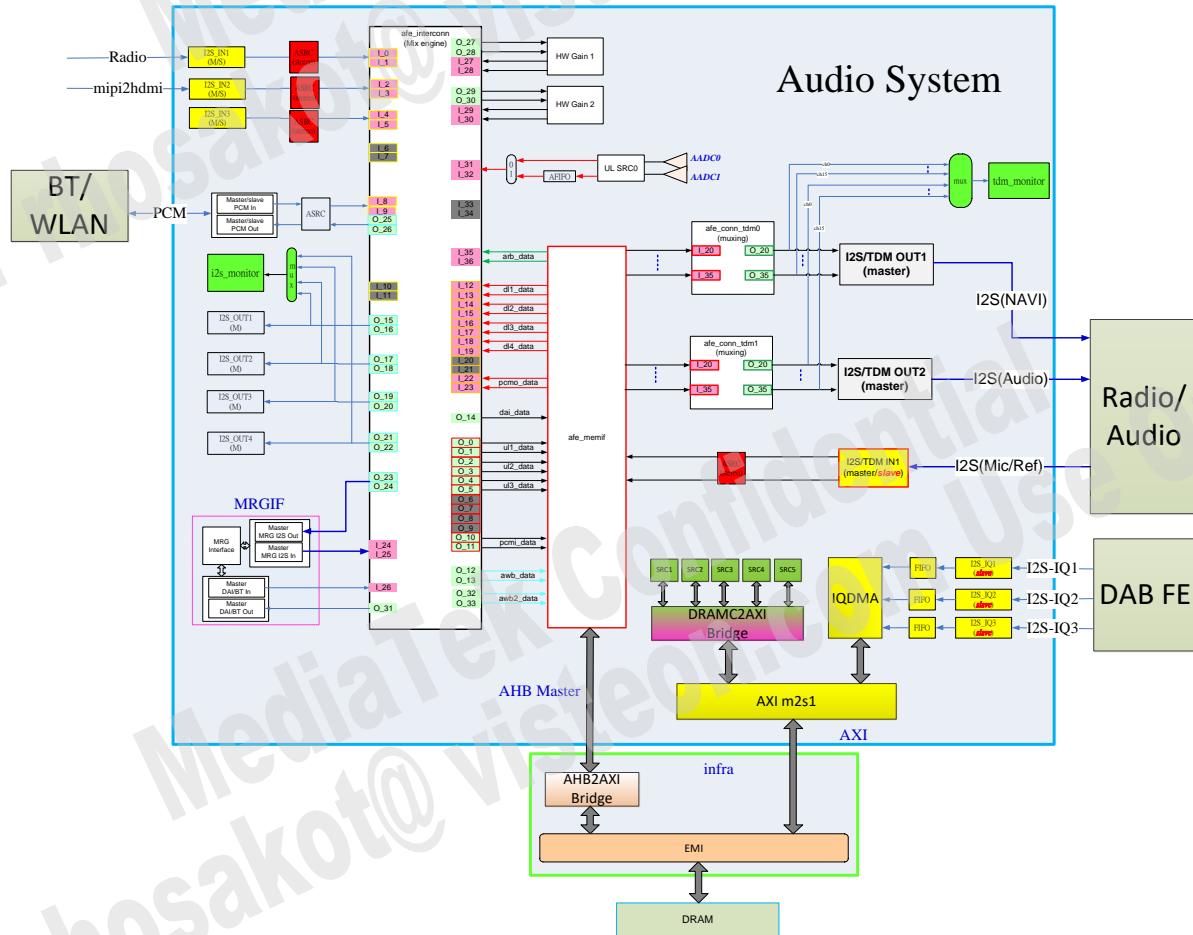


Figure 6-19 Audio System Block Diagram

6.1.4 Audio AC Timing

6.1.4.1 Audio Electrical Characteristics

I2S AC Electrical Characteristics:

Table 6-2 I2S AC Timing Characteristics

Parameter	Description	Min.	Typ.	Max.	Unit
FS(I2S)	Sampling frequency	8	-	192	kHz
tws	Word select period	$32 * (1 / f_{BCK})$	-	$64 * (1 / f_{BCK})$	ns
fMCK	Master clock frequency	0.768	-	24.576	MHz

Parameter	Description	Min.	Typ.	Max.	Unit
f_{BCK}	Serial clock frequency	$32 * FS(I2S)$	-	$64 * FS(I2S)$	KHz
t_{BCK_H}	BCK high-level time	-	$0.5 * (1 / f_{BCK})$	-	ns
t_{BCK_L}	BCK low-level time	-	$0.5 * (1 / f_{BCK})$	-	ns
t_{V_DO}	Output Data valid time	0	-	10	ns
t_{V_WS}	Output LRCK valid time	0	-	10	ns
$t_{(LS)}$	LRCK setup time to BCK rising edge	10	-	-	ns
$t_{(LH)}$	LRCK hold time to BCK rising edge	10	-	-	ns
$t_{(DS)}$	DATA setup time to BCK rising edge	10	-	-	ns
$t_{(DH)}$	DATA setup time to BCK rising edge	10	-	-	ns

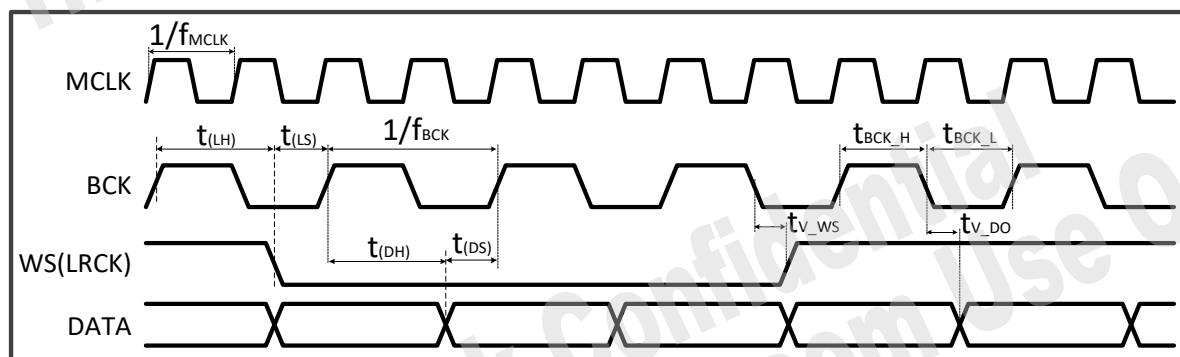


Figure 6-20 I2S Mode Timing Diagram

TDM AC Electrical Characteristics:

Table 6-3 TDM AC Timing Characteristics

Parameter	Description	Min.	Typ.	Max.	Unit
FS	Sampling frequency	8	-	192	kHz
t_{ws}	Word select period	$32 * (1 / f_{BCK})$	-	$512 * (1 / f_{BCK})$	ns
f_{MCK}	Master clock frequency	0.768	-	24.576	MHz
f_{BCK}	Serial clock frequency	0.256	-	12.288	MHz
t_{BCK_H}	BCK high-level time	-	$0.5 * (1 / f_{BCK})$	-	ns
t_{BCK_L}	BCK low-level time	-	$0.5 * (1 / f_{BCK})$	-	ns
t_{V_DO}	Output Data valid time	0	-	10	ns
t_{V_WS}	Output LRCK valid time	0	-	10	ns
$t_{(LS)}$	LRCK setup time to BCK rising edge	10	-	-	ns
$t_{(LH)}$	LRCK hold time to BCK rising edge	10	-	-	ns
$t_{(DS)}$	DATA setup time to BCK rising edge	10	-	-	ns
$t_{(DH)}$	DATA setup time to BCK rising edge	10	-	-	ns

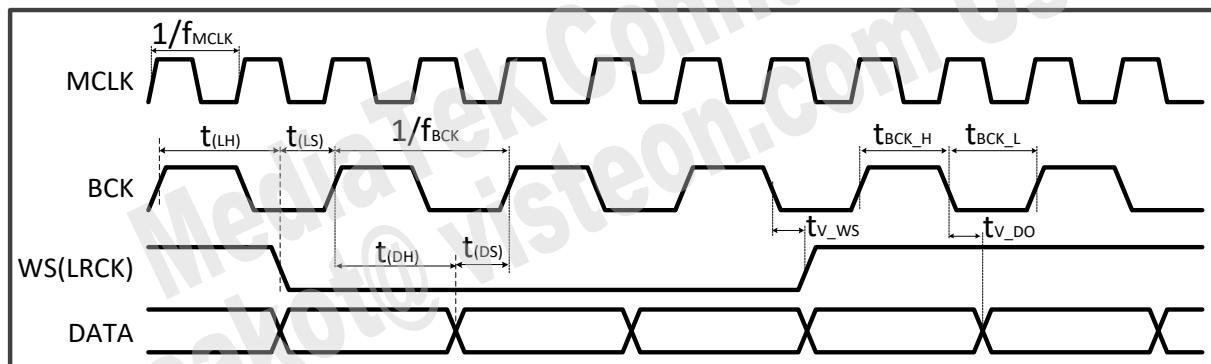


Figure 6-21 TDM Mode Timing Diagram

PCM AC Electrical Characteristics:

Table 6-4 PCM AC Timing Characteristics

Parameter	Description	Min.	Typ.	Max.	Unit
FS	Sampling frequency	8	-	48	kHz
t_{ws}	Word select period	$32 * (1 / f_{BCK})$	-	$64 * (1 / f_{BCK})$	ns
f_{MCK}	Master clock frequency	0.768	-	24.576	MHz
f_{BCK}	Serial clock frequency	$32 * FS$	-	$64 * FS$	kHz
t_{BCK_H}	BCK high-level time	-	$0.5 * (1 / f_{BCK})$	-	ns
t_{BCK_L}	BCK low-level time	-	$0.5 * (1 / f_{BCK})$	-	ns
t_{v_do}	Output Data valid time	0	-	10	ns
t_{v_ws}	Output LRCK valid time	0	-	10	ns
$t_{(LH)}$	LRCK setup time to BCK rising edge	10	-	-	ns
$t_{(LS)}$	LRCK hold time to BCK rising edge	10	-	-	ns
$t_{(DS)}$	DATA setup time to BCK rising edge	10	-	-	ns
$t_{(DH)}$	DATA setup time to BCK rising edge	10	-	-	ns

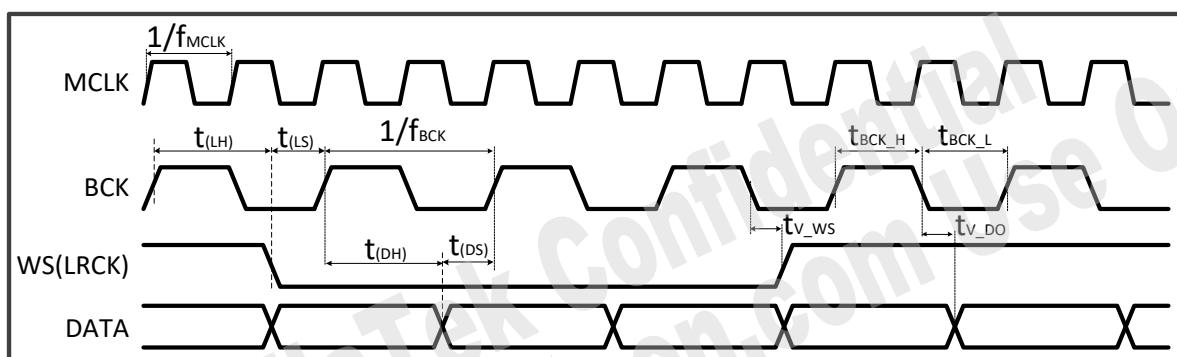


Figure 6-22 PCM Mode Timing Diagram

I2S-IQ AC Electrical Characteristics:

Table 6-5 I2S-IQ AC Timing Characteristics

Parameter	Description	Min.	Typ.	Max.	Unit
FS	Sampling frequency	8	-	—	kHz
t _{ws}	Word select period	32*(1 / f _{BCK})	-	256*(1 / f _{BCK})	ns
f _{BCK}	Serial clock frequency	-	-	65	MHz
t _{BCK_H}	BCK high-level time	-	0.5*(1 / f _{BCK})	-	ns
t _{BCK_L}	BCK low-level time	-	0.5*(1 / f _{BCK})	-	ns
t _(LS)	LRCK setup time to BCK rising edge	10	-	-	ns
t _(LH)	LRCK hold time to BCK rising edge	10	-	-	ns
t _(DS)	DATA setup time to BCK rising edge	10	-	-	ns
t _(DH)	DATA setup time to BCK rising edge	10	-	-	ns

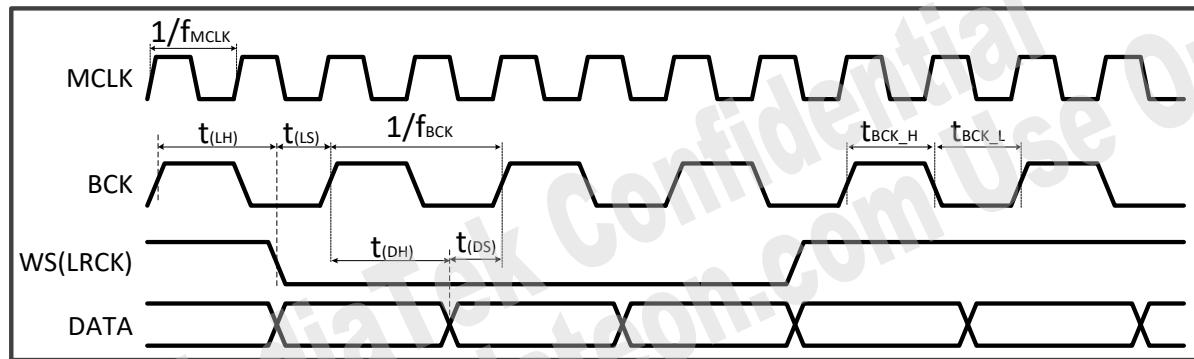


Figure 6-23 I2S-IQ Slave Mode Timing Diagram

6.1.5 Audio Clock Structure

Audio system clock structure is shown as Figure 6-24. There are mainly two APPLLs for audio interface (APPLL1 for 48KHz domain and APPLL2 for 44.1KHz domain).

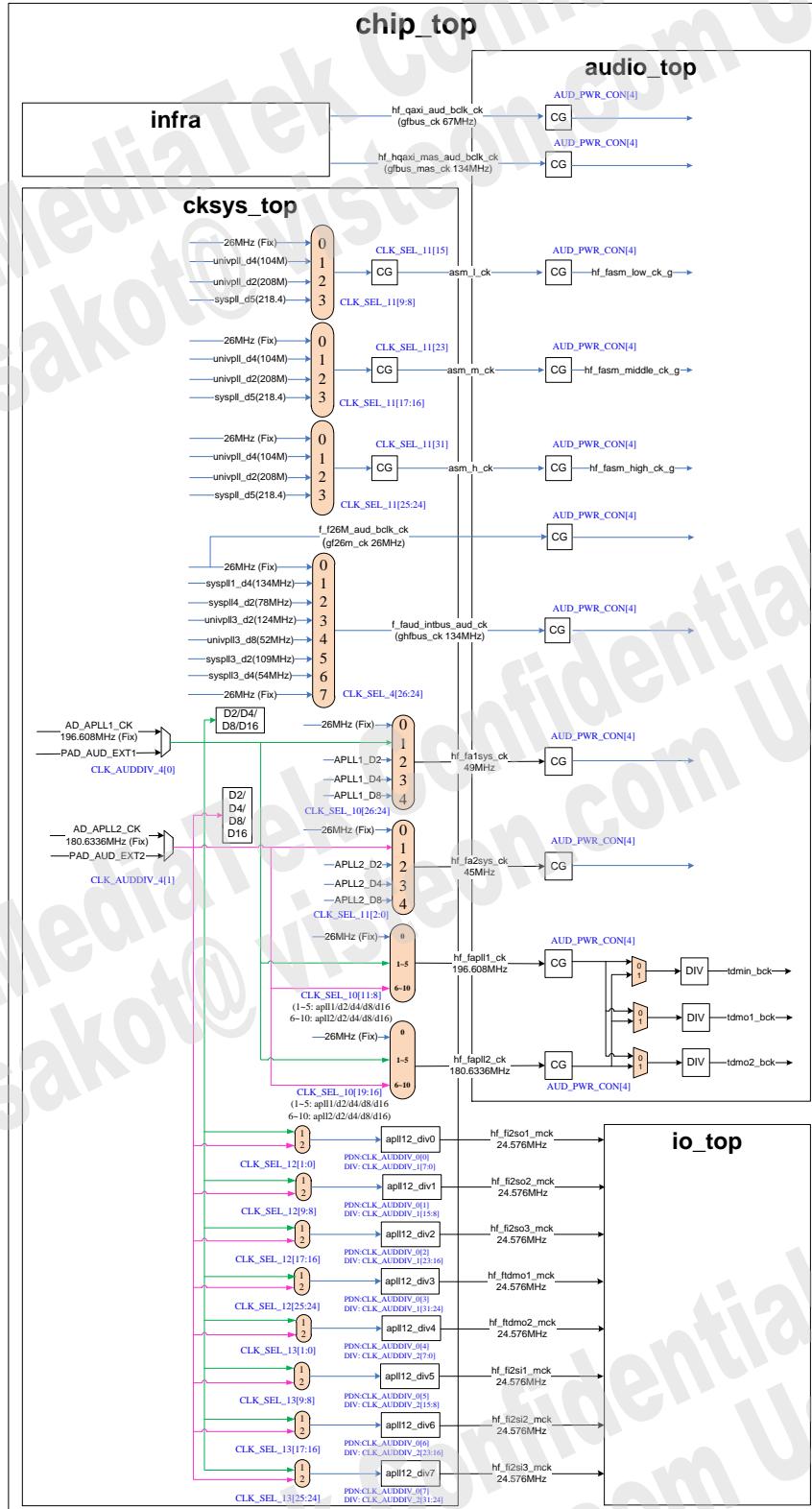


Figure 6-24 Audio System Clock Structure

6.1.6 Interface Timing

I2S interface timing

- In the audio system, there are four I2S outputs and three I2S inputs of I2S interfaces. All the IO timing information are shown as below:

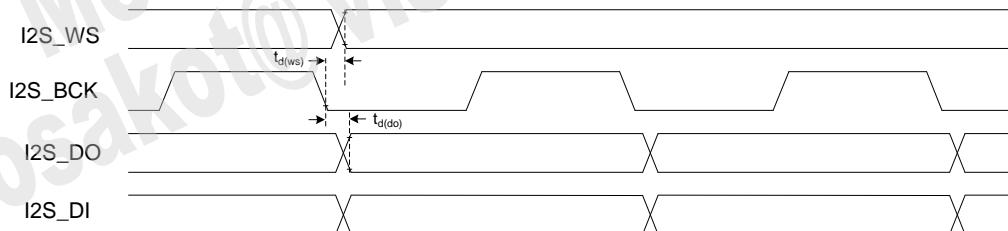


Figure 6-25 I2S Interface Timing

Table 6-6 I2S Interface Timing Delay

Parameter	Description	Max IO Delay	Unit
$t_{d(ws)}$	WS signal delay	10	ns
$t_{d(do)}$	Do signal delay	10	ns

PCM interface timing

- The IO interface timing of PCM is shown as below:

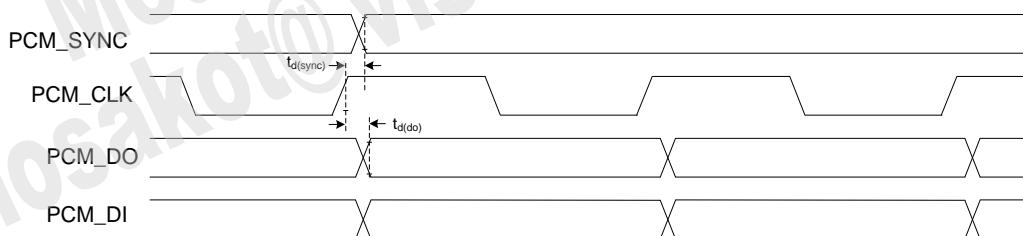


Figure 6-26 PCM Interface Timing

Table 6-7 PCM Interface Timing Delay

Parameter	Description	Max IO Delay	Unit
$t_{d(sync)}$	Sync signal delay	10	ns
$t_{d(do)}$	Do signal delay	10	ns

TDM interface timing

- The IO interface timing of TDM is shown as below:

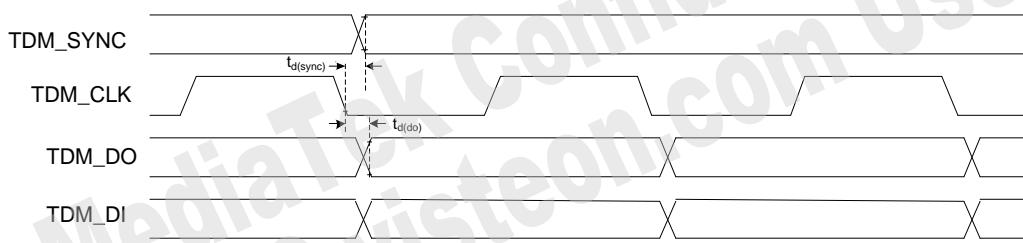


Figure 6-27 TDM Interface Timing

Table 6-8 TDM Interface Timing Delay

Parameter	Description	Max IO Delay	Unit
$t_{d(sync)}$	Sync signal delay	10	ns
$t_{d(do)}$	Do signal delay	10	ns

6.1.7 Register Definition

For register details, please refer to Chapter 4.1 of “MT2712 IVI Application Processor Registers”.

6.1.8 Programming Guide

6.1.8.1 I2S

6.1.8.1.1 Normal Use

- Enable clock
 - AUDIO_TOP_CON4
 - ASYS_TOP_CON
- Configure I2S related registers
 - ASYS_I2SI*_CON
 - ASYS_I2SO*_CON
- Enable I2S
 - ASYS_I2SI*_CON[0]
 - ASYS_I2SO*_CON[0]

6.1.8.1.2 One-to-one Internal Loopback (I2SOUT Loopback to I2SIN)

- Enable clock
 - AUDIO_TOP_CON4
 - ASYS_TOP_CON

- Use couple mode(I2SIN)
 - ASYS_I2SI*_CON[17]
- Configure I2S IN and I2S OUT related registers
 - ASYS_I2SI*_CON
 - ASYS_I2SO*_CON
- Enable I2S OUT
 - ASYS_I2SO*_CON[0]

6.1.8.1.3 Multi-channel Output Mode

- Enable clock
 - AUDIO_TOP_CON4
 - ASYS_TOP_CON
- Use one-heart mode
 - ASYS_I2SO*_CON[16]
- Configure I2S OUT related registers
 - ASYS_I2SO*_CON
- Software reset
 - ASYS_I2SO1_CON[30]
- Enable I2S OUT
 - ASYS_I2SO1_CON[0]

6.1.8.2 TDM

6.1.8.2.1 Normal Use

- Enable clock
 - AUDIO_TOP_CON1
 - AUDIO_TOP_CON2
 - AUDIO_TOP_CON4
 - ASYS_TOP_CON
- Configure TDM related registers
 - AFE_TDM_G*_CON*
 - AFE_TDM_G*_CONN_CON*
 - AFE_TDM_IN_CON*
- Enable TDM
 - AFE_TDM_G1_CON1[0]
 - AFE_TDM_G2_CON1[0]
 - AFE_TDM_IN_CON1[0]

6.1.8.2.2 One-to-one Internal Loopback

- Enable clock
 - AUDIO_TOP_CON1
 - AUDIO_TOP_CON2
 - AUDIO_TOP_CON4
 - ASYS_TOP_CON
- Use TDM in out sync mode
 - AFE_TDM_IN_CON1[18]
- Configure TDM IN and TDM OUT related registers
 - AFE_TDM_G*_CON*
 - AFE_TDM_G*_CONN_CONN*
 - AFE_TDM_IN_CON*
- Enable TDM IN
 - AFE_TDM_IN_CON1[0]
- Enable TDM OUT
 - AFE_TDM_G1_CON1[0]
 - AFE_TDM_G2_CON1[0]

6.1.8.3 PCM

PCM interface only has two control registers: PCM_INTF_CON1 and PCM_INTF_CON2. The suggested sequence for PCM data path register setting is as follows:

- Enable clock
 - AUDIO_TOP_CON4
 - ASYS_TOP_CON
- Configure PIN MUX
- Configure afe_interconn and afe_memif
 - AFE_CONN*
 - AFE_DAC_CON*
 - AFE_MEMIF_MINLEN*
 - AFE_MEMIF_HD_CON*
 - AFE_PCM*_BASE
 - AFE_PCM*_END
- Configure PCM interface and ASRC if needed.
 - AFE_PCM_INTF_CON*
 - ASRC (Refer to ASRC part)
- Enable PCM interface
 - AFE_PCM_INTF_CON1[0]

6.1.8.4 MRGIF

MRGIF interface only has two control registers: AFE_MRGIF_CON and AFE_DAIBT_CON0. The suggested sequence for MRGIF data path register setting is as follows:

- Enable clock
 - AUDIO_TOP_CON4
 - ASYS_TOP_CON
- Configure PIN MUX
- Configure afe_interconn and afe_memif
 - AFE_CONN*
 - AFE_DAC_CON*
 - AFE_MEMIF_MINLEN*
 - AFE_MEMIF_HD_CON*
 - AFE_*_BASE
 - AFE_*_END
- Configure mrgif and daibt.
 - AFE_DAIBT_CON0
 - AFE_MRGIF_CON
- Enable mrgif interface.
 - AFE_MRGIF_CON[0]

6.1.8.5 MEMIF

- Configure agent's sample rate
 - AFE_DAC_CON1
 - AFE_DAC_CON2
 - AFE_DAC_CON3
- Set start/end address
 - AFE_*_BASE
 - AFE_*_END
- Set other options
 - AFE_MEMIF_HD_CON*
 - AFE_MEMIF_PBUF_SIZE
 - AFE_MEMIF_MINLEN*
- Enable agents
 - AFE_DAC_CON0

6.1.8.6 ASRC

For the ASRC programming guide, please refer to the following steps:

Table 6-9 ASRC Programming Guide

1. Determine the operating frequency (49.152M/45.1584M) by the internal sample rate.		
- 49.152M : a1sys, 8k/12k/16k/24k/32k/48k/96k/192k/384k		
- 45.1584M : a2sys, 11.025k/22.05k/44.1k/88.2k/176.4k/352.8k		
2. Determine TX/RX mode based on whether the internal data are TX/RX.		
- RX mode : frequency mode, converting sample rate = IFS : OFS = input frequency : output frequency (IFS = input frequency, OFS = output frequency)		
- TX mode : period mode, converting sample rate = IFS : OFS = 1/OFS : 1/IFS = output period : input period (1/OFS = output period, 1/IFS = input period)		
SRC ratio uses IFS: OFS (frequency mode). For HW implementation, we need to keep OFS fix.		
In the TX mode, OFS is an external source and not fixed. As a result, the period mode is used, and period is the reciprocal of the frequency.		
So TX mode IFS: OFS = 1/OFS: 1/IFS = output period (TX IFS, period calibration): input period (TX OFS, fix period).		
3. Configure control registers.		
- If it is down-sample case, please set IIR coefficient first:		
AFE_ASRC_CON0	0x00000002	1. Set [1] = 1, meaning CPU controls IIR coefficient SRAM
AFE_ASRC_CON11	0x00000000	2. Set to 0. IIR coefficient SRAM address
AFE_ASRC_CON10	iir_coeff[index]	3. Write IIR coefficient to SRAM (max. 48 coefficients). Coefficient is generated by MATLAB. for(index=0;index<48;index++) AFE_ASRC_CON10 = iir_coeff[index];
Following is the common control flow:		
AFE_ASRC_CON1	1. TX OFS	TX OFS (period mode, fixed) = (49.152M/IFS)*64
AFE_ASRC_CON2	2. RX OFS	RX OFS (frequency mode, fixed) = Denominator/TX IFS
AFE_ASRC_CON3	3. RX IFS	RX IFS (frequency mode, updated by HW) , initialized to Denominator/TX OFS (or calibration result)
AFE_ASRC_CON4	4. TX IFS	TX IFS (period mode, updated by HW) , initialized to (49.152M/OFS)*64 (or calibration result)
AFE_ASRC_CON13	Auto reset threshold high	Set for calibration FS ((period mode)*0x11)>>4 If the calibration period result exceeds this threshold, ASRC HW will auto reset itself when calibration result is back to normal.
AFE_ASRC_CON14	Auto reset threshold low	Set for calibration FS ((period mode)*0xF)>>4 If the calibration period result is less than this threshold, ASRC HW will auto reset itself when the calibration result is back to normal.
- If the frequency calibrator is needed:		
AFE_ASRC_CON7	Denominator	For 49.152M: 0x3C00, for 45.1584M: 0x3720
AFE_ASRC_CON6	0x003F8886	[31:16] cali_cycle_minus_1, [15] auto_reset_en, [0] cali_en
AFE_ASRC_CON6	0x003F8887	[0] cali_en
AFE_ASRC_CON6	0x003F988F	[12] auto_tune_freq3, [3] auto_tune_freq2
4. User can read the following registers to check frequency/period calibration result.		
AFE_ASRC_CON8		Period calibration result = ((49.152M or 45.1584M)*cali_cycle/cali_FS)

AFE_ASRC_CON9		Frequency calibration result = Denominator/period_cali_result
5. After all settings are done, set AFE_ASRC_CON0 bit[0] to enable ASRC.		
AFE_ASRC_CON0		[31]one_heart, [15:14]ofs, [13:12]ifs, [4]clear If IIR is used, configure [11]iir_en, [10:8]iir_stage-1
AFE_ASRC_CON0		[31]one_heart, [15:14]ofs, [13:12]ifs, [4]clear, [0]on If IIR is used, configure [11]iir_en, [10:8]iir_stage-1 Set [4] to clear history every time ASRC is enabled. If ONE_HEART mode is used, only turn on the first UL/DL ASRC_CON0 bit[0] to enable all ASRC modules.
6. For Frequency/Period settings, please refer to the Register Map		
User can keep frequency settings the same by choosing a proper Denominator in 49.152M/45.1584M.		
However, Period settings are different in 49.152M/45.1584M. Please search the sample rate settings in the Register Map according to scenarios.		
7. User can use the calibration result as the initial value of AFE_ASRC_CON3/AFE_ASRC_CON4.		
1) Let calibration signal work.		
2) Enable CALI.		
3) Enable auto_tune_freq2 or 3; then AFE_ASRC_CON3/AFE_ASRC_CON4 will be updated by HW.		
4) The calibration result is also shown in AFE_ASRC_CON8/AFE_ASRC_CON9.		

6.1.8.7 HW Interrupt

The audio system has the following seven IRQs. The first two IRQs are for general usage and the last five are for HW ASRC.

- Configure IRQ's sample rate/irq_cnt
 - ASYS_IRQ*_CON
 - AFE_IRQ_MCU_CON
 - AFE_IRQ_MCU_CNT*
- Enable IRQ
 - ASYS_IRQ*_CON[31]

Table 6-10 IRQ Information

IRQ	Timer Unit		irq_on	irq_mode	irq_cnt	irq_status	Function
IRQ 1	Max fs=192 K	afe_irq_1	AFE_IRQ_MCU_CON[0]	AFE_IRQ_MCU_C ON[7:4]	AFE_IRQ_MCU_C NT1[17:0]	AFE_IRQ_MCU_S TATUS[0]	Common
		afe_irq_2	AFE_IRQ_MCU_CON[1]	AFE_IRQ_MCU_C ON[11:8]	AFE_IRQ_MCU_C NT2[17:0]	AFE_IRQ_MCU_S TATUS[1]	Common
IRQ 2	Max fs=384 K	asys_irq_1	ASYS_IRQ1_CO N[31]	ASYS_IRQ1_CON[28:24]	ASYS_IRQ1_CON[23:0]	ASYS_IRQ_STATU S[0]	Common
		asys_irq_2	ASYS_IRQ2_CO N[31]	ASYS_IRQ2_CON[28:24]	ASYS_IRQ2_CON[23:0]	ASYS_IRQ_STATU S[1]	Common

IRQ	Timer Unit		irq_on	irq_mode	irq_cnt	irq_status	Function
		asys_ir_q_3	ASYS IRQ3_CO_N[31]	ASYS IRQ3_CON[28:24]	ASYS IRQ3_CON[23:0]	ASYS IRQ_STATU S[2]	Common
		asys_ir_q_4	ASYS IRQ4_CO_N[31]	ASYS IRQ4_CON[28:24]	ASYS IRQ4_CON[23:0]	ASYS IRQ_STATU S[3]	Common
		asys_ir_q_5	ASYS IRQ5_CO_N[31]	ASYS IRQ5_CON[28:24]	ASYS IRQ5_CON[23:0]	ASYS IRQ_STATU S[4]	Common
		asys_ir_q_6	ASYS IRQ6_CO_N[31]	ASYS IRQ6_CON[28:24]	ASYS IRQ6_CON[23:0]	ASYS IRQ_STATU S[5]	Common
		asys_ir_q_7	ASYS IRQ7_CO_N[31]	ASYS IRQ7_CON[28:24]	ASYS IRQ7_CON[23:0]	ASYS IRQ_STATU S[6]	Common
		asys_ir_q_8	ASYS IRQ8_CO_N[31]	ASYS IRQ8_CON[28:24]	ASYS IRQ8_CON[23:0]	ASYS IRQ_STATU S[7]	Common
		asys_ir_q_9	ASYS IRQ9_CO_N[31]	ASYS IRQ9_CON[28:24]	ASYS IRQ9_CON[23:0]	ASYS IRQ_STATU S[8]	Common
		asys_ir_q_10	ASYS IRQ10_CO_N[31]	ASYS IRQ10_CON[28:24]	ASYS IRQ10_CON[23:0]	ASYS IRQ_STATU S[9]	Common
		asys_ir_q_11	ASYS IRQ11_CO_N[31]	ASYS IRQ11_CON[28:24]	ASYS IRQ11_CON[23:0]	ASYS IRQ_STATU S[10]	Common
		asys_ir_q_12	ASYS IRQ12_CO_N[31]	ASYS IRQ12_CON[28:24]	ASYS IRQ12_CON[23:0]	ASYS IRQ_STATU S[11]	Common
		asys_ir_q_13	ASYS IRQ13_CO_N[31]	ASYS IRQ13_CON[28:24]	ASYS IRQ13_CON[23:0]	ASYS IRQ_STATU S[12]	Common
		asys_ir_q_14	ASYS IRQ14_CO_N[31]	ASYS IRQ14_CON[28:24]	ASYS IRQ14_CON[23:0]	ASYS IRQ_STATU S[13]	Common
		asys_ir_q_15	ASYS IRQ15_CO_N[31]	ASYS IRQ15_CON[28:24]	ASYS IRQ15_CON[23:0]	ASYS IRQ_STATU S[14]	Common
		asys_ir_q_16	ASYS IRQ16_CO_N[31]	ASYS IRQ16_CON[28:24]	ASYS IRQ16_CON[23:0]	ASYS IRQ_STATU S[15]	Common
IRQ 3							Memory ASRC1
IRQ 4							Memory ASRC2
IRQ 5							Memory ASRC3
IRQ 6							Memory ASRC4
IRQ 7							Memory ASRC5

6.2 MFlexGraphics

6.2.1 Introduction

MFlexGraphics (MFG) is a highly integrated high-performance and low-power Graphic Processing Unit (GPU), and supports full capability of industry graphics standard Application Programming Interfaces (APIs) – OpenGL ES 1.1, OpenGL ES 2.0, and OpenGL ES 3.0.

6.2.2 Features

MFG is powered by ARM Mali-T880 series. The Mali-T880 GPU is a hardware accelerator for 2D and 3D graphics systems.

Following API graphics industry standards are supported:

- OpenGL ES 1.1 Specification
- OpenGL ES 2.0 Specification
- OpenGL ES 3.0 Specification
- OpenCL 1.0 Specification
- OpenCL 1.1 Specification
- OpenCL 1.2 Specification

6.2.3 3D Features

- A rich API feature set with high-performance support for both shader-based and fixed-function graphics APIs
- Anti-aliasing capabilities
- An effective core for General Purpose computing on GPU (GPGPU) applications
- High memory bandwidth and low power consumption for 3D graphics content
- Scalability for products from smart phones to high-end mobile computing
- Image quality using double-precision FP64, and anti-aliasing
- Standard bus interfaces
- An ACE-Lite interface to access external memory
- Latency tolerance
- Compressed texture formats
- Configurable per-core power management for enabling the optimal power and performance combination for each application
- Coherency aware interconnects for system memory and resource sharing
- Frame buffer compression

6.2.4 Block Diagram

MFG contains Mali-T880 mp4, mfg_top_config, mfa_apb_async_rx and mfg_async_vcore. mfg_top_config provides a control for clock and reset. mfg_apb_async_rx receives configuration setting from CPU.

mfg_async_vcore provides a clock domain conversion from mfg_core_ck domain to hf_mem_ck, which is DRAM clock domain.

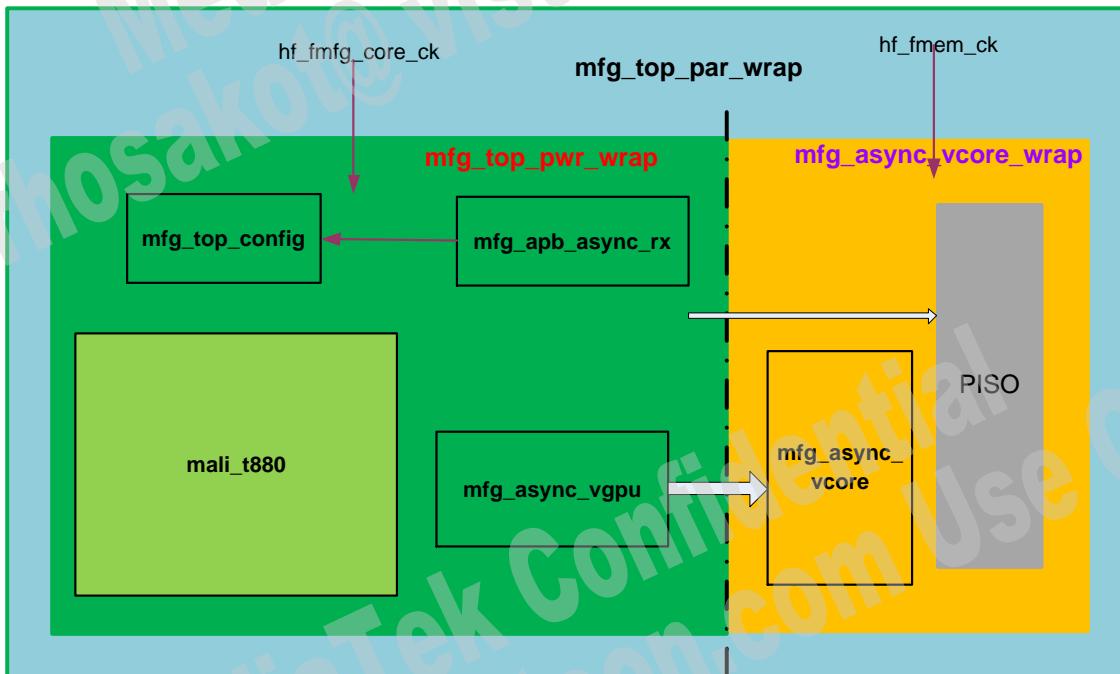


Figure 6-28 MFlexGraphics Block Diagram

6.2.5 Register Definition

For register details, please refer to Chapter 4.2 of “MT2712 IVI Application Processor Registers”.

6.2.6 Programming Guide

MFlexGraphics is powered by the Mali-T880 series core from ARM. For more information on register descriptions and programming guide, refer to the ARM TRM (Technical Reference Manual).

6.3 Image Subsystem

6.3.1 Introduction

MT2712's IMGSYS_TOP subsystem contains MIPI Camera Serial Interface (CSI), Composite Video Baseband Signal (CVBS) decoder and Picture Quality (PQ) handle. The CSI receives data from external camera and pack data. All the data are sent to DRAM buffer then. For CVBS input source, the TV Decoder (TVD) module decodes the source and saves the data to DRAM buffer. The PQ mainly includes De-interlace (DI) and Noise Reduction (NR). After PQ, all the data are saved to DRAM buffer too.

6.3.2 Features

The IMGSYS_TOP subsystem has the following features:

- Support 2 CSI (4×1280×1024@30 Hz or 1920×1080@60 Hz)
- Support CVBS input with 3D comb filter
- 3D Noise Reduction
- De-interlace (support up to 1080i DI)

6.3.3 Block Diagram

Figure 6-29 depicts the IMGSYS_TOP block diagram.

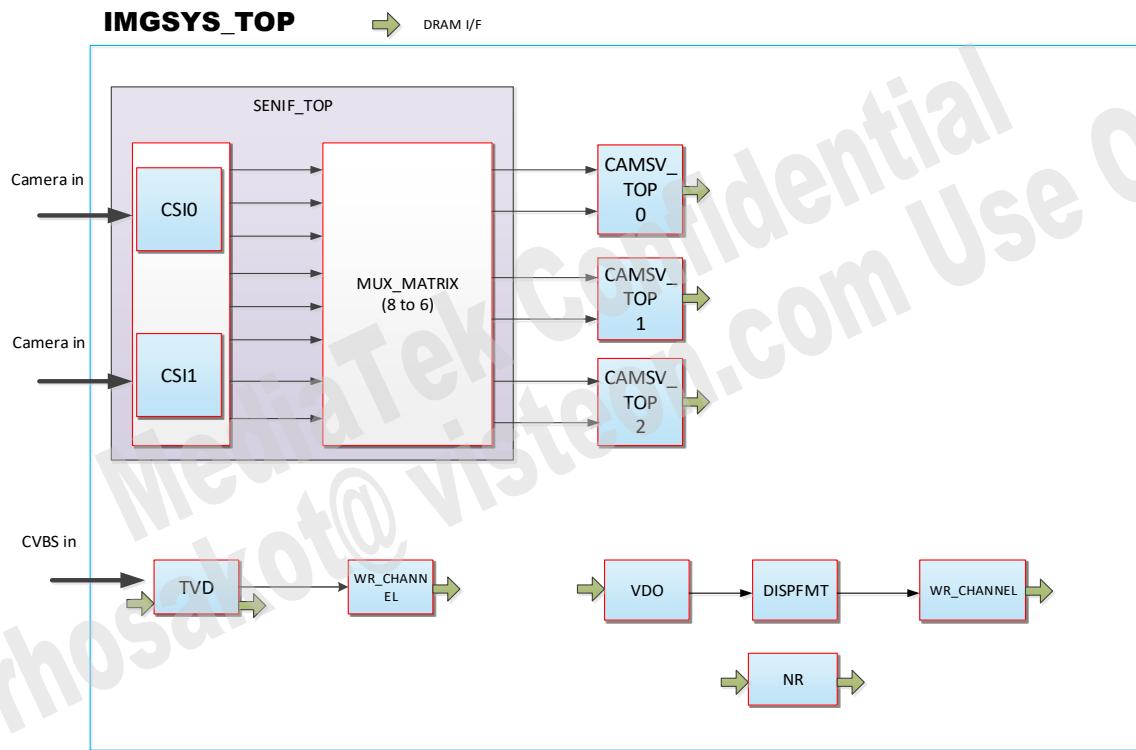


Figure 6-29 IMGSYS_TOP Function Blocks

6.3.4 Register Definition

For register details, please refer to Chapter 4.3 of “MT2712 IVI Application Processor Registers”.

6.4 Camera Serial Interface

6.4.1 Introduction

Camera Serial Interface (CSI) is based on MIPI Alliance Specification, supporting high-speed serial data transfer between the host processor and the peripheral devices such as camera modules.

6.4.2 Features

The CSI engine has the following features for Camera Serial Interface:

- 1 clock lane and up to 4 data lanes
- Throughput up to 1.5 G bps for 1 data lane
- Main cam 1.5 G x 4-lane
- Sub cam 1.5 G x 4-lane
- Compliant with electrical characteristics of D-PHY v1.1
- Compliant with MIPI CSI2 v1.3

6.4.3 Block Diagram

The CSI block diagram is shown as Figure 6-30. The clock domains are shown in Figure 6-31. The Analog receives the clock sent from sensor and divides it to BYTE clock. The number of cam interface at seninf_top is eight, but now only six camsvs are connected in MT2712.

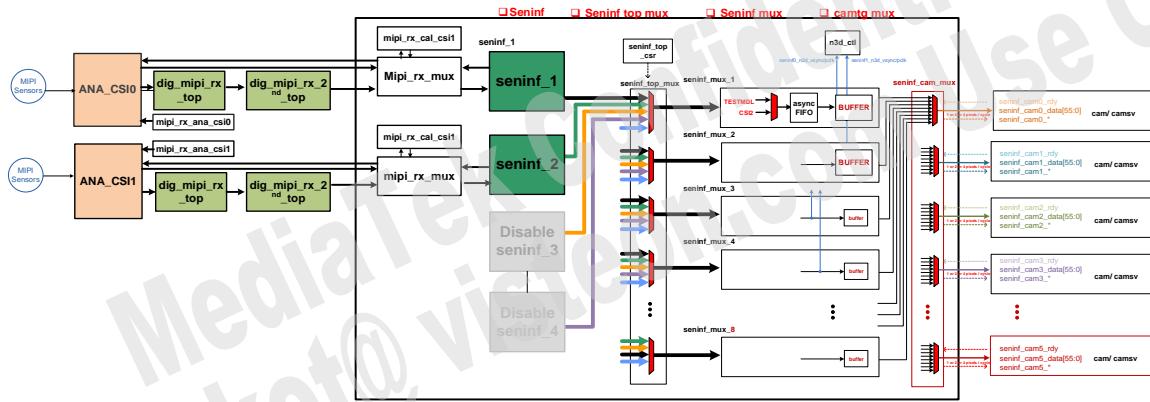


Figure 6-30 CSI Block Diagram

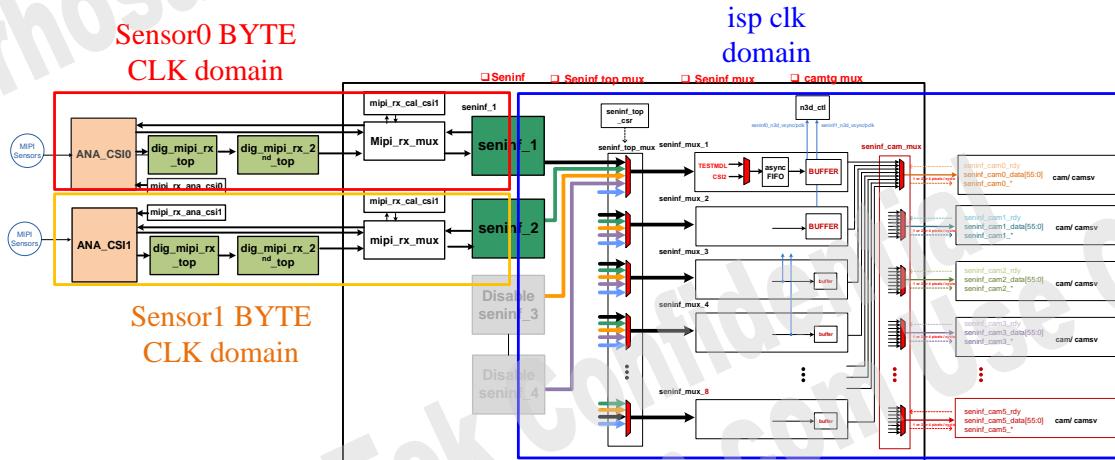


Figure 6-31 CSI CLK Domain

6.4.4 Register Definition

For register details, please refer to Chapter 4.3 of “MT2712 IVI Application Processor Registers”.

6.4.5 Programming Guide

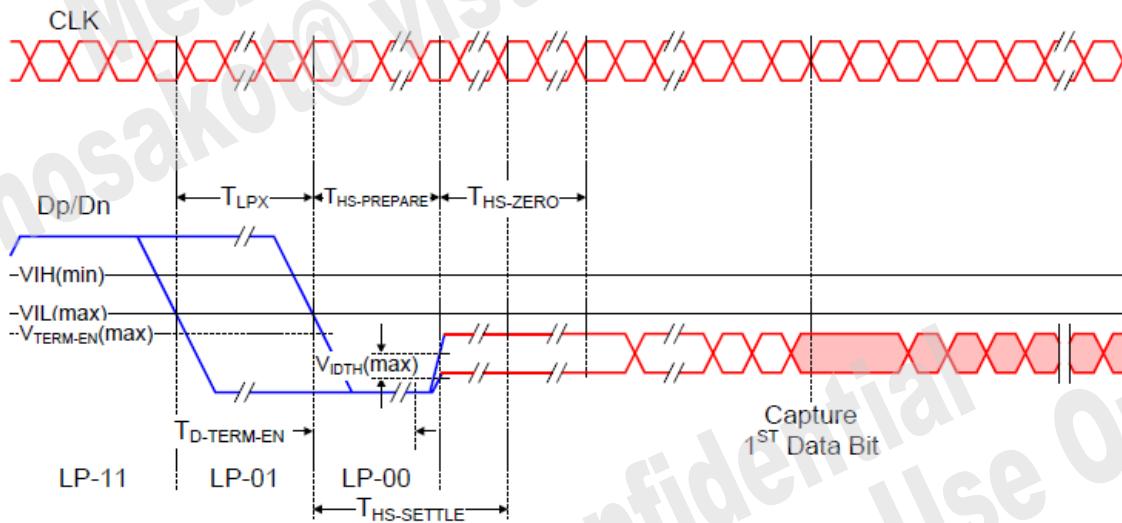


Figure 6-32 Settle Delay Calculations

$T_{hs-settle}$ is shown in Figure 6-32.

The Settle Parameters (SP) = $(T_{hs-settle})/\text{csi_ck period} - 3$, where 3 is the digital pipe delay.

6.5 CAM Write Channel

6.5.1 Introduction

CAM Write Channel (WR_CHANNEL) can receive video source of multiple formats from master module and then write video data into DRAM. The color space of input video is either YCbCr 444 or YCbCr 422. If it is YCbCr 444, WR_CHANNEL will perform a 444 to 422 conversion. The data written to DRAM can be YCbCr 422 8-bit or YCbCr 420 8-bit. Mostly, YCbCr 420 8-bit is used in order to save the DRAM bandwidth.

Data will be written in two planes. Both block and scan-line modes are supported.

6.5.2 Features

- Supports YCbCr 444/422 12/10/8-bit input
- Supports YCbCr 420 8-bit 2 planes block/scan-line output
- Supports YCbCr 422 8-bit 2 planes scan-line output
- Support for 1080p@60 Hz input
- Supports YCbCr data channel swap
- Supports data bit swap

6.5.3 Block Diagram

Figure 6-33 shows the internal blocks of WR_CHANNEL. The input is from master module, and the output is directly sent to DRAM. There is an APB interface for SW control.

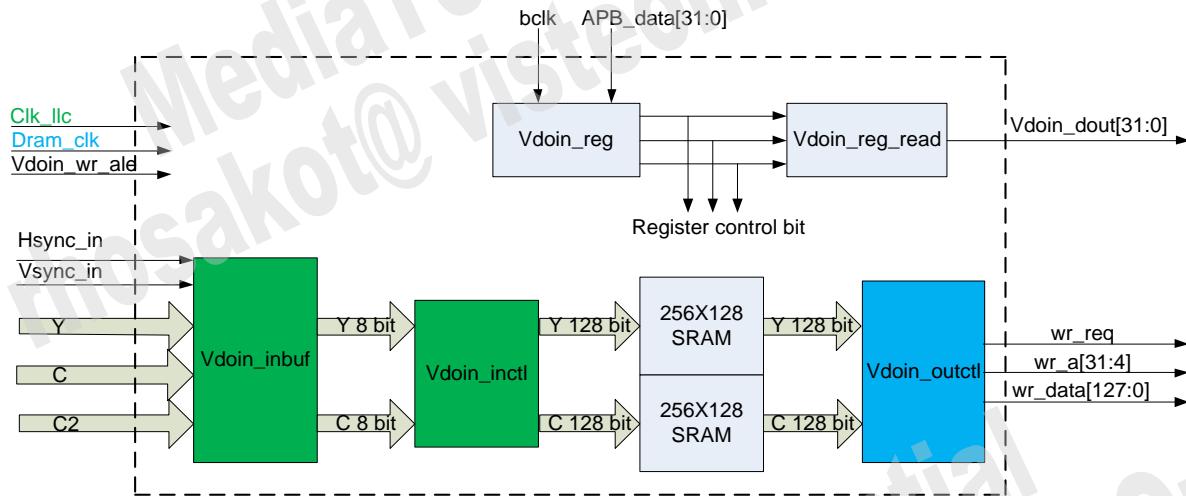


Figure 6-33 WR_CHANNEL Block Diagram

6.5.4 Register Definition

For register details, please refer to Chapter 4.4 of “MT2712 IVI Application Processor Registers”.

6.5.5 Programming Guide

- Enable WR_CHANNEL: VDOIN_EN[0]
- VDOIN_EN[25] selects whether to write 420 or 422 mode to DRAM.
- VDOIN_EN[14] selects whether to write block or raster mode to DRAM.
- WR_CHANNEL will generate interrupt every field or frame. SW can use the interrupt to switch DRAM address.
- VSCALE[13] can be used for debugging purpose which will write grayscale pictures to DRAM.

6.6 H.264 Video Encoder

6.6.1 Introduction

This design is with the main stream H.264 video encoder. It is capable of encoding 1080p video at 30 frames per second (FPS) with promising superior video quality. This IP supports various encoding methods that satisfy the basic requirements of higher software controllability. Furthermore, with advanced encoding technology, it

brings astonishingly high quality and low memory bandwidth requirements. It also considers the usage of portable devices so it provides several power saving capabilities.

6.6.2 Features

The supported video codec and the capabilities are listed in Table 6-11.

Table 6-11 Main Features

H.264 Encoder	Profile	High
	Level	4.1
	Speed	1080p@30fps

The video encoder takes DRAM as input, output and working buffer. It reads input frame buffers, executes video encoding and writes encoded bitstream to the output buffer. The driver software maintains all buffers and assigns proper values to the video encoder to allow the hardware to work properly. Figure 6-34 shows the procedure of the video encoder. YUV420 two plane scan-line (NV12/NV21) and YUV420 three plane scan-line (YV12/I420) are supported.



Figure 6-34 Video Encoder Procedure

6.6.3 Block Diagram

Figure 6-35 shows the brief IP architecture and local on-chip-bus architecture. The control interface consists of ARM APB bus and MediaTek's proprietary Smart Multimedia Interface (SMI) bus. It reports hardware events through interrupts or software polling. In addition, it adopts several SMI ports and one APB port. The video encoder is configured by software through the APB interface. As the register is configured, the sequencer will send the corresponding control signals to trigger sub-modules. DMA will acquire and store back the image data and bitstream from and to memory according to the configured address. Motion Estimation (ME) will conduct motion estimation to decide the motion vector for later encoding. Motion Compensation (MC) will conduct motion compensation to give predicted pixel values. Transform and Quantization (TQ) will conduct transformation and quantization operation and write reconstructed pixels to De-blocking filter DB and quantized transformed coefficient to Entropy encoder (EC). DB will conduct de-blocking operation and allow the DMA to store back the processed frame as the next frame's reference frame. EC will conduct entropy encoding, and the coding can be variable length code, context-based arithmetic code or context-based variable length code. The encoded bitstream will be written to memory by DMA.

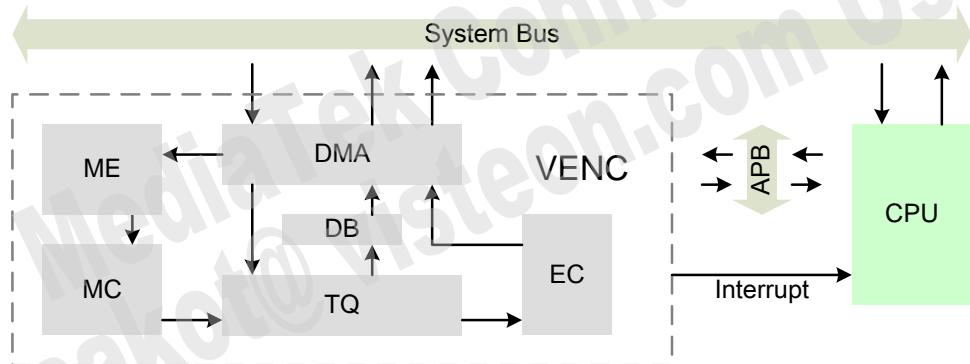


Figure 6-35 Video Encoder Block Diagram

6.6.4 Register Definition

For register details, please refer to Chapter 4.6 of “MT2712 IVI Application Processor Registers”.

6.7 Video Decoder

6.7.1 Introduction

MediaTek’s Video Decoder (VDEC) in MT2712 supports multi-standard video compression format. It can greatly reduce CPU loading and achieve high performance video decompression. VDEC considers the usage of portable devices and provides considerable power saving capabilities.

6.7.2 Features

VDEC supports the following multimedia video formats:

- VP9 decoder:
 - Profile 0, 2
- HEVC decoder:
 - Main, main_10 Profile@L4.1
- H.264 decoder:
 - Constrained Baseline Profile (CBP)
 - Main Profile (MP)
 - Progressive High Profile (ProHiP)
 - Constrained High Profile(CHiP)
 - High Profile (HiP)@4.2
- MPEG-4 decoder:
 - Sorenson H.263/H.263
 - SP@L6/ASP@L5
- MPEG-1 video

- MPEG-2 video
- MP@ML and MP@HL
- VP8 decoder
- WMV decoder
- WMV 7/8/9, VC-1 main profile, advanced profile@L4
- Adaptive MPEG de-blocking filter
- Error handling

VDEC supports full-HD 60fps streams (Does not support picture width > 2048 or picture height > 1088).

6.7.3 Adaptive MPEG De-blocking Filter

Since the MPEG video is compressed in a block-by-block manner, inter-block discontinuity may occur in low bit rate cases. MediaTek's proprietary adaptive MPEG de-blocking filter is provided to improve the video quality, delivering a more pleasant viewing experience, as shown in Figure 6-36, where the de-blocked result is shown on the right side.



Figure 6-36 De-blocking Image

6.7.4 Error Handling

The video package may be lost or corrupted during broadcast transmission. Elementary streams may also be damaged due to the attrition of BD/DVD disk. These corruption will cause error bitstream. In many situations, error bitstream cannot be recovered by channel coding or re-sent in broadcasting or internet. Our hardware provides some error handling tools to display better video quality for error bitstream. Two examples are shown in Figure 6-37 and Figure 6-38.

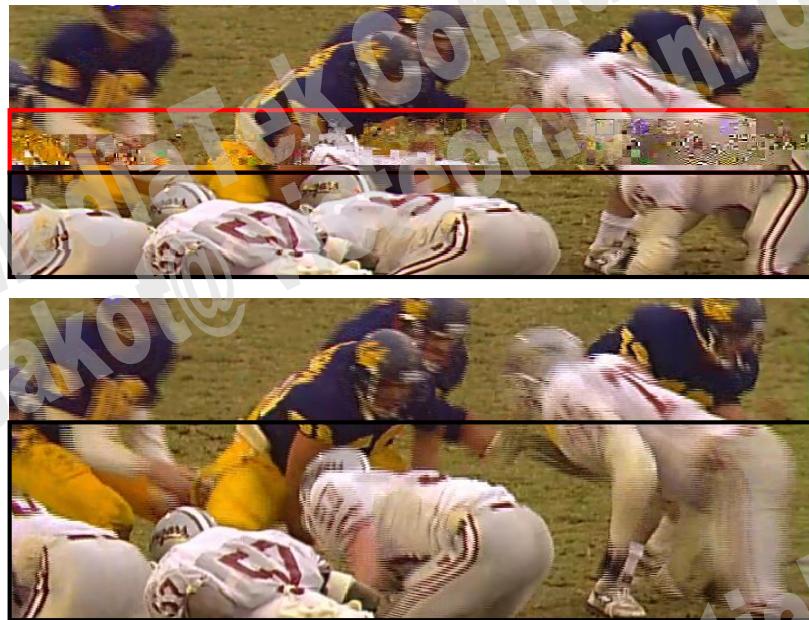


Figure 6-37 Error Handling Example 1

In Figure 6-37, the upper part shows the result of using free software decoder; the lower part shows the result of using our hardware.

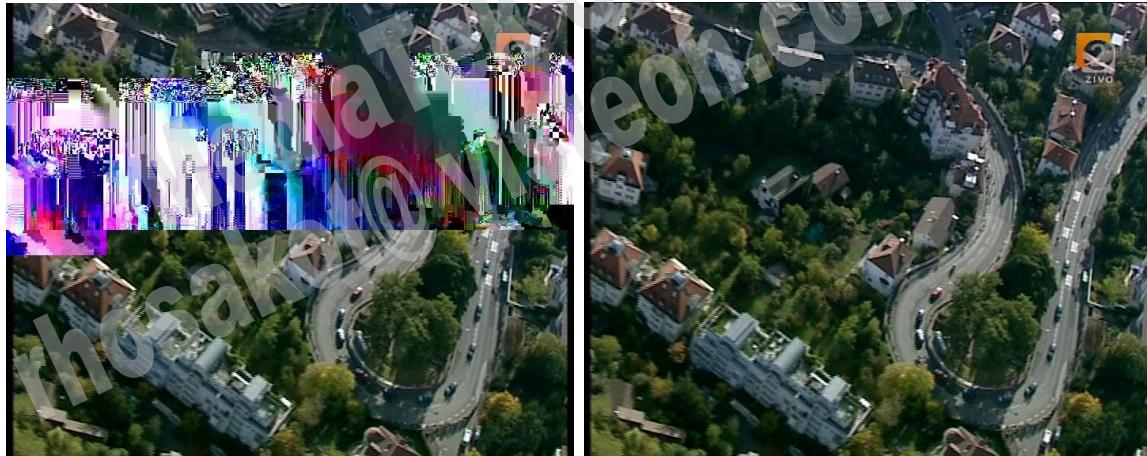


Figure 6-38 Error Handling Example 2

In Figure 6-38, the left part shows the result of using free software decoder; the right part shows the result of using our hardware.

6.7.5 Block Diagram

The architecture and core blocks of VDEC are shown in Figure 6-39, which include the following parts: VDEC 442 MHz clock, Entropy Decoder, IS/IQ/IT, MV Calculation, Intra Prediction, Motion Compensation and De-

blocking Filter. The input to VDEC is a compressed video bitstream. After the decoding process, the reconstructed video will be sent to the display stage.

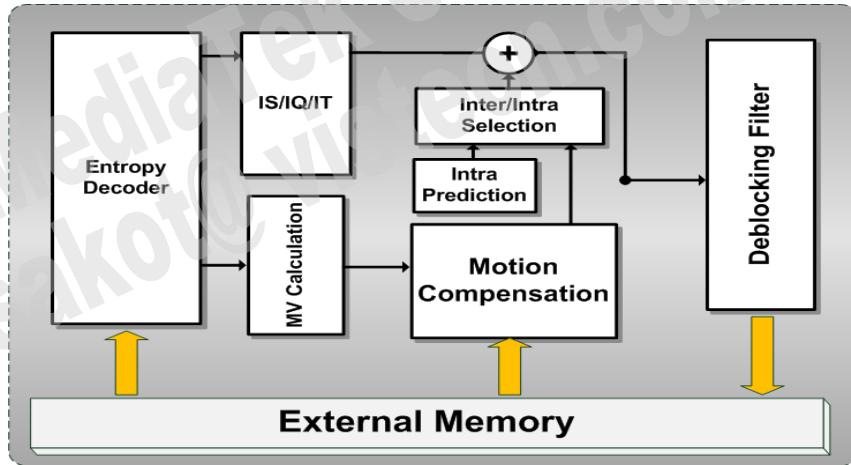


Figure 6-39 VDEC Block Diagram

6.7.6 Register Definition

For register details, please refer to Chapter 4.8 of "MT2712 IVI Application Processor Registers".

6.8 JPEG Decoder

6.8.1 Introduction

At present, most images must be stored as JPEG format compressed files. The hardware JPEG decoder is developed to display this kind of file and boost image processing performance. The JPEG decoder is designed to decode baseline JPEG file with general YUV sampling combinations. (Note that the progressive decoding is NOT supported in the hardware decoder).

To obtain the best speed performance, the JPEG decoder handles all portions of JPEG files except for the 17-byte SOF marker. The software only needs to program the related control registers based on the SOF marker, and waits for an incoming interrupt from the hardware. Figure 6-40 shows the basic JPEG file structure, and starting address that JPEG decoder needs. The information of DQT and DHT table is included in the JPEG file, which needs to be parsed by the JPEG decoder, and then stored in the memory.

Taking the limited size of memories into consideration, the hardware decoder also supports breakpoints insertion in huge JPEG files. Breakpoints insertion allows software to load partial JPEG file from the external flash into the internal memory, if the JPEG file is too large to be stored internally at a time.

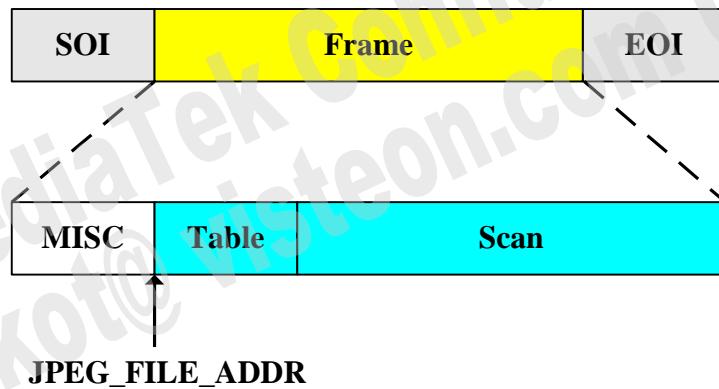


Figure 6-40 Basic Structure of JPEG Files

6.8.2 Features

The sleep controller receives the command from system software and controls the built-in power management unit to cut off the power supply of an external clock source. The power of an external clock source can be resumed by several events predefined by users. These events are issued by:

- Baseline JPEG decoding (No progressive decoding, no restart marker decoding, bypass to software solution)
- Sampling format limitation: See Figure 6-41
- Hardware table parsing (SW handles SOF parsing)
- Support file breakpoint
- 1/2/4/8 block resize (some limitation due to format conversion, see Section 6.8.3)
- Support MCU rows pause/resume
- Error handling by bit stream overflow detection
- Input sampling format of JPEG decoder: See Figure 6-41
- Software solutions are adopted for input sampling format that is not supported by the hardware

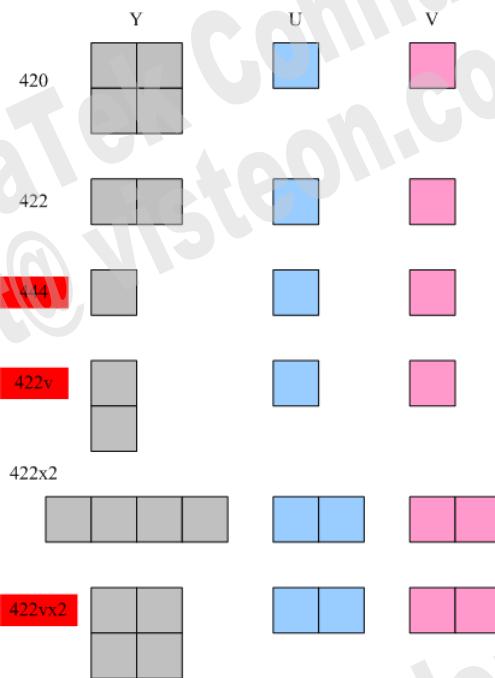


Figure 6-41 Supported Sampling Format of Input JPEG File

6.8.3 Block Diagram

As shown in Figure 6-42, the whole module is under the same clock domain. The JPEG bit stream flows in from the SMI bus controlled by BSDMA block. The Huff table block is used for saving Huffman value which is decoded from bit stream header. Similarly, the Q table block is used for storing the quantification value from header. After processing the bit stream by VLD and IDCT, a whole frame flowing out from SMI bus is controlled by the WDMA block.

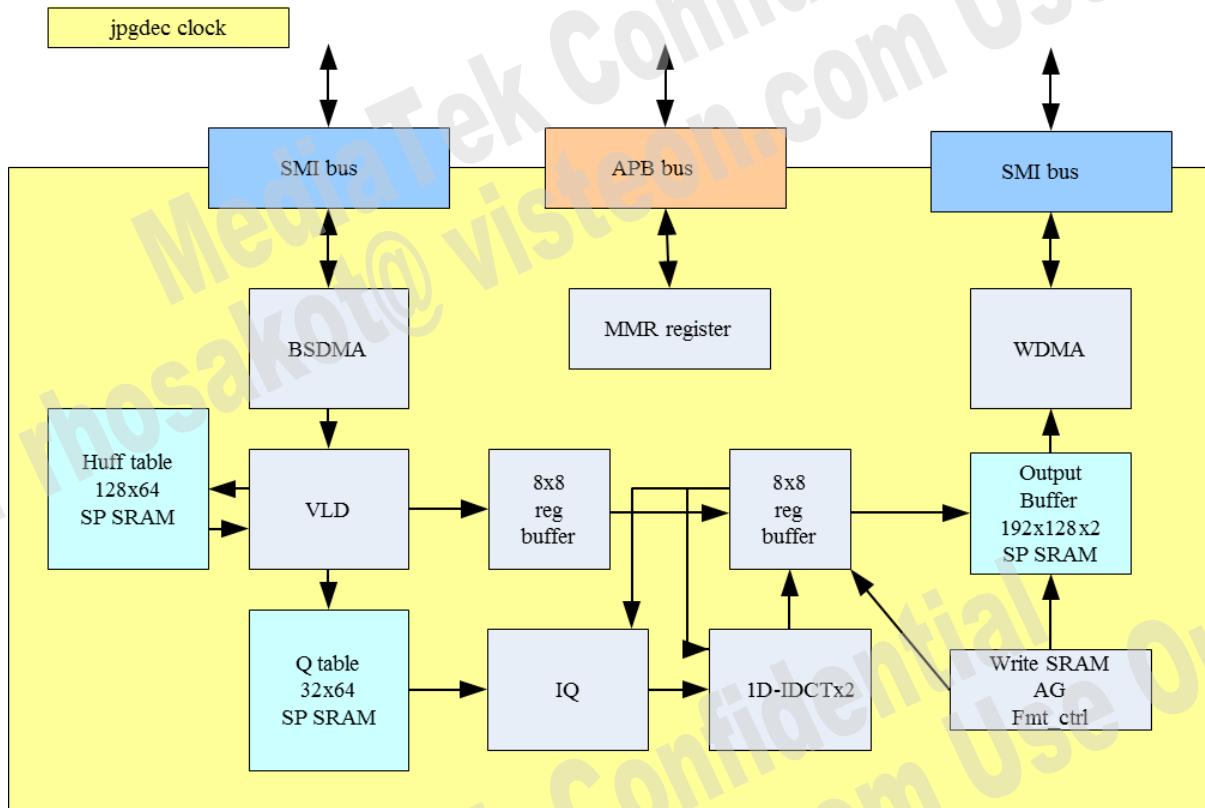


Figure 6-42 jpgdec Block Diagram

6.8.4 Register Definition

For register details, please refer to Chapter 4.7 of “MT2712 IVI Application Processor Registers”.

6.8.5 Programming Guide

6.8.5.1 Normal Mode (Full-frame Mode)

1. Set the memory related information
2. Set the address of output data
3. Three buffer addresses: Y, U, V
4. Y stride, UV stride
5. Set the address of bit stream
6. Set the image information
7. Sampling format
8. SOF pre-parsing information (COMP_ID, QT_ID)
9. Total MCU number
10. Set color format conversion related register
11. Set control options
12. Block resize factor for resize or format conversion if needed

13. Trigger module
14. Wait for IRQ

6.8.5.2 Breakpoint Mode

1. Set the memory related information
2. Set the address of output data
3. Three buffer addresses: Y, U, V
4. Y stride, UV stride
5. Set the address of bit stream
6. Set the breakpoint of bit stream
7. Set the image information
8. Sampling format
9. SOF pre-parsing information (COMP_ID, QT_ID)
10. Total MCU number
11. Set color format conversion related register
12. Set control options
13. Block resize factor for resize or format conversion if needed
14. Trigger module
15. Wait for IRQ
16. If it is not the end of the bit stream, go to step 1 after the new bit stream segment has been prepared.

6.8.5.3 Pause/Resume Mode

1. Set the memory related information
2. Set the address of output data
3. Three buffer addresses: Y, U, V
4. Y stride, UV stride
5. Set the address of bit stream
6. Set the image information
7. Sampling format
8. SOF pre-parsing information (COMP_ID, QT_ID)
9. Total MCU number
10. Set color format conversion related register
11. Set pause MCU index
12. Set control options
13. Block resize factor for resize or format conversion if needed
14. Trigger module
15. Wait for IRQ
16. Go to step 1 if the last MCU is not processed, or else complete the decoding process

6.9 Image Resizer

6.9.1 Introduction

Image Resizer (IMG_RSZ) in MT2712 can scale images for Video Decoder (VDEC). In detail, the decode format “RMVB” in video decoder will use the image resizer to perform scaling in mode “RPR”. The image resizer engine can issue commands via CPU/BIM or command queue. Based on the setting of control registers, the engine reads source image via read buffer, and then the source image is scaled in two directions. Next, the engine writes destination image to DRAM via write buffer. The engine involves three types of DRAM interfaces. One is command queue read request, and the other two are read/write image requests.

6.9.2 Features

The image resizer engine in MT2712 supports Video format.

- Video source format:
Raster scan or block based YCbCr 422/420 formats.
- Video source format output:
Raster scan or block based YCbCr 422/420 formats.

The image resizer contains the following filters for scaling:

- Horizontal 8-tap filter.
- Vertical 4-tap filter.

The range of source image width/height is [1: 65535] pixels.

The range of destination image width/height is [1: 4095] pixels.

The scale ratio is 1/128 ~ 128.

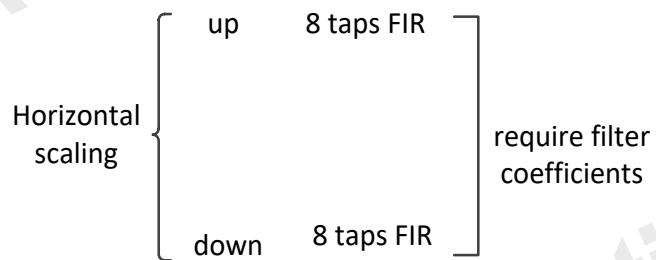


Figure 6-43 Filter Types of Image Resizer in Horizontal Scaling

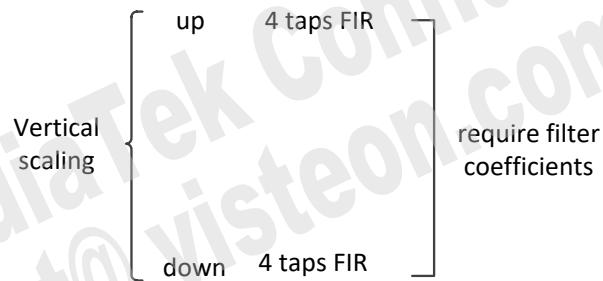


Figure 6-44 Filter Types of Image Resizer in Vertical Scaling

6.9.3 Block Diagram

Figure 6-45 illustrates the architecture and core blocks of IMG_RSZ, including control registers, horizontal scaling, line buffer, vertical scaling, read buffer and write buffer. The input data comes from Dynamic Random-Access Memory Controller (DRAMC), and after scaling, output data is sent to DRAMC.

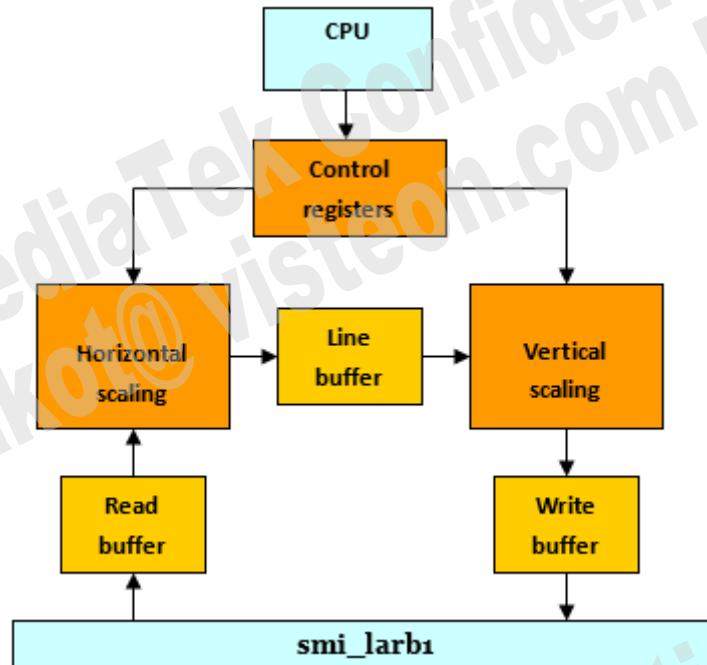


Figure 6-45 IMG_RSZ Block Diagram

6.9.4 Register Definition

For register details, please refer to Chapter 4.9 of “MT2712 IVI Application Processor Registers”.

6.9.5 Programming Guide

6.9.5.1 Power-On Sequence

In order to avoid asynchronous interface problem, image resizer should follow a specific sequence as follows:

1. Base address + 0x000: Write 0x0000_008C (Pull down soft reset before any register setting)
2. The other functional settings, ex. width/height.
3. Base address + 0x000: Write 0x0000_0080 (Pull up)
4. Base address + 0x000: Write 0x0000_0083 (Enable clock)
5. Base address + 0x000: Write 0x0000_0082 (Enable clock, start image resizer)

6.10 Display Subsystem

6.10.1 Introduction

The Display Subsystem includes Multimedia Subsystem (MMSYS_TOP) and Image Subsystem (IMGSYS_TOP).

The MMSYS_TOP contains multimedia controller, Multimedia Data Path v2.0 (MDP 2.0) and Display (DISP). The multimedia controller includes bus fabric control, Smart Memory Interface (SMI) control, memory access second-level arbiter, and multimedia configuration. MMSYS_TOP plays the key role to handle different handshakings between infra subsystem, video subsystem and image subsystem. MDP 2.0 is the time-sharing pipeline data flow controller to process resizing and rotation by memory access. The display pipeline outputs pixels to display interface with overlay, color enhancement, adaptive ambient light processing, color correction and gamma correction.

The IMGSYS_TOP contains MIPI Camera Serial Interface (CSI), Composite Video Broadcast Signal (CVBS) decoder and Picture Quality (PQ) handle. The CSI receives data from external camera and packs the received data, and then all the data are sent to the DRAM buffer. For CVBS input source, the TVD module decodes the source and saves the data to DRAM buffer. The PQ mainly includes De-interlace (DI) and Noise Reduction (NR), and after PQ handle, all the data are saved to DRAM buffer too.

6.10.2 Features

MT2712's MMSYS_TOP supports the following features:

- APB bus fabric control center
- SMI controller
- Second-level arbiter for memory access request
- MDP 2.0 has four read DMAs, three resizers, three 2D-sharpness enhancements, one write DMA and three write rotators

- Two display pipelines. The display pipe has its own read DMA, overlay, color engine, adaptive ambient light processing, color correction, gamma correction, over drive and display interface controller as basic components
- One video recorder path. The path includes read DMA, overlay, color engine, gamma correction, and one write DMA
- Color enhancement engine
- Adaptive ambient light processing for backlight power saving and sunlight visibility improvement
- Color correction and gamma correction for accurate image reproduction
- Concurrent dual display output
- Display output interface: 2 × 4 lane DSI, 2 dual-link Low-Voltage Differential Signaling (LVDS)
- Video safety detection: four display monitors. The IP monitor DSI/LVDS input data inside the Region of Interest (ROI) and compute the Cyclic Redundancy Check (CRC), and then compare the data with the golden value. The comparison result indicates whether the display content is as expected

MT2712's IMGSYS_TOP supports the following features:

- Two CSIs (4 × 1280×1024@30 Hz or 1920×1080@60 Hz)
- CVBS input, with 3D comb filter
- 3D Noise Reduction
- De-interlace (supports up to 1080i DI)

6.10.3 Block Diagram

There are two main blocks in MMSYS_TOP: one is MDP v2.0, and the other is DISP pipeline with picture quality enhancement and high-resolution.

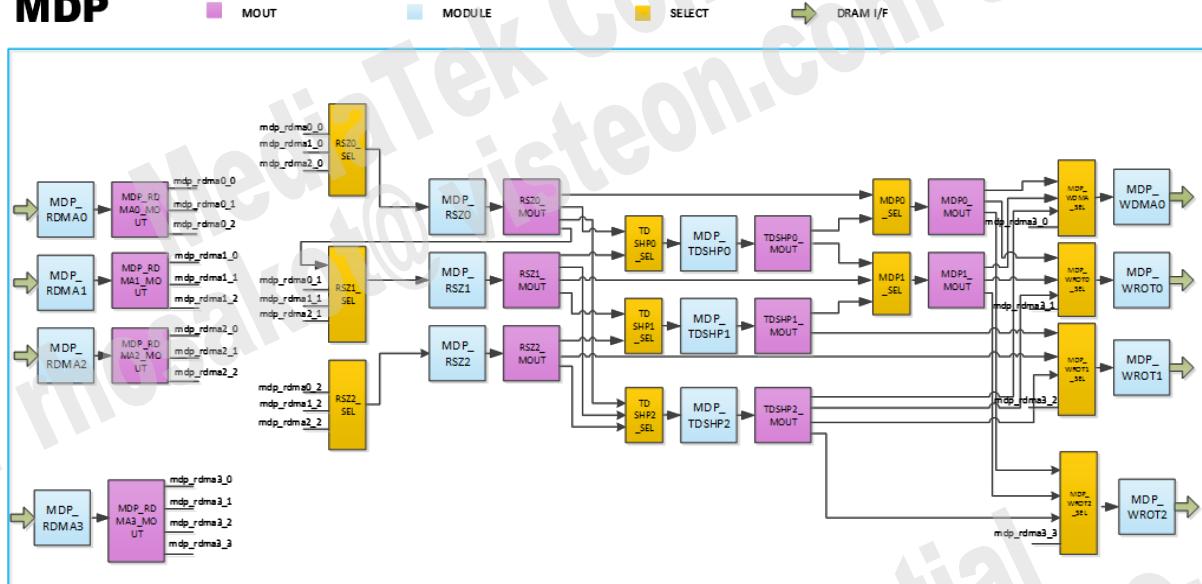
MDP

Figure 6-46 MDP Functional Blocks of Multimedia Partition

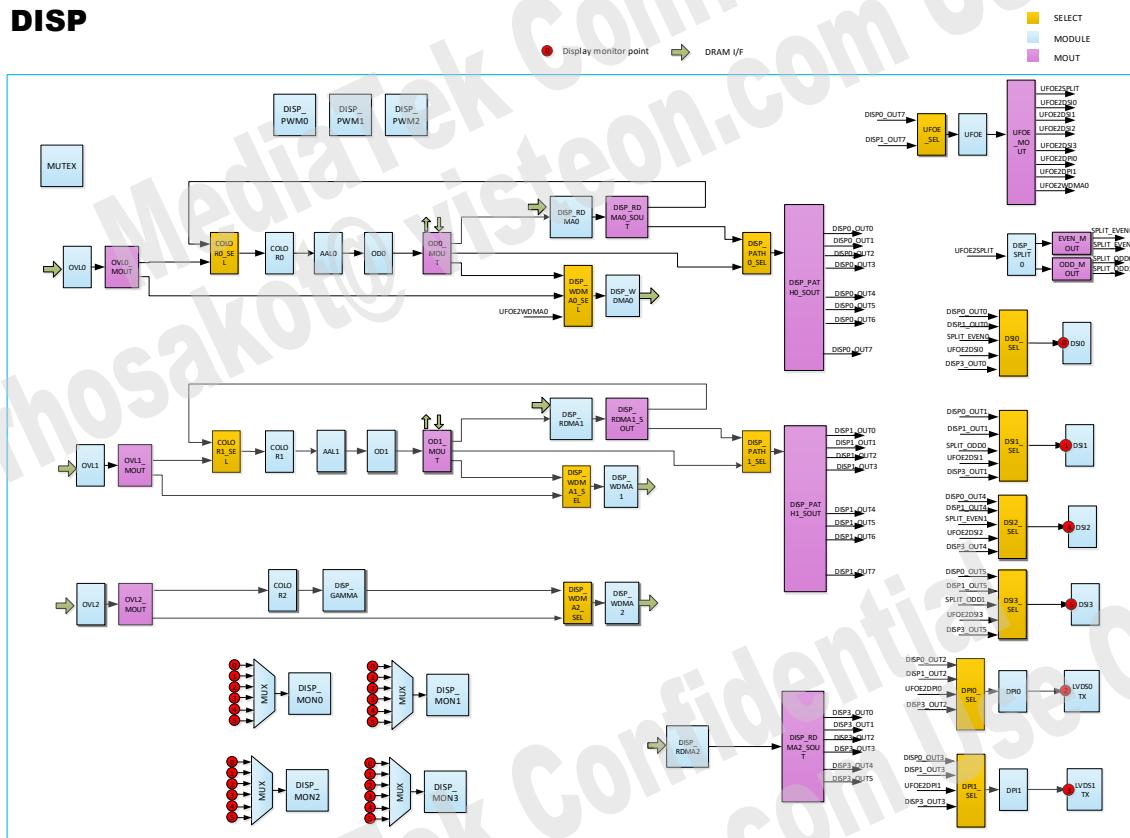
DISP

Figure 6-47 DISP Functional Blocks of Multimedia Partition

Figure 6-48 illustrates the functional blocks of IMGSYS_TOP.

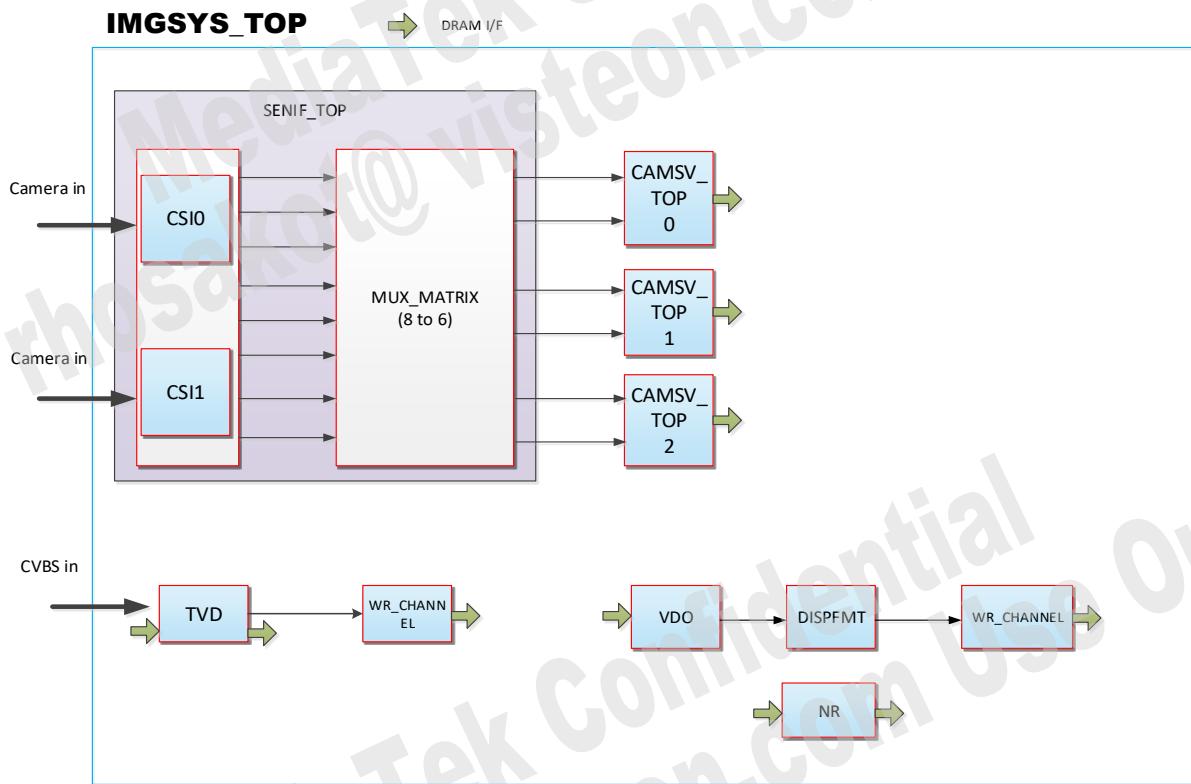


Figure 6-48 IMGSYS_TOP Functional Blocks

6.10.4 Register Definition

For register details, please refer to Chapter 4.9 of “MT2712 IVI Application Processor Registers”.

6.10.5 Programming Guide

Video Stream Programming Guide

Before introducing the programming model, two basic operation modes and one term “STREAM” must be explained. The basic operation modes are:

- Single mode

In this mode, one SW trigger only makes modules process one frame. Usually, it is a memory-in-memory-output operation (e.g. memory -> RDMA0 -> RSZ0 -> WDMA0 -> memory), or an operation output to the display device with frame buffer (e.g. memory -> OVL1 -> COLOR1 -> AAL1 -> DSI1). SW must wait for the processing of one frame to be done before triggering the next frame.

- Refresh mode

In this mode, pixels must be outputted to the display device without frame buffer (e.g. DSI video mode and DPI). After a data path is configured into refresh mode and starts, it will process frame-by-frame automatically until the data path stops. The process follows the refresh timing (vsync, hsync and data enable) of the display output interface.

In the refresh mode, the display output interface refresh timing is asynchronous to software, and users must guarantee each module receives complete settings for one frame when it starts to process this frame. Therefore, a mutex between software and hardware is used to achieve this.

A mutex is used to specify a STREAM and to guarantee a complete setting of this STREAM is seen by modules in this STREAM.

- A STREAM means a data stream from Source to Sink.
- Source can be any module with read memory capability, which is RDMA or OVL.
- Sink can be any module with capability of outputting to external display devices or to memory, which is DSIO/1, DPI or WDMA/WROT.

In MMSYS_TOP, there can be some concurrent STREAMs. Therefore, ten mutexes are provided to support this feature, with every mutex's function and register interface being the same.

Mutex has the following attributes, which can be set in disp_mutex.

- Source of Start of Frame (SOF)

It means which module is the Sink of this STREAM. Due to the display timing of a STREAM, when to start a frame/when to end a frame is decided by the Sink, and the display subsystem must know it to route the SOF signal to all modules in this mutex.

- Which modules are in STREAM.

Given one mutex describes one STREAM, there is a constraint that one module can be set in only one mutex. In other words, one module cannot be set in more than one mutex.

The programming sequence of a STREAM is shown as Figure 6-49.

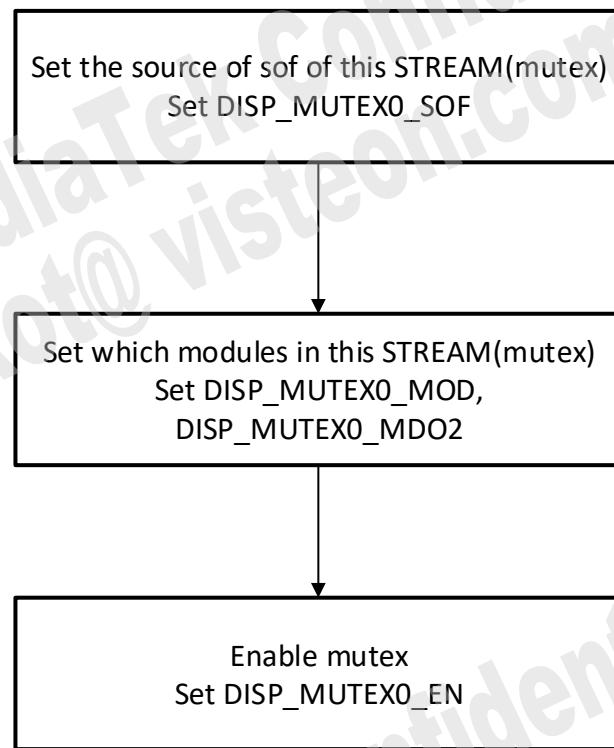


Figure 6-49 Stream Programming Sequence

Figure 6-50 shows the detailed waveform of mutex.

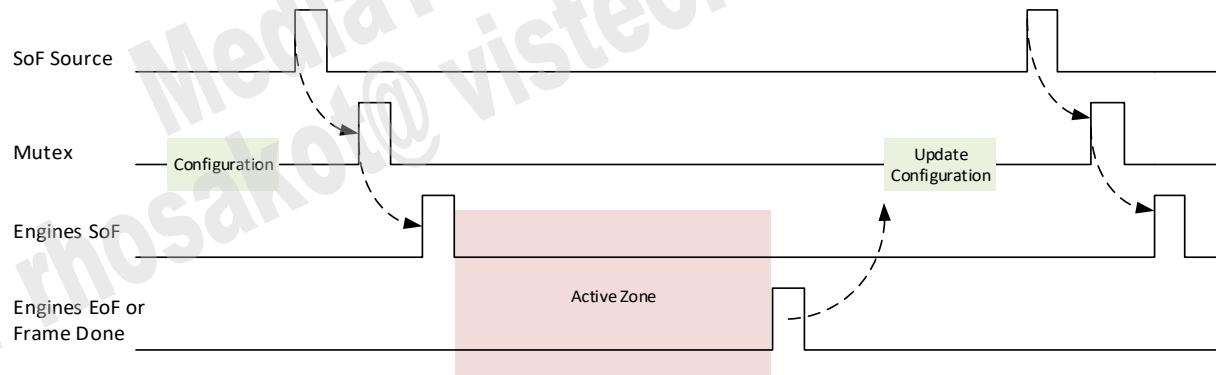


Figure 6-50 Detailed Mutex Waveform

Video Safety Programming Guide

The four Display Monitors support 24-bit pixel data input, and each monitor possesses a maximum of 12 independent regions.

Figure 6-51 depicts the `DISP_MONITOR` block diagram.

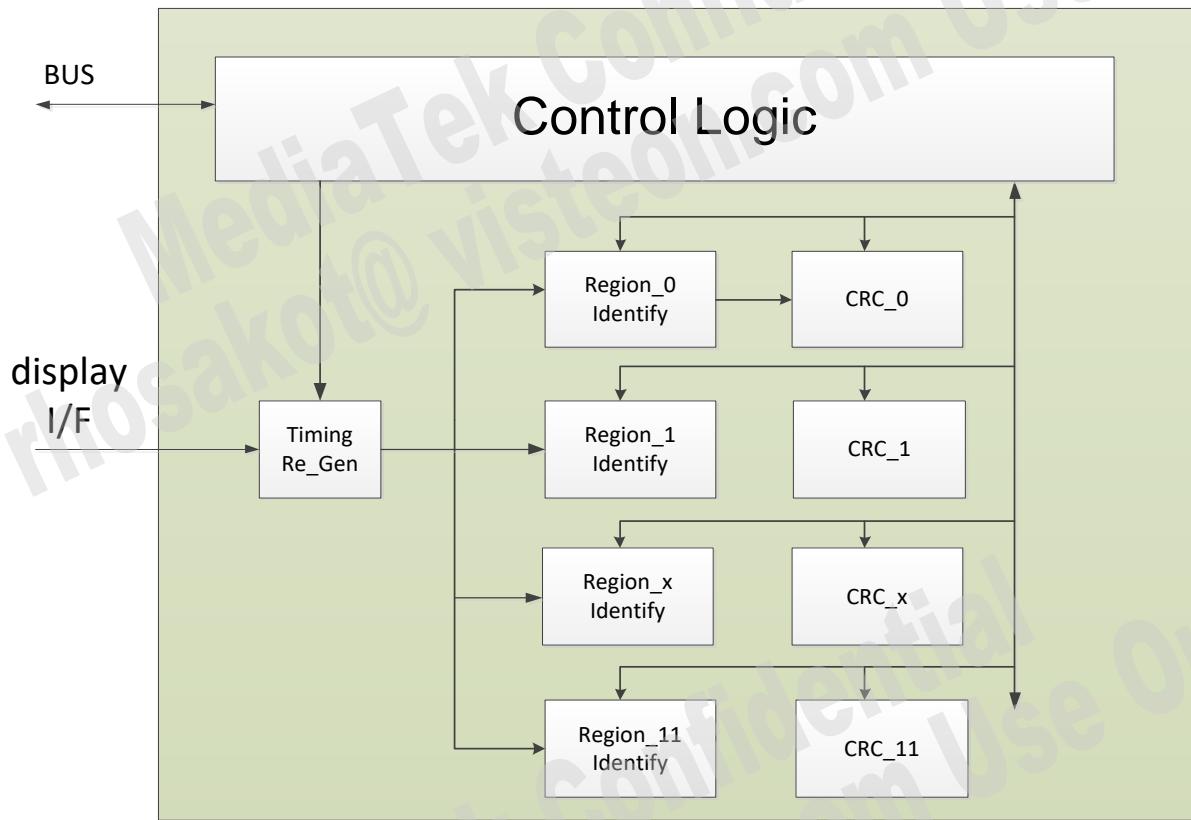


Figure 6-51 DISP_MONITOR Function Blocks

Timing Re_gen:

Generate the timing information (such as H/V counter) according to the input data and the resolution.

Region_x Identify ("x" indicates any value from 0 to 11):

To determine whether the coming data are inside the region.

CRC_x ("x" indicates any value from 0 to 11):

Compute the CRC value for active data (inside the region).

CRC is calculated using the 32-bit polynomial with an initial CRC Sum of 32'hFFFFFFF:

$\text{crc [31:0]} = 1 + x^1 + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32};$

Here x is 24 bits and the sequence from MSB to LSB is {R[7:0],G[7:0],B[7:0]}

Control Logic:

Control the IP behaviors such as: configuring the parameter, recording the status, enabling or disabling monitor, controlling interrupt, etc.

Following shows the basic flow to configure a monitor:

Step 1: Disable clock gate and release reset

Step 2: Select source engine

Step 3: Configure source size

Step 4: Configure region parameters

Step 5: Enable region

Step 6: Configure Interrupt Request (IRQ)

Step 7: Enable engine

Step 8: IRQ handle or poll the ready status

Step 9: Read the region CRC and continue ...

Note: When enabling the disp_monitor, please disable the “DITHER” function, because “DITHER” will cause data to be changed frame by frame.

Figure 6-52 shows the disp_monitor IRQ status distribution.

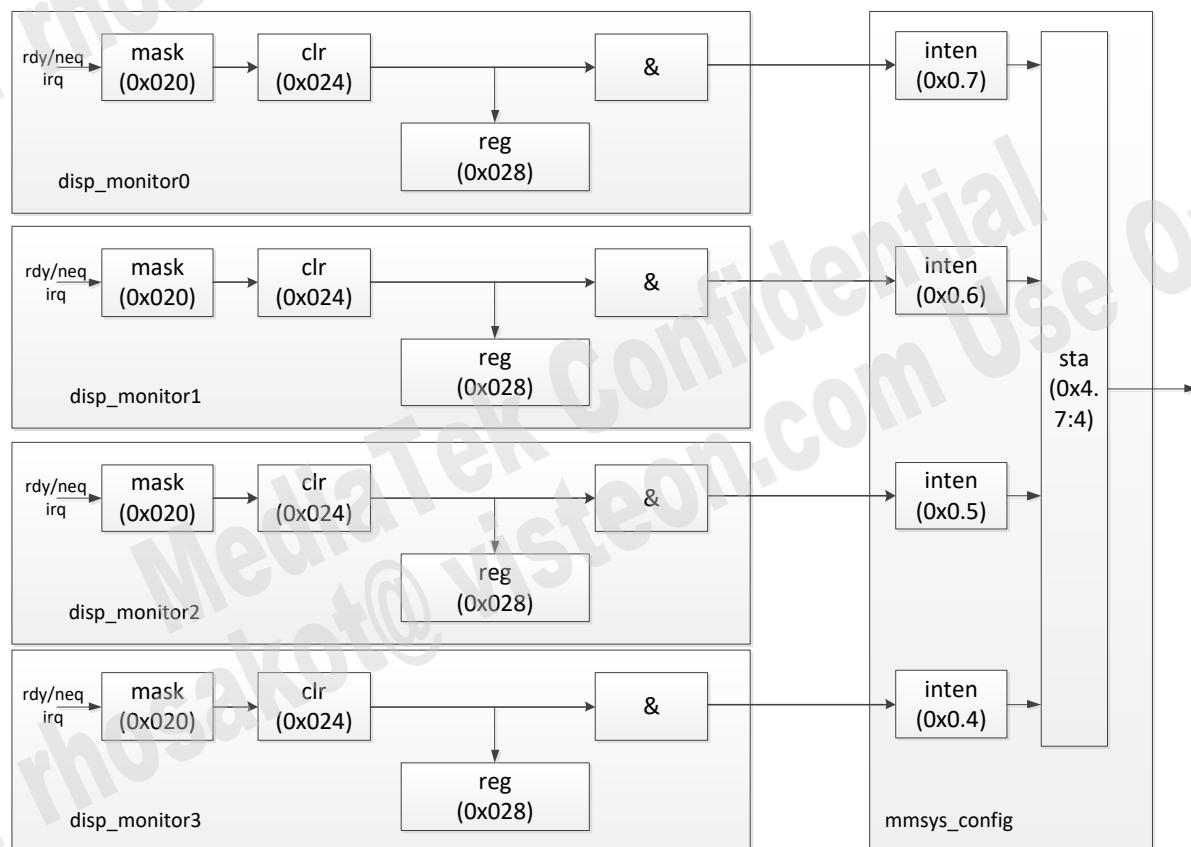


Figure 6-52 IRQ Status

Following is an example of the handle flow once IRQ happens:

1. Find the IRQ source engine (disp_monitor 0/1/2/3).
2. Find the sub IRQ source engine (region x in disp_monitor).
3. Clear the sub IRQ status and go to Interrupt Service Request (ISR).
4. Clear top IRQ status.
5. Exit IRQ handle routine.

Following diagram depicts the DISP_MONITOR clock relationship.

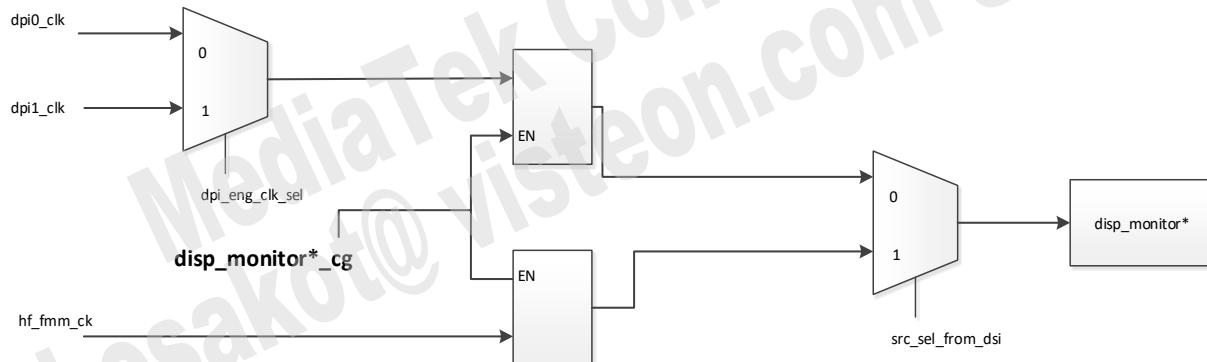


Figure 6-53 DISP_MONITOR Clock Diagram

Note:

1. disp_monitor* represents disp_monitor0/1/2/3.
2. dpi0_ck represents clock of dpi0's output domain; dpi1_ck represents clock of dpi1's output domain.
3. hf_fmm_ck is free-run.
4. disp_monitor*_cg, src_sel_from_dsi, and dpi_eng_clk_sel refer to the register description of DISP_MONITOR_CTRL in the Register Map.

As shown in Figure 6-53, the clock depends on the data source. If data come from DPI, the clock should also come from DPI; if data come from DSI, the clock source should also be from DSI.

When setting clock on, make sure the source module's clock is already on.

6.11 Multimedia Data Path Read DMA

6.11.1 Introduction

Multimedia Data Path Read DMA (MDP_RDMA) is used to read images of multiple source formats in memory. Output pixels are in the scan-line sequence to the following engine. It supports several functions, such as:

- Input image clipping
- Different input formats
- Color conversion between RGB and YUV
- Chroma upsampling with cosited or non-cosited YUV422/420 source

To support a larger image size with cost burden, tile mode scheme is applied in MDP. MDP_RDMA is fully tile mode-ready and can fulfill the single or tile mode operation. It also has a 3×3 color conversion inside. When the input is in RGB domain and output is in YCbCr domain, the 3×3 color conversion should be set to get the correct data.

6.11.2 Features

MDP_RDMA supports the following features:

- Multiple formats of input images
- YUV420/422 scanline 1/2/3 planes and RGB 16/24/32 bits
- Arbitrary byte swap for YUV or RGB source
- Cropping/clipping.
- Chroma upsampling filter to YUV444
- Default optimized or programmable RGB2YUV
- sblk format 10-bit mode w/ and w/o UFO compression.
- Tile mode-ready, and support source width up to 131072 pixels.
- Direct link to Resizer (RSZ) with YUV 444

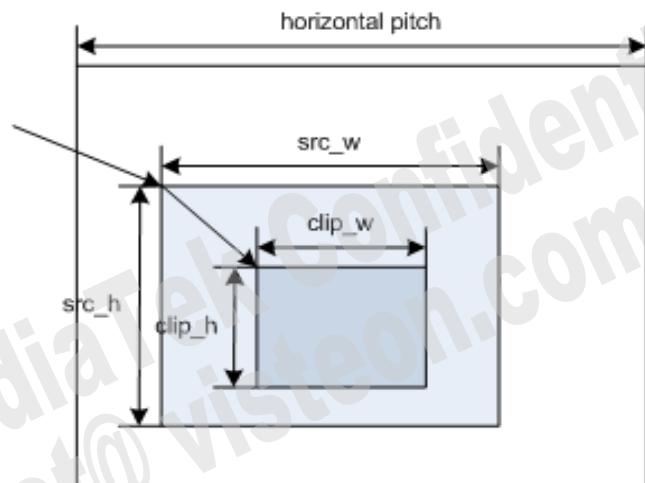


Figure 6-54 Support Clipping from Source Frame Buffer

6.11.3 Block Diagram

The internal pipeline of MDP_RDMA is shown in Figure 6-55. The input is from external memory via SMI read port, and output is directly linked to the next engine depending on the display system's data-path setting, like RSZ. There is also an APB interface for SW control.

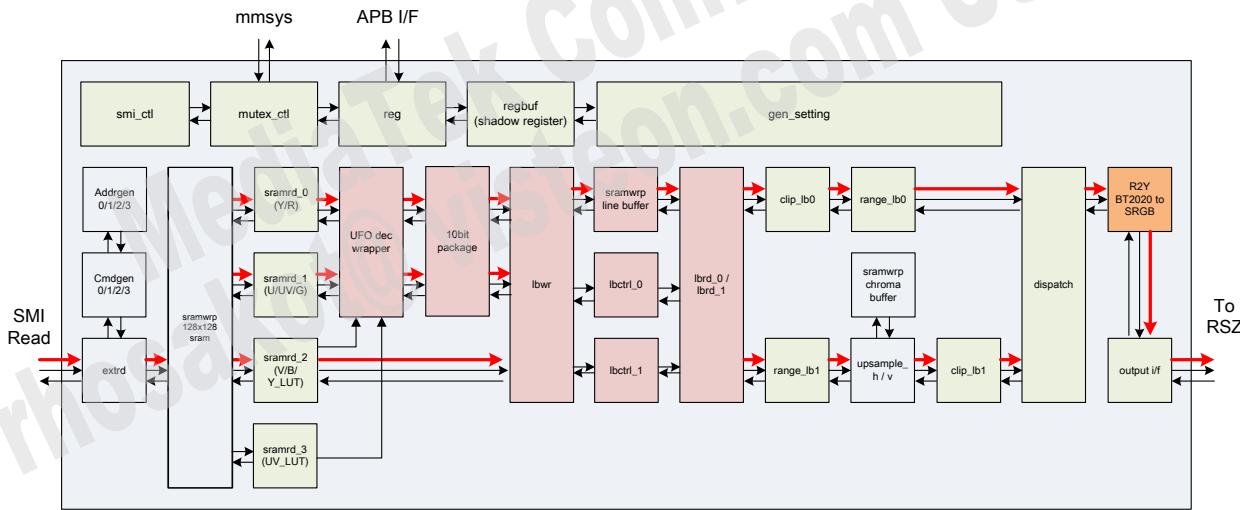


Figure 6-55 MDP_RDMA Block Diagram

6.11.4 Register Definition

For register details, please refer to Chapter 4.10.1 of “MT2712 IVI Application Processor Registers”.

6.11.5 Programming Guide

6.11.5.1 SW Flow

To enable MDP_RDMA to read a frame, SW may program in the following sequence. Take the example, with an YCbCr_420_P_SW 1920×1080 frame provided in DRAM. MDP_RDMA will try to clip it into 256×256 as the following example setting:

1. Release reset and enable all interrupts.
 $\text{DISP_ROT_RESET} = 0x0000_0000$
 $\text{DISP_ROT_INTERRUPT_ENABLE} = 0x0000_0007$
2. Set controls to determine the behavior of MDP_RDMA.
 $\text{DISP_ROT_CON} = 0x0000_0000$
3. Set SMI configuration for performance issues.
 $\text{DISP_ROT_GMCIF_CON} = 0x0000_1771$
4. Set input format.
 $\text{DISP_ROT_SRC_CON} = 0x0000_0000$
5. Set base address 0 for first plane of the source frame. If the source frame contains more than one plane, then the base address 1 and base address 2 should be set as well. Note that the offset of starting address is relative to operations, please refer to the Register Map for more information.
 $\text{DISP_ROT_SRC_BASE_0} = 0x0007_7880$

- DISP_ROT_SRC_BASE_1 = 0x0080_6c40
DISP_ROT_SRC_BASE_2 = 0x00A0_1040
6. Calculate the byte number of the line pitch. It is the distance from the first byte of the current line to the first byte of the next line. The value for sub frame is required for multiple planar sources.
DISP_ROT_MF_BKGD_SIZE_IN_BYTE = 0x0000_0780
DISP_ROT_SF_BKGD_SIZE_IN_BYTE = 0x0000_0780
 7. Set the width and the height of the source frame before it is rotated.
DISP_ROT_MF_SRC_SIZE = 0x0100_0100
 8. Set the clip size if necessary.
DISP_ROT_MF_CLIP_SIZE = 0x0100_0100
 9. Set offset in source format if necessary.
DISP_ROT_MF_OFFSET_1 = 0x0000_0000
 10. Set up the color transform options and parameters if necessary.
DISP_ROT_TRANSFORM_0 = 0x0000_0000
 11. Assert the ROTEN to enable MDP_RDMA to wait shadow register update.
DISP_ROT_EN = 0x0000_0001
 12. Assert register update to start MDP_RDMA's operation. This control signal comes from Display Mutex.

To enable 10-bit mode, something should be noticed.

1. BPP = 1.25, beware of address offset/pitch.
2. Pitch setting for 10-bit mode:
 - a. MF_BKGD_WB_IN_BYTE = 40 * BKGD_SRC_W
 - b. SF_BKGD_SB_IN_BYTE = 20 * BKGD_SRC_W
3. Offset setting for 10-bit mode:
 - a. offset = offset_ori * 5/4
4. Additional register to be programmed:
 - a. BKGD_SIZE_IN_PXL = BKGD_SIZE_IN_BYTE*4/5
 - b. SRC_OFFSET_0_P = SRC_OFFSET_0*4/5

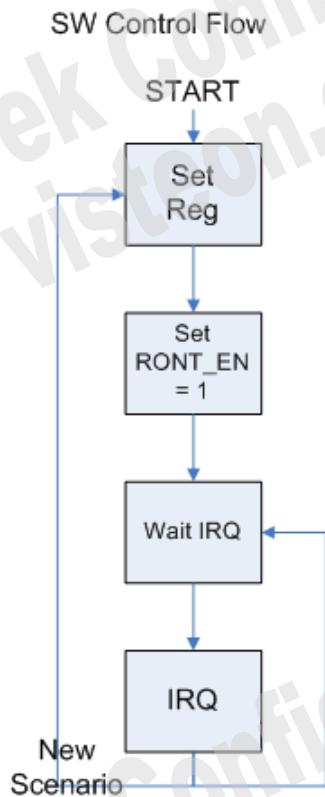


Figure 6-56 SW Control Flow

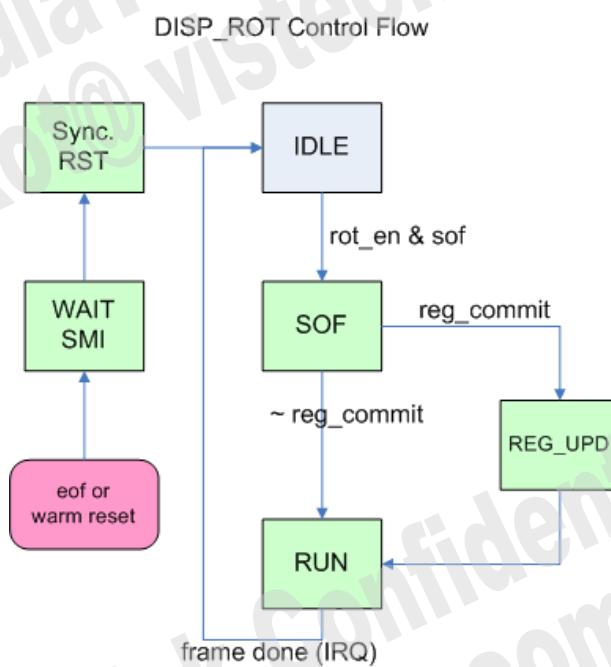


Figure 6-57 HW FSM and Mutex Control Flow

6.12 Multimedia Data Path Resizer (MDP_RSZ)

6.12.1 Introduction

There are three resizer modules for MT2712 MDP: MDP_RSZ0, MDP_RSZ1 and MDP_RSZ2. The algorithm of each may be slightly different for different requirements. Table 6-13 lists the corresponding algorithm specifications. Table 6-12 shows the functional specifications of MDP_RSZ0, MDP_RSZ1 and MDP_RSZ2. MDP_RSZ0 and MDP_RSZ1 scale for multiple purposes in MDP 2.0 structure, mainly including generating image for display, video codec, and jpeg codec. MDP_RSZ2 is dedicated for display. Digital zoom is accomplished by MDP_RSZ0 or MDP_RSZ1 in MDP 2.0. MDP_RSZ0 and MDP_RSZ1 support YUV444 input and YUV444 output.

Table 6-13 shows the hardware specifications of MDP_RSZ0, MDP_RSZ1 and MDP_RSZ2. They support three scaling algorithms, including 4-tap finite impulse response (FIR), 4n-tap cubic accumulation and n-tap source accumulation. The difference between MDP_RSZ0, MDP_RSZ1 and MDP_RSZ2 is the maximum image width. MDP_RSZ0 and MDP_RSZ1 supports up to 768 in image width, and MDP_RSZ2 supports max. 544.

Table 6-12 Resizer Functional Specifications

	MDP_RSZ0	MDP_RSZ1	MDP_RSZ2
Input data format	8-bit YUV444 (unsigned)	8-bit YUV444 (unsigned)	8-bit YUV444 (unsigned)
Output data format	8-bit YUV444 (unsigned)	8-bit YUV444 (unsigned)	8-bit YUV444 (unsigned)
Scaling ratio	Between 1/128x and 64x	Between 1/128x and 64x	Between 1/128x and 64x
Crop function	Supported	Supported	Supported
Function	1. Digital zoom 2. MDP general-purpose resizing	1. Digital zoom 2. MDP general-purpose resizing	1. Digital zoom 2. MDP general-purpose resizing
Supported width (OTF)	N/A	N/A	N/A
Supported width (tile mode)	6-tap: 768 n-tap: 768 4n-tap: 384	6-tap: 768 n-tap: 768 4n-tap: 384	6-tap: 544 n-tap: 544 4n-tap: 272

Table 6-13 Resizer Hardware Specifications

	MDP_RSZ0	MDP_RSZ1	MDP_RSZ2
Luma interpolation method (H)	(1) 6-tap FIR (2) 4n-tap CA (3) n-tap SA	(1) 6-tap FIR (2) 4n-tap CA (3) n-tap SA	(1) 6-tap FIR (2) 4n-tap CA (3) n-tap SA
Luma interpolation method (V)	(1) 6-tap FIR (2) 4n-tap CA (3) n-tap SA	(1) 6-tap FIR (2) 4n-tap CA (3) n-tap SA	(1) 6-tap FIR (2) 4n-tap CA (3) n-tap SA
Chroma interpolation method (H)	(1) 6-tap FIR (2) 2n-tap TA (3) n-tap SA	(1) 6-tap FIR (2) 2n-tap TA (3) n-tap SA	(1) 6-tap FIR (2) 2n-tap TA (3) n-tap SA

	MDP_RSZ0	MDP_RSZ1	MDP_RSZ2
Chroma interpolation method (V)	(1) 6-tap FIR (2) 2n-tap TA (3) n-tap SA	(1) 6-tap FIR (2) 2n-tap TA (3) n-tap SA	(1) 6-tap FIR (2) 2n-tap TA (3) n-tap SA
Interpolation order	H->V (xn-tap) V->H (6-tap)	H->V (xn-tap) V->H (6-tap)	H->V (xn-tap) V->H (6-tap)
Line buffer (frame mode)	NA	NA	NA
Line buffer (tile mode)	384x48x6 (Logical) -> 192x96x6	384x48x6 (Logical) -> 192x96x6	272x48x6 (Logical) -> 136x96x6

6.12.2 Features

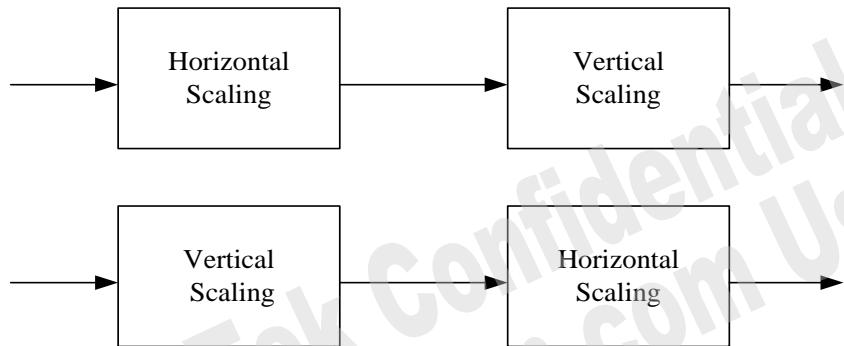


Figure 6-58 Separate 1D FIR Operations

Basically, the rescaling procedure is composed of two separate 1D FIR operations (see Figure 6-58). There are three major types of 1D FIR operation, namely, 6-tap FIR, 4n-tap cubic accumulation and n-tap source accumulation. These three algorithms have different rescaling characters and tap numbers. 6-tap is suitable for up-scaling and down-scaling ($1x \sim 1/2x$). It is a fixed 6-tap FIR operation and needs 18-line buffer to perform vertical scaling operation.

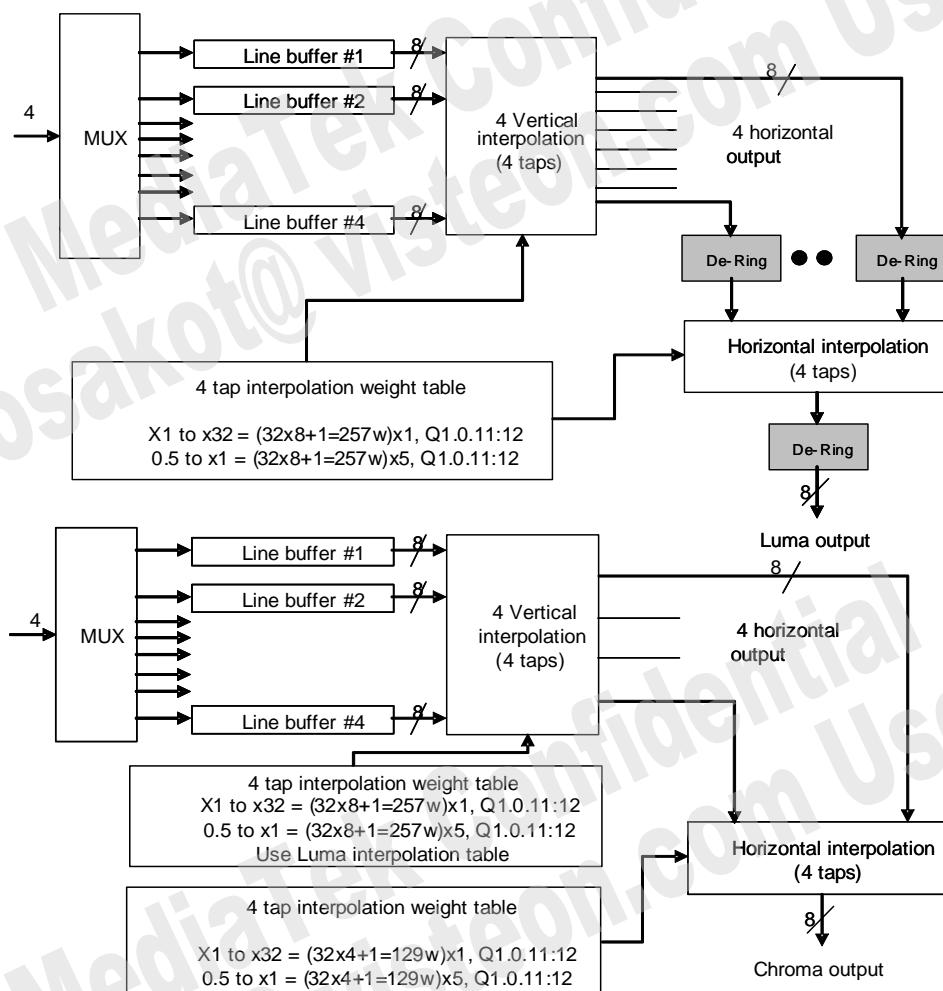


Figure 6-59 4-Tap Cubic Block Diagram

4n-tap cubic accumulation has a better quality than 6-tap FIR when the down-scaling ratio is bigger than 1/2x. It is essential that a variable tap FIR operation and its tap number is determined by scaling ratio (4 times n; n is scaling ratio). For example, when the scaling ratio is 1, the tap number is 4 (see Figure 6-60). When the scaling ratio is 1/2, the tap number is 8 (see Figure 6-61). In addition, n-tap source accumulation is also a variable tap number FIR algorithm. It has poor sharpness and is an extremely low-cost solution.

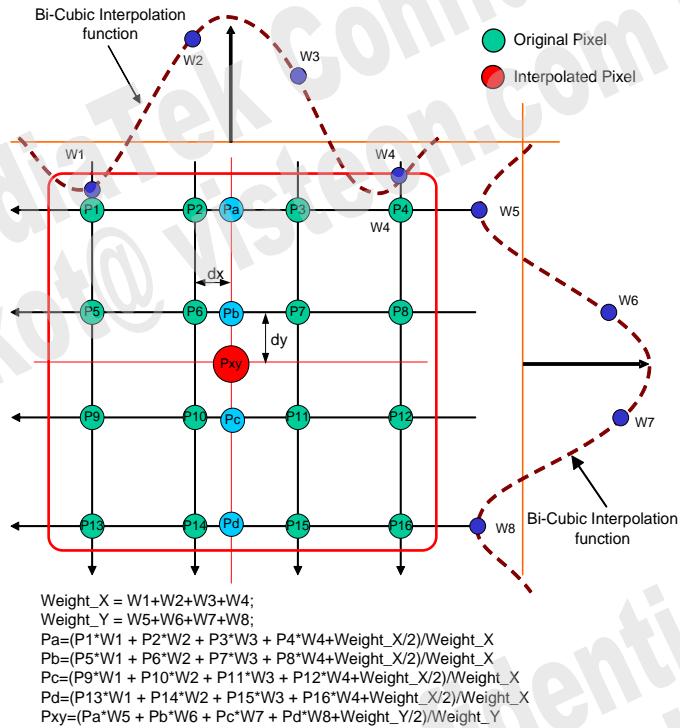


Figure 6-60 Cubic Accumulation (scaling ratio = 1x)

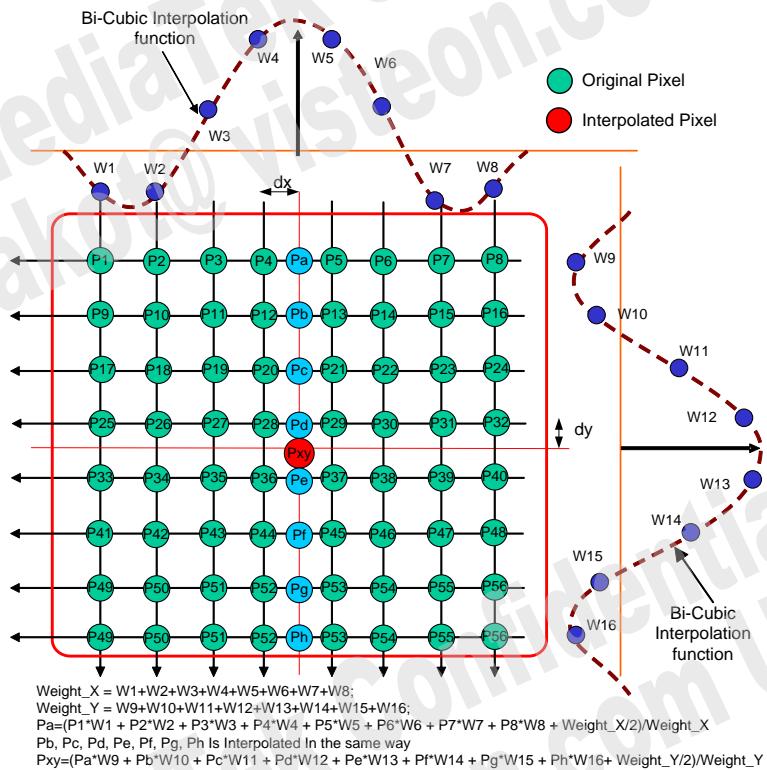


Figure 6-61 Cubic Accumulation (scaling ratio = 1/2x)

6.12.3 Block Diagram

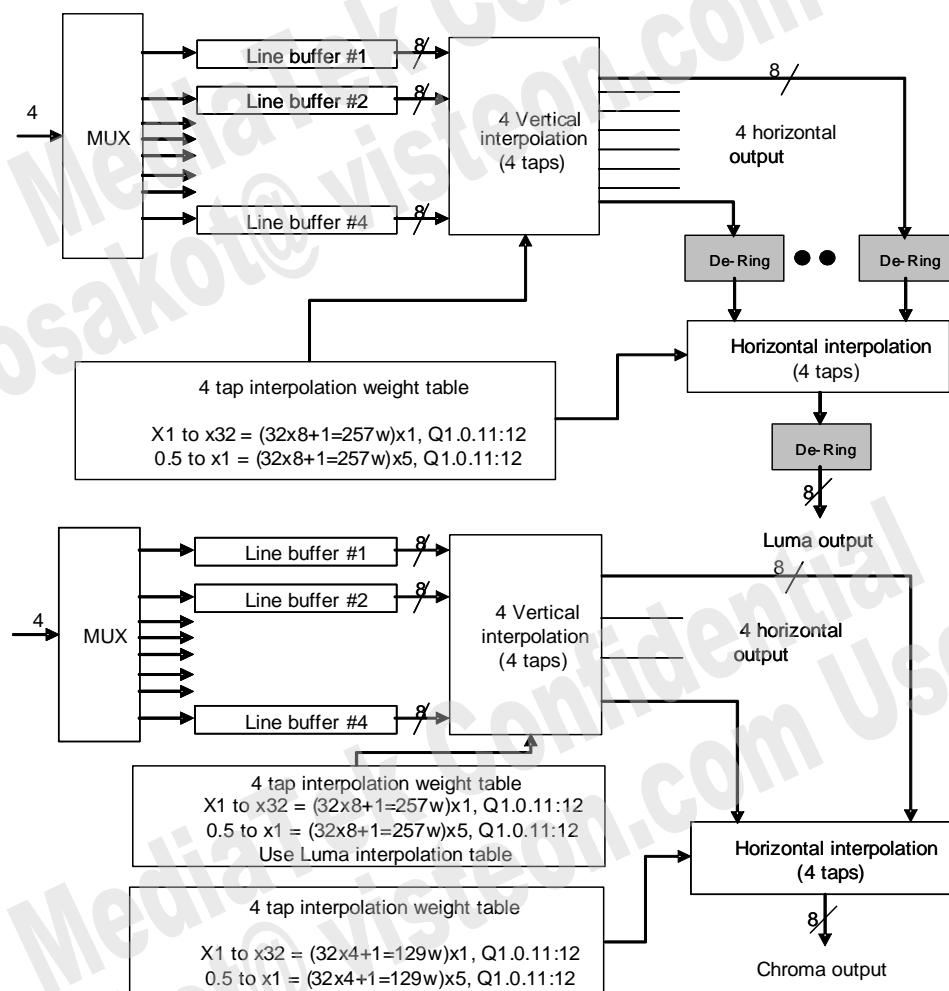


Figure 6-62 MDP_RSZ Block Diagram

6.12.4 Register Definition

For register details, please refer to Chapter 4.10.2 of “MT2712 IVI Application Processor Registers”.

6.12.5 Programming Guide

6.12.5.1 SW Flow

Suggested Algorithm

Corresponding to applied scaling ratio, following is the suggested algorithm for the resizer to choose from. The algorithm is suggested for the best quality. The principles are:

6-tap: 32x~1x

6n-tap: $1x \sim 1/64x$

n-tap: $1x \sim 1/128x$

However, the suggested algorithm may contradict with the limited supporting size in each resizer engine. Take CDRZ for example, as described in Figure 6-63. If a certain application needs to use CDRZ to resize from 3000 to 1400, it is suggested to use the 6n-tap cubic accumulation algorithm. However, the supported size of 6n-tap in CDRZ is 1152, which is smaller than 1400. Therefore, in this case, choose n-tap to be applied.

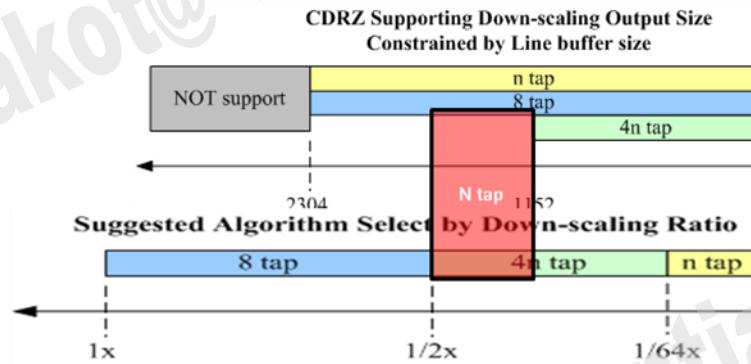


Figure 6-63 Suggested Algorithm vs. Supported Size

Coefficient Step Calculation

The coefficient step is calculated in different ways for different algorithms. Refer to the descriptions below.

6-tap:

```
Unit base: 32768 (2^15)
M_m1 = Input_size - In_Int_Offset - (In_Sub_Offset==0? 0: 1)
N_m1 = Output_size
Cstep = (M_m1*UNIT+(N_m1>>1))/N_m1
```

6-tap:

Unit base: **65536** (2^{16}) (note the offset settings)

6n-tap:

```
Unit base: 1048576 (2^20)
M_m1 = Input_size - In_Int_Offset - (In_Sub_Offset==0? 0: 1)
N_m1 = Output_size
Cstep = (N_m1*UNIT+(M_m1))/M_m1
Transfer offset from input to output
```

N-tap:

```
Unit base: 1048576 (2^20)
M_m1 = Input_size - In_Int_Offset - (In_Sub_Offset==0? 0: 1)
N_m1 = Output_size
Cstep = (N_m1*UNIT+(M_m1))/M_m1
Transfer offset from input to output
```

Input/Output Offset Transfer

For 6-tap interpolation, set “Input Offset” into register.

Set register Int_Ofst = In_Int_Ofst;

Set register Sub_Ofst = In_Sub_Ofst;

For Accumulation, set “Output Offset” into register.

Transfer Input Offset to Output Offset

Out_Int_Ofst = (In_Int_Ofst*Cstep + In_Sub_Ofst*Cstep/Unit)/Unit

Out_Sub_Ofst = (In_Int_Ofst*Cstep + In_Sub_Ofst*Cstep/Unit) % Unit

Table Selection

6-tap: (up scaling only)

Table 10

6n-tap (downscaling only)

10 is recommended.

N-tap (downscaling only): No table selection required

The entire programming guide includes the following steps:

1. Decide input/output size.
2. Check “Maximum Size” constraint.
3. Calculate the ratio.
4. Choose Algorithm from “Suggested Algorithm”.
5. Check “Supporting Size” constraint.
6. Trigger Resizer and wait for IRQ.

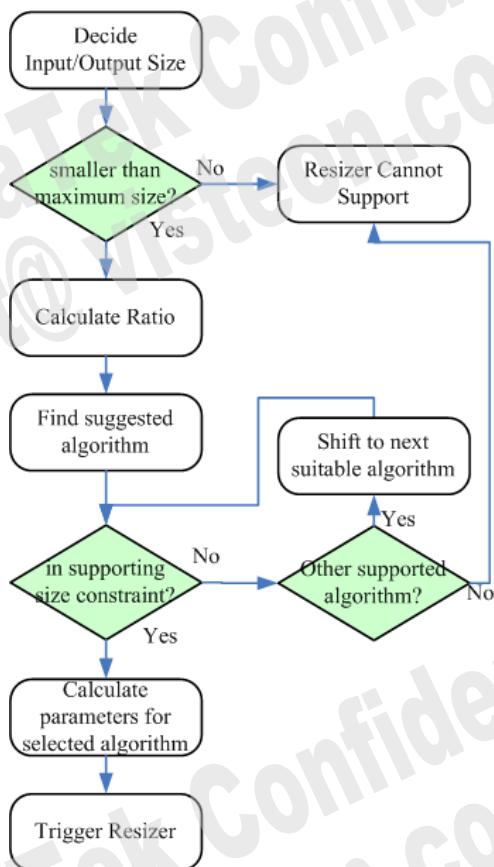


Figure 6-64 Programming Guide

6.13 Multimedia Data Path 2D Sharpness

6.13.1 Introduction

Multimedia Data Path 2D Sharpness (MDP_TDSHP) provides a better picture quality for panel display. It restores the image details, sharpens the edge and provides a vivid touch for pictures and videos.

6.13.2 Features

MDP_TDSHP supports the following features:

- 2-dimensional sharpness filter
- Peaking by color (PBC)

The 2D filters are used to extract middle/high-frequency components from the input signal. Each extracted AC component is enhanced individually and then added back to the input signal. The enhancement units are composed of coring, gain, limit and clip.



Before 2-dimensional sharpness

After 2-dimensional sharpness

Figure 6-65 Visual Effect of 2-dimensional Sharpness

Sometimes users may prefer different sharpness levels in different color tones. PBC is used for such preference. PBC detects at most three different colors and applies different sharpness levels to them.



Before green PBC

After green PBC

Figure 6-66 Visual Effect of Peaking by Color

6.13.3 Block Diagram

Figure 6-67 is the block diagram of display 2D sharpness. The First-In First-Out (FIFO) unit is used to pre-fetch pixel data, and thus the overall throughput can be improved in some situations.

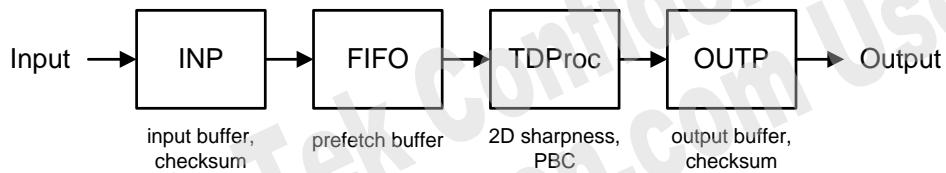
**Figure 6-67 Display 2D Sharpness Block Diagram**

Figure 6-68 is the sharpness core block diagram. The luminance data will be processed and modified in the sharpness function. The chrominance data will flow into PBC, but there will be no modification at the output.

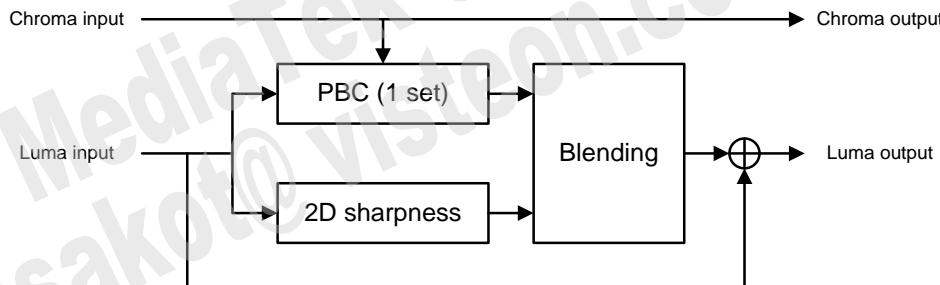


Figure 6-68 Sharpness Core Block Diagram

6.13.4 Register Definition

For register details, please refer to Chapter 4.10.3 of “MT2712 IVI Application Processor Registers”.

6.13.5 Programming Guide

6.13.5.1 SW Flow

Sharpness Adjustment

There are a total of three possible usage scenarios for different situations.

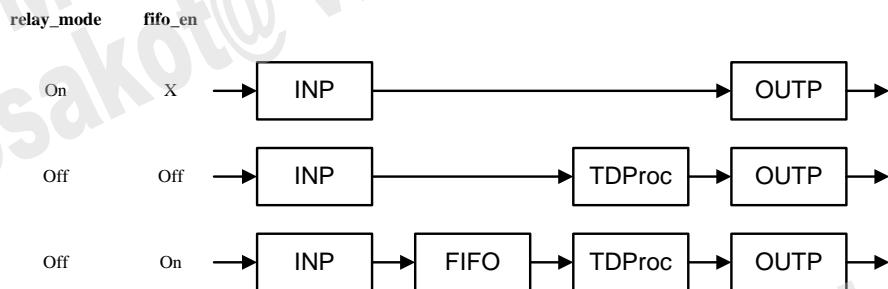


Figure 6-69 Usage Scenarios

The sharpness function is designed to provide different sharpness levels. The sharpness level can be programmed by the gain curve control in the 2-dimensional sharpness unit. A higher sharpness level enhances more details of the image and sharpens the edge. However, sharpness setting that is too strong will induce some artificial side effects. Therefore, a balanced setting with an adequate sharpness level is preferred.

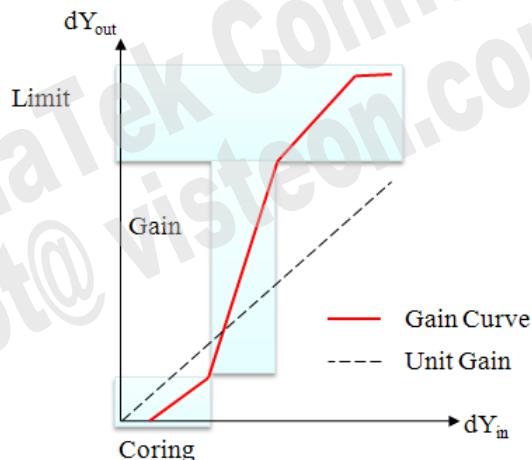


Figure 6-70 Gain Curve Control of Sharpness

Luma Adjustment

The color processor applies a luma mapping curve to the input luma (see Figure 6-71). The full luma range is equally divided into 16 segments, and registers Y_FTN_* are responsible for the adjustments.

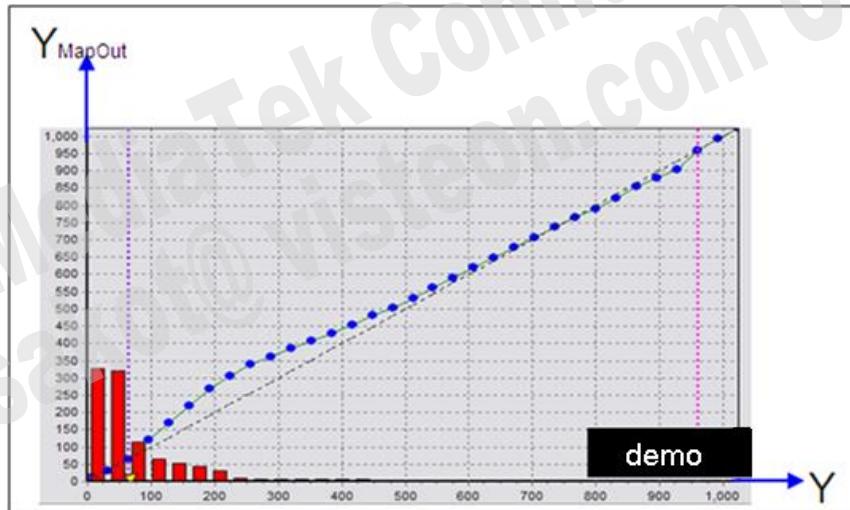


Figure 6-71 Demo Curve Control of Luma Adjustment

Register programming sequence

1. Set MDP_TDSHP_INPUT_SIZE
2. Set MDP_TDSHP_OUPUT_OFFSET
3. Set MDP_TDSHP_OUPUT_SIZE
4. Set MDP_TDSHP_CFG to configure relay_mode, fifo_en, and fifo_post
5. Set MDP_TDSHP_00 to configure tds_en, adap_luma
6. Set TDSHP settings
7. Set MDP_TDSHP_CTRL.TDSHP_CTRL_EN to enable MDP_TDSHP

6.14 Multimedia Data Path Write DMA

6.14.1 Introduction

Multimedia Data Path Write DMA (MDP_WDMA) and Display Write DMA (DISP_WDMA) have the same hardware architecture, but slightly different SRAM sizes, which influences the line-width support and outstanding ability. WDMA performs the job of DMA writing out the data in display/mdp pipeline into DRAM. In the following sections, WDMA is called DISP_WDMA.

6.14.2 Features

MDP_WDMA supports the following features:

- Dither
- Programmable parameter color transform
- Input color format YUV444/RGB888
- Output format RGB565/RGB888/ARGB8888/UYVY/YV12/NV12/NV21
- 3-tap filter in horizontal and 2-tap filter in vertical for YUV420 downsampling
- Byte swap/color swap/UV swap

6.14.3 Block Diagram

DISP_WDMA has a 256×128 two-port SRAM for DMA FIFO. One 320×64 single-port SRAM is for vertical filtering line_buffer (2,560 pixels). For line-width larger than 2560, YUV420 vertical downsampling will fall back to the drop-pixel mode.

MDP_WDMA has a 128×128 two-port SRAM for DMA FIFO. One 64×64 single-port SRAM is for vertical filtering line_buffer (512 pixels). For line-width larger than 512, YUV420 vertical downsampling will fall back to the drop-pixel mode.

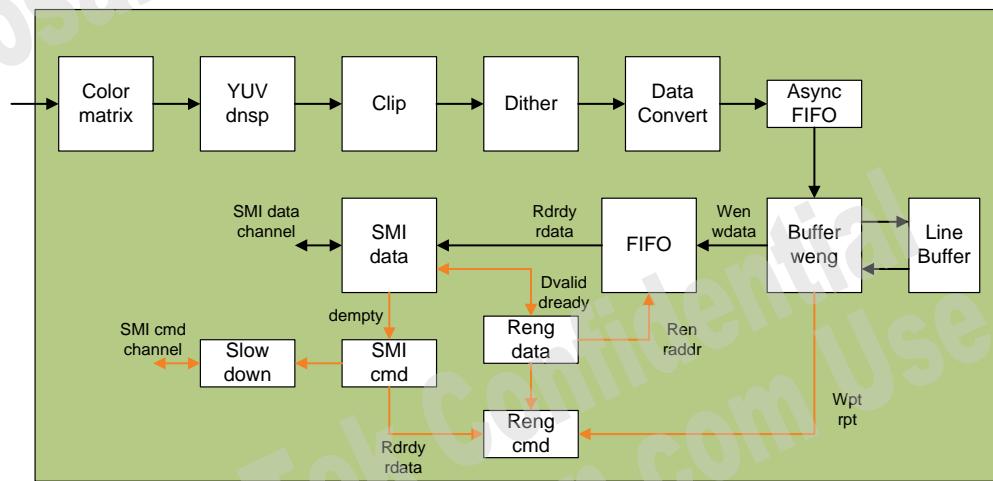


Figure 6-72 DISP_WDMA Block Diagram

6.14.4 Register Definition

For register details, please refer to Chapter 4.10.4 of “MT2712 IVI Application Processor Registers”.

6.14.5 Programming Guide

6.14.5.1 SW Flow

The fundamental key parts are base address/input output format/data strides in memory. If color transform is needed, color transform matrix will also be needed. Other settings such as dither/filter are optional.

MUST program

WDMA_CFG
WDMA_SRC_SIZE
WDMA_CLIP_SIZE
WDMA_CLIP_COORD
WDMA_DST_ADDR0
WDMA_DST_W_IN_BYTE
WDMA_DST_ADDR1
WDMA_DST_ADDR2
WDMA_DST_UV_PITCH
WDMA_DST_ADDR_OFFSET0
WDMA_DST_ADDR_OFFSET1
WDMA_DST_ADDR_OFFSET2

The read result of WDMA_CT_DBG means the input counter of the WDMA.

The MSB part is line counter and the LSB part is pixel counter.

The read result of WDMA_FLOW_CTRL_DBG means the current state of WDMA.

The state machine of the WDMA is as follows:

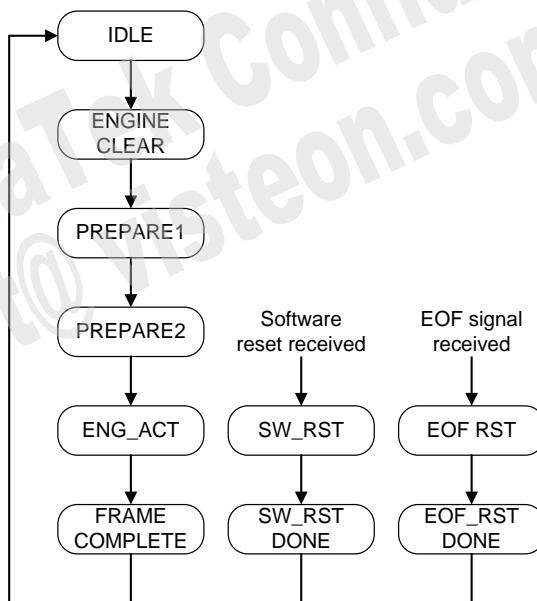


Figure 6-73 Typical MDP_WDMA Programming Sequence

6.15 Multimedia Data Path Rotation DMA

6.15.1 Introduction

Multimedia Data Path Rotation (MDP_WROT) DMA is a write rotate DMA agent supporting 8 rotation/flip options. Users may hold the phone in any orientation, so it is necessary to rotate the image in the correct direction.

6.15.2 Features

MDP_WROT supports the following features:

- Rotation Angles: 0°, 0° + H_Flip, 90°, 90° + H_Flip, 180°, 180° + H_Flip, 270°, and 270° + H_Flip as illustrated in Figure 6-74.
- Format and Footprint: YUV422 1/2/3 planes, YUV420 2/3 planes, RGB888, ARGB8888, RGB565, Y only
- Programmable RGB color matrix
- Dither engine

6.15.3 Block Diagram

Figure 6-74 illustrates the engine architecture of MDP_WROT, including color matrix, dither, sub-sampling, rotator and DMA engines.

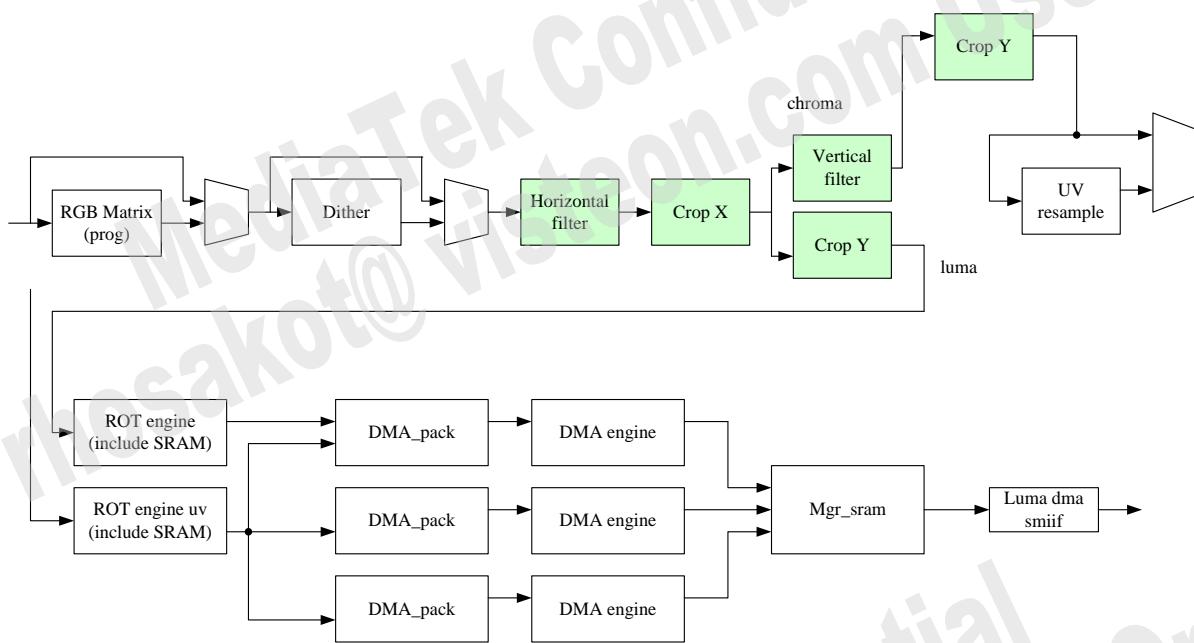


Figure 6-74 MDP_WROT Block Diagram

6.15.4 Register Definition

For register details, please refer to Chapter 4.10.5 of “MT2712 IVI Application Processor Registers”.

6.15.5 Programming Guide

6.15.5.1 SW Flow

Firmware Settings for Rotation/Flip

To output data to DRAM with rotation/flip, MDP_WROT should calculate the correct address to put the first data. Then the next data can be written in a certain order which also depends on rotation/flip settings.

However, the circuit for calculating the address to put the first data is only applied at the beginning of the entire image or tile. For cost effectiveness, the address offset for the first data compared to BASE_ADDR can be calculated by firmware and transferred into hardware by register OFST_ADDR. For all kinds of rotation/flip settings, the required positions of OFST_ADDR are different. The illustrations for the position of OFST_ADDR on DRAM footprint are shown in Figure 6-75 to Figure 6-78. Note that the xsize and ysize are defined according to input data’s scan-line direction. The direction of xsize is parallel to the input scan-line direction, and the direction of ysize is perpendicular to the input scan-line direction. The stride setting is defined according to DRAM footprint, which is the direction after rotation/flip.

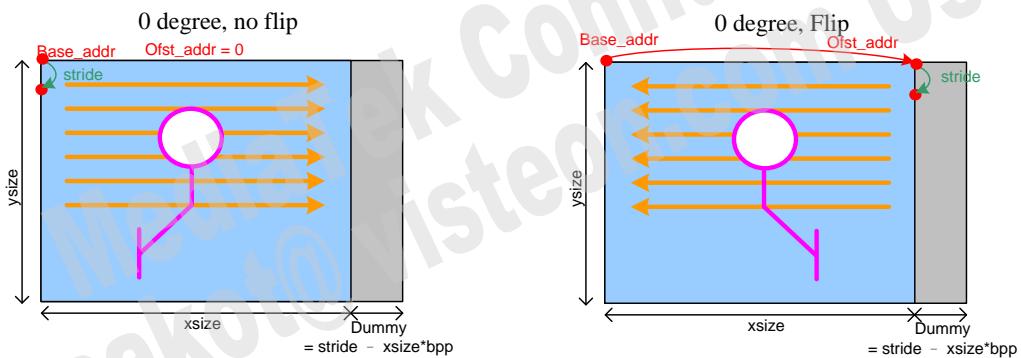


Figure 6-75 Firmware Settings of OFST_ADDR in 0° Rotation (scan-line)

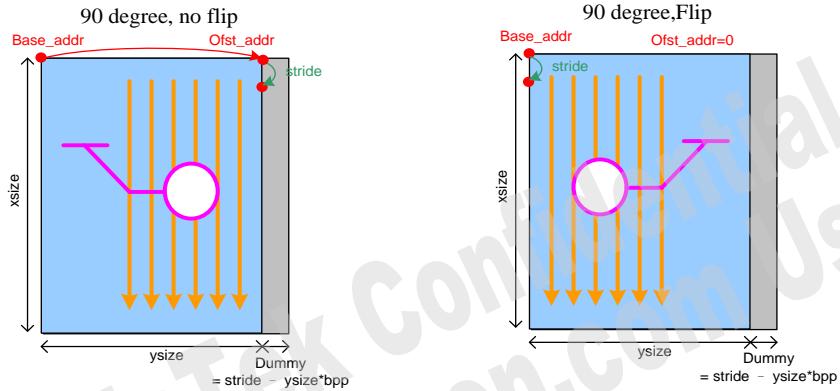


Figure 6-76 Firmware Settings of OFST_ADDR in 90° Rotation (scan-line)

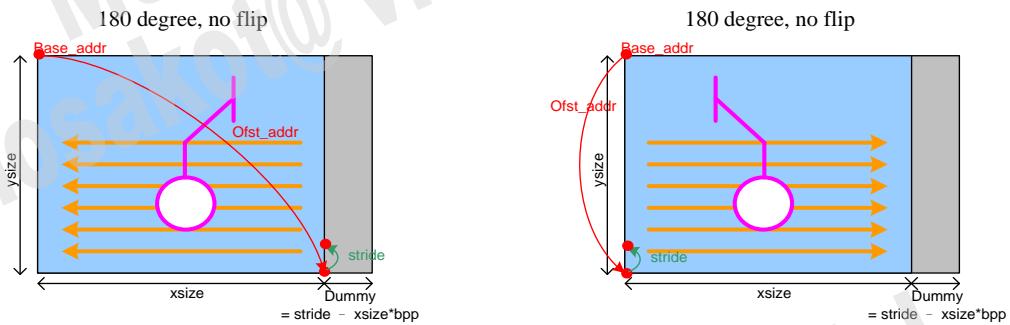


Figure 6-77 Firmware Settings of OFST_ADDR in 180° Rotation (scan-line)

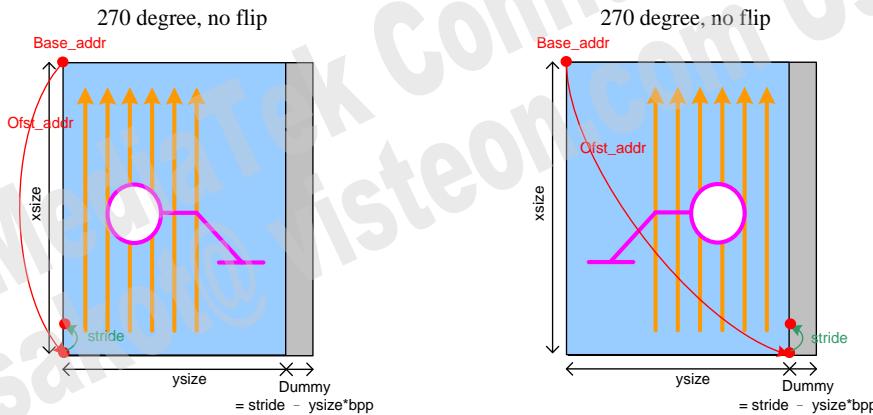


Figure 6-78 Firmware Settings of OFST_ADDR in 270° Rotation (scan-line)

Summary is given in Table 6-14.

Table 6-14 MDP_WROT_OFST_ADDR Settings for Rotation/Flip

Rotation	No Flip (Flip = 0)	Horizontal Flip (Flip = 1)
0 degree (Rotation=0)	0	Xsize-1
90 degree (Rotation=1)	Ysize-1	0
180 degree (Rotation=2)	Stride*(Ysize-1) + (Xsize-1)	Stride*(Ysize-1)
270 degree (Rotation=3)	Stride*(xsize-1)	Stride*(Xsize-1) + (Ysize-1)

For an interpolation-based UV-resampler, which can perform better quality for YUV420, VIDO UVSEL should be set corresponding to different rotation/flip settings. According to Table 6-15, the suggested settings provide the proper chroma sampled points.

Table 6-15 VIDO UV SEL for YUV420 Format Cooperated with CRSP

Rotation	Flip	UV_SELX	UV_SELY
0	0	0	1
0	1	1	1
1	0	0	0
1	1	0	0
2	0	1	1
2	1	0	1
3	0	0	1
3	1	0	1

Working Buffer Height Setting

Setting up the buffer width and buffer height for MDP_WROT is necessary, no matter whether the rotation is enabled or not. Set N mod M = 0 for better performance. If N mod M is not 0, yet users want the efficiency to drop as little as possible, derive an algorithm for height calculation.

First of all, users need to know the buffer size for each format.

```
if (UYVY 2-plane or 3-plane)
    y_max_buf_size = 256x48
    uv_max_buf_size = 128x48
else if (YUV420)
    y_max_buf_size = 256x64
    uv_max_buf_size = 128x32
else
    y_max_buf_size = 256x32
    uv_max_buf_size = 256x32
```

Algorithm: (Width is tile width.)

Phase 1:

Users can apply the algorithm on Y channel first to get the approximation first.

```
Coeff 1 = floor (MAX_BUF_SIZE / WIDTH/2) *2
Coeff 2 = Ceiling (WIDTH / Coeff1)
Buf_line_num = Ceiling (WIDTH/Coeff 2 / 4) *4 (To make sure if Buf_line_num mod 4 = 0)
```

```
If (buf_line_num > width)
Buf_line_num = ceiling (width / 4) * 4 (To make sure width >= Buf_line_num)
```

```
else if (Buf_line_num * Buf_line_num * Coeff 2 > MAX_BUF_SIZE) (buffer overflow)
Buf_line_num = buf_height.
```

Phase 2:

Check if the setting is over the buffer size or not.

```
Y_buf_check = 0
Uv_buf_check = 0
```

```
//internal buffer check
while ((y_buf_check !=0) | (uv_buf_check!=0)){

    // Y buffer check
    Internal_y_buf_width = Ceiling(width/buf_line_num) x buf_line_num //multiple of buf_line_num and >=
width
}
```

```
Internal_y_buf_usage = Internal_y_buf_width x buf_line_num
If (internal_y_buf_usage > y_max_buf_size){
    buf_line_num = buf_line_num -4
    Y_buf_check =0
    Uv_buf_check =0
}else{
    Y_buf_check =1
}

// UV buffer check
if (YUV422 rotate 0/180)
    Uv_blk_width = main_blk_width/2
    Uv_blk_line = main_buf_line_num
Else if ((YUV422 rotate 90/270)
    Uv_blk_width = main_blk_width
    Uv_blk_line = main_buf_line_num/2
Else if (YUV420)
    Uv_blk_width = main_blk_width/2
    Uv_blk_line = main_buf_line_num/2
Else
    Uv_blk_width = main_blk_width
    Uv_blk_line = main_buf_line_num

Internal_uv_buf_width = Ceiling(Uv_blk_width / Uv_blk_line) x Uv_blk_line

Internal_uv_buf_usage = Uv_blk_width x Uv_blk_line
If (internal_uv_buf_usage > uv_max_buf_size){
    Main_buf_line_num = maian_buf_line_num -4
    Y_buf_check =0
    Uv_buf_check =0
}else{
    uv_buf_check =1
}
}
```

6.15.5.2 SMI 256 Byte Boundary Restriction

Methods to Handle Boundary Restriction

SMI has a restriction that any burst issued from DMA cannot cross the 256-byte address boundary, whether in read or write action. The reason is the address mapping mechanism inside DRAM controller, which changes the DRAM Bank address at each 256-byte boundary.

The probability of crossing the 256-byte boundary by random access is analyzed here. As illustrated in Figure 6-79. The probability P is

$$P = (\text{byte per burst length})/256$$

It depends on the designed burst length of MDP_WROT. For example, if the minimum burst length is 16 with 128-bit data bus, the probability $P = 64/256 = \frac{1}{4}$, which means that 25% of burst requests will be separated. Though the total data amount is fixed, it increases 25% burst request times and will reduce DRAM access utility.

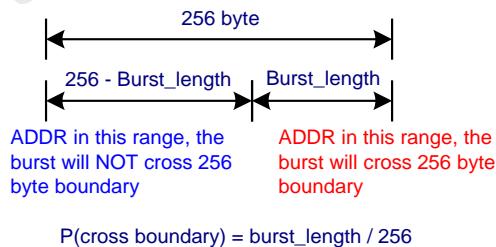


Figure 6-79 Probability of Crossing 256-byte Boundary by Random Access

To handle this restriction from SMI, there are two common solutions. One is to adopt a burst separator between MDP_WROT and the SMI port. As illustrated in Figure 6-80, the original DMA requests are represented in black arrows. Because the 256-byte boundary restriction is not considered, some requests should be able to cross the 256-byte boundary. The burst separator filters all requests from MDP_WROT, which allows the legal request to pass, as represented in blue arrows, but separate the illegal request into two requests, as represented in yellow arrows. Although it is simple to implement, it increases the total request number.

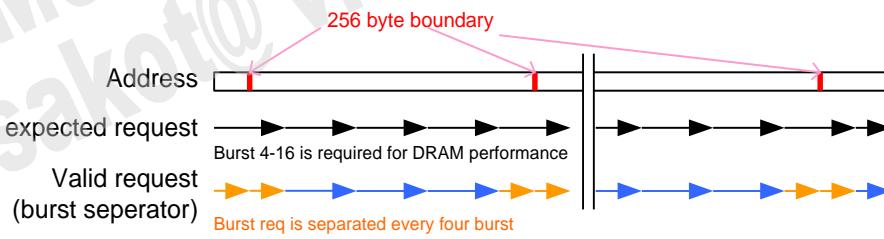


Figure 6-80 Scan-line Request without Considering 256-byte Boundary

The second method is to handle the 256-byte boundary restriction inside the control logics of MDP_WROT. After the data are stored in burst accumulator, MDP_WROT will adjust the burst length to meet the boundary when the target address is close to the 256-byte boundary. In common cases, the minimum burst length is a factor of 256 bytes; therefore, the requests after the adjusted request will automatically align with the 256-byte boundary. This method is difficult to implement but can efficiently reduce the overhead of DMA requests (see Figure 6-81).

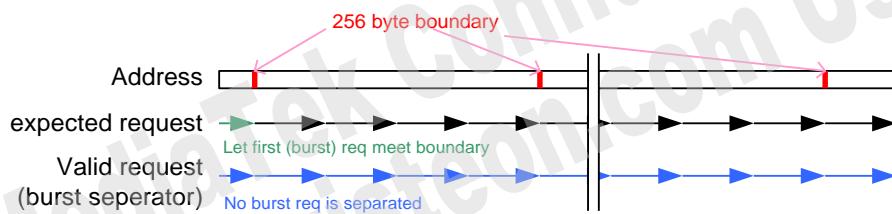


Figure 6-81 Scan-line Request Considering 256-byte Boundary

SMI Boundary Restriction In Case of Rotation

The method described in the previous section is based on continuous scan-line access of DMA. It works only in 0° or 180° rotation with or without flip. And it does not work in 90° or 270° rotation with or without flip.

Considering a general condition with 90° or 270° rotation, if the stride is not a multiple of 256 bytes, the distribution of 256-byte boundary will differ in every line as described in Figure 6-82. To handle this, the SMI_IF controller issues requests with different burst lengths according to the current address.

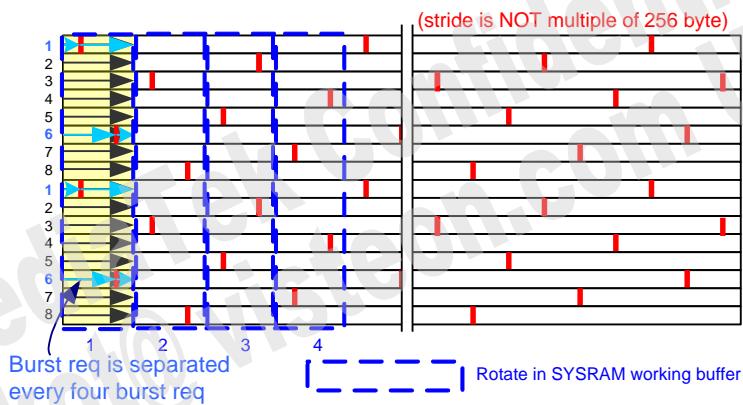


Figure 6-82 4-16 Burst in 270° Rotation with/without Flip

6.16 Display Read Direct Memory Access

6.16.1 Introduction

The Display Read Direct Memory Access (DISP_RDMA) engine is responsible for providing data to the interface engines, e.g. Digital Serial Interface (DSI) and Digital Parallel Interface (DPI). Because the interface engines need the real time service, the RDMA engine contains one line buffer to store the sufficient pixel data. It also detects the usage of data buffer to trigger the deep sleep mode of External Memory Interface (EMI).

6.16.2 Features

DISP_RDMA has the following features:

- Direct link input mode
- Memory input mode
 - Input format: YUV422, UYVY422, YVYU422, UYVY422, RGB565, RGB888, ARGB8888
 - Input footprint: Raster-scan mode, 64-byte-aligned tile mode
 - Slowdown mode
- Output control
 - Byte swap, RGB swap
 - Progressive mode, interlace mode
 - Programmable YUV to RGB matrix
 - Non-stop output mode if the data buffer is under-running
- Buffer control
 - 512 × 16 byte data buffer (2730 pixels with RGB888 format)
 - Programmable request/pre-ultra/ultra control mechanism

6.16.3 Block Diagram

Figure 6-83 shows the detailed block diagram of the RDMA engine. The clocks are automatically configured and one asynchronous First-In First-Out (FIFO) is used for the output clock domain.

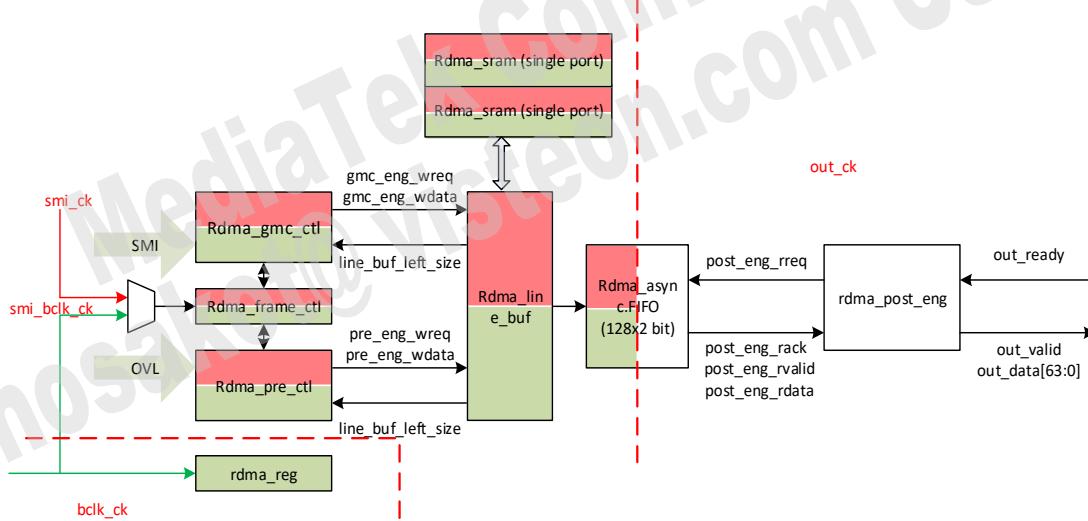


Figure 6-83 Block Diagram of RDMA Engine

6.16.4 Register Definition

For register details, please refer to Chapter 4.10.6 of “MT2712 IVI Application Processor Registers”.

6.16.5 Programming Guide

Figure 6-84 shows the general programming sequence.

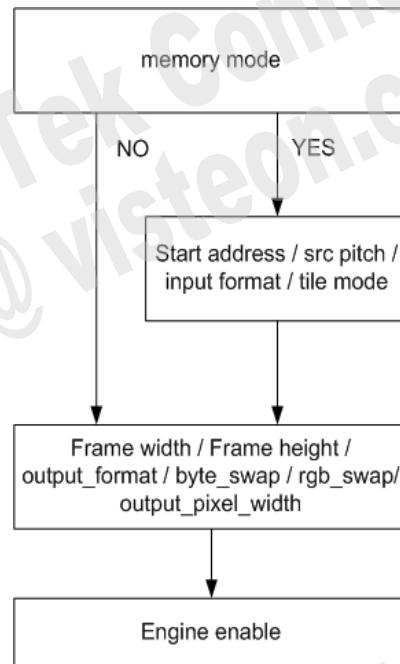


Figure 6-84 General Programming Sequence

The direct link mode can be easily configured by output's frame width and height. On the other hand, the memory mode has several modes, which will be discussed in the following sub-sections.

6.16.5.1 Memory Mode Control: Raster Scan Input, Progressive Output

1. Set 'MEM_MODE_START_ADDR' = Address of the first pixel.
2. Set 'MEM_MODE_SRC_PITCH' = Width of the source frame.

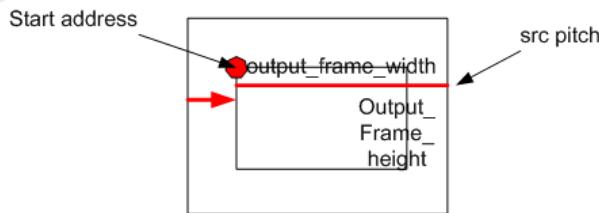


Figure 6-85 Basic Memory Mode Configuration

6.16.5.2 Memory Mode Control: Raster Scan Input, Interlace Output

1. Set 'MEM_MODE_START_ADDR' = Address of the first pixel.
2. Set 'MEM_MODE_SRC_PITCH' = *Double* of the width of the source frame.

6.16.5.3 YUV to RGB Transfer Formula

$$\begin{pmatrix} Y_{out} \\ U_{out} \\ V_{out} \end{pmatrix} = \begin{pmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{pmatrix} \times \begin{pmatrix} Y_{input} + pre_add_0 \\ U_{input} + pre_add_1 \\ V_{input} + pre_add_2 \end{pmatrix} + \begin{pmatrix} post_add_0 \\ post_add_1 \\ post_add_2 \end{pmatrix}$$

Figure 6-86 Programmable Color Matrix

6.16.5.4 Byte Swap/RGB Swap

Set up the corresponding registers according to the data format on the memory.

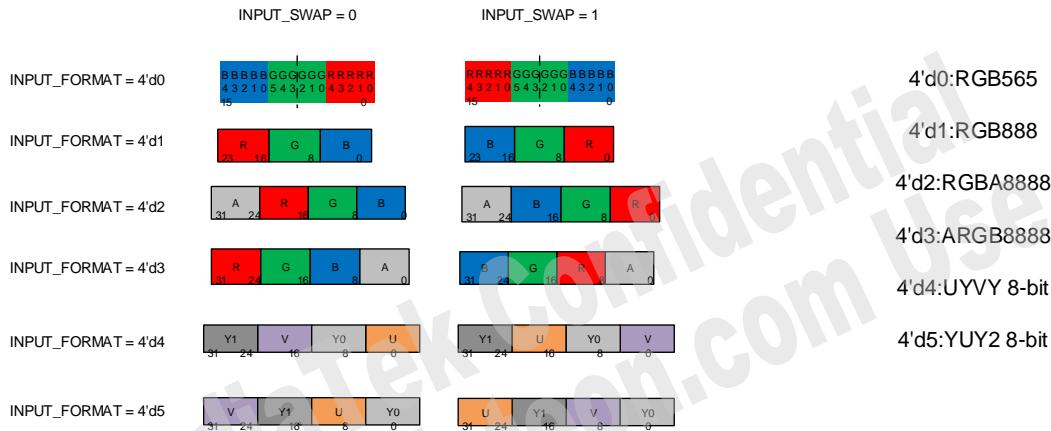


Figure 6-87 INPUT_FORMAT/SWAP

6.17 Display Write Direct Memory Access

6.17.1 Introduction

Multimedia Data Path WDMA (MDP_WDMA) and Display WDMA (DISP_WDMA) have the same hardware architecture, but slightly different SRAM sizes, which influences the line-width support and outstanding ability. Write Direct Memory Access (WDMA) performs the job of DMA writing out the data in display/mdp pipeline into DRAM. In the following sections, WDMA is called DISP_WDMA.

6.17.2 Features

DISP_WDMA supports the following features:

- Dither
- Programmable parameter color transform
- Input color format YUV444/RGB888

- Output format RGB565/RGB888/ARGB8888/UYVY/YV12/NV12/NV21
- 3-tap filter in horizontal and 2-tap filter in vertical for YUV420 downsampling
- Byte swap/color swap/UV swap

6.17.3 Block Diagram

DISP_WDMA has a 256×128 two-port SRAM for DMA FIFO. One 320×64 single-port SRAM is for vertical filtering line_buffer (2,560 pixels). For line-width larger than 2560, YUV420 vertical downsampling must be with the drop-pixel scheme.

MDP_WDMA has a 128×128 two-port SRAM for DMA FIFO. One 64×64 single-port SRAM is for vertical filtering line_buffer (512 pixels). For line-width larger than 512, YUV420 vertical downsampling must be with the drop-pixel scheme.

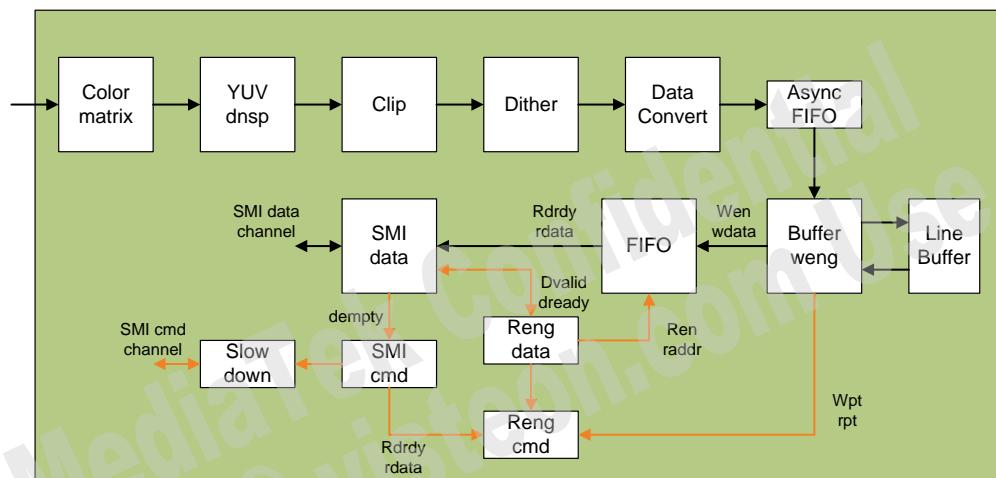


Figure 6-88 DISP_WDMA Block Diagram

6.17.4 Register Definition

For register details, please refer to Chapter 4.10.7 of “MT2712 IVI Application Processor Registers”.

6.17.5 Programming Guide

The fundamental key parts are base address/input output format/data strides in memory. If color transform is needed, color transform matrix will also be needed. Other settings such as dither/filter are optional.

MUST program

WDMA_CFG
WDMA_SRC_SIZE
WDMA_CLIP_SIZE
WDMA_CLIP_COORD
WDMA_DST_ADDR0

WDMA_DST_W_IN_BYTE
WDMA_DST_ADDR1
WDMA_DST_ADDR2
WDMA_DST_UV_PITCH
WDMA_DST_ADDR_OFFSET0
WDMA_DST_ADDR_OFFSET1
WDMA_DST_ADDR_OFFSET2

The read result of WDMA_CT_DBG means the input counter of the WDMA.

The Most Significant Bit (MSB) part is line counter and the Least Significant Bit (LSB) part is pixel counter.

WDMA_CT_DBG can be used as debugging register.

The read result of WDMA_FLOW_CTRL_DBG means the current state of WDMA.

The state machine of the WDMA is as the following:

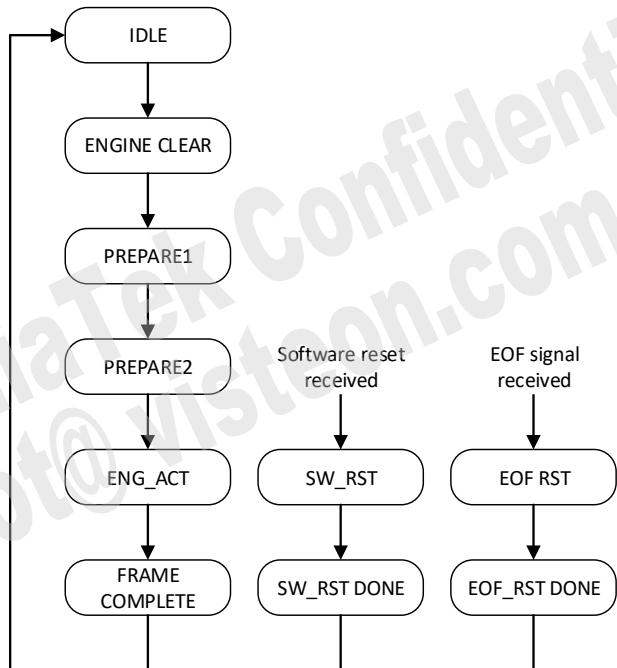


Figure 6-89 Typical DISP_WDMA Programming Sequence

6.18 Display Overlay

6.18.1 Introduction

Display overlay (DISP_OVL) can do up to four alpha blending layers. The four source layers should only come from the memory. Four RDMA (Read DMAs) and four sets of color transformation are included.

6.18.2 Features

Table 6-16 DISP_OVL Features

Item	Main Function	Description
1	Resolution	Supports 4096*2160
2	4-layer overlay	Supports four layers of blending
3	Color format uniform	Supports color format uniform and swap control (RGB565/RGB888/RGBA8888/ARGB8888/YUYV/YUV2)
4	3D display	Interleave left and right image for 3D display (landscape and portrait mode)
5	Color conversion	Support fixed matrix coefficient color conversion
6	Color key	Source color key
7	Alpha blending	Supports pixel alpha blending
8	Flexible ROI (Region of Interest) system	Supports individual color depth, window size, vertical and horizontal offset.
9	Flip function	Vertical/horizontal/180-degree flip

6.18.3 Block Diagram

There are four overlay RDMA in DISP_OVL; each RDMA contains a 128×128 single port SRAM (ping-pong buffer).

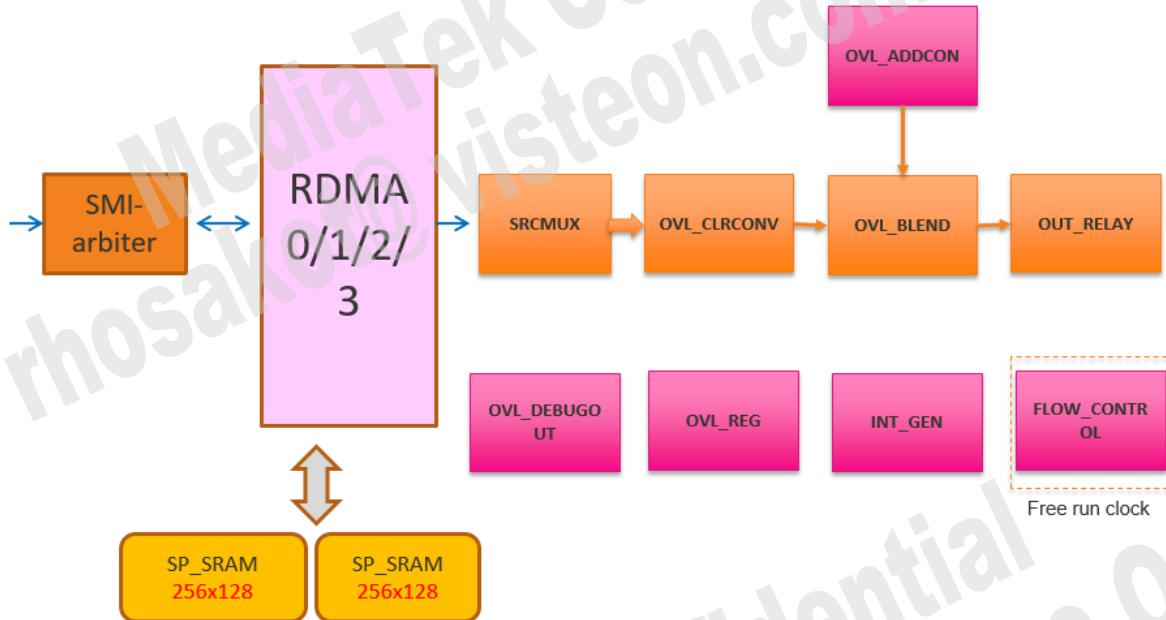


Figure 6-90 DISP_OVL Block Diagram

6.18.4 Register Definition

For register details, please refer to Chapter 4.10.8 of “MT2712 IVI Application Processor Registers”.

6.18.5 Programming Guide

Register settings:

1. Configure DATAPATH.
 - a. OVL_SRC_CON
 - b. OVL_L*_CON
 - c. OVL_DATAPATH_CON
2. Configure RDMA's parameters.
 - a. OVL_L*_ADDR
 - b. OVL_L*_PITCH
 - c. OVL_RDMA*_CTRL
3. Configure layer's parameters.
 - a. OVL_L*_SRC_SIZE
 - b. OVL_L*_OFFSET
 - c. OVL_ROI_SIZE

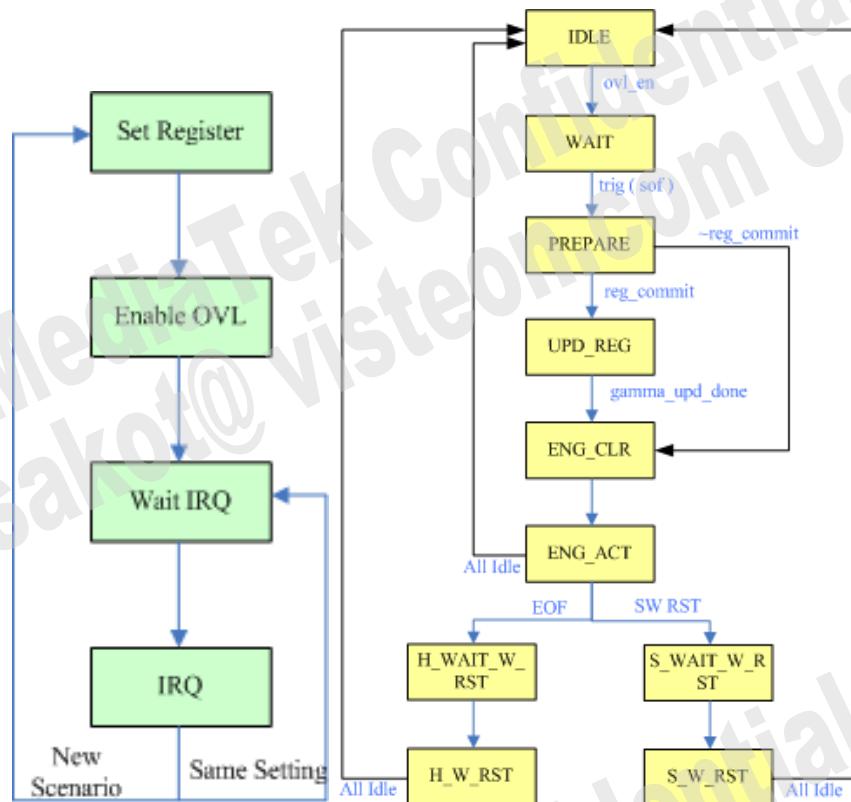


Figure 6-91 Typical DISP_OVL Programming Procedure

6.19 Display Color Engine (DISP_COLOR)

6.19.1 Introduction

The color management engine is highly configurable and programmable; it fits different display panels and satisfies different users' preferences for various purposes.

The color management engine is designed for two applications, namely, getting better picture quality and having one panel that resembles the other in their output characteristics.

6.19.2 Features

The color processor is very flexible for luma/saturation/hue adjustments. Main features are:

- Flexible architecture: Picture Quality (PQ) processing at many possible stages
- Input/output color space conversion
- Hue engine:
 - Partial hue: Modifies hue angle of specific hue phase
- Y engine:
 - Adaptive luma: Adaptively adjusts Y curve according to image content
 - Global contrast/brightness adjustment
 - Chroma boost: Compensates saturation value due to Y change
- Sat engine:
 - Partial S: Modifies saturation value of specific hue phase
 - Global saturation adjustment
- Histogram statistics: Includes Y histogram and chroma histogram

6.19.3 Block Diagram

The color engine provides various luma/saturation/hue adjustments. The block diagram is shown as Figure 6-92.

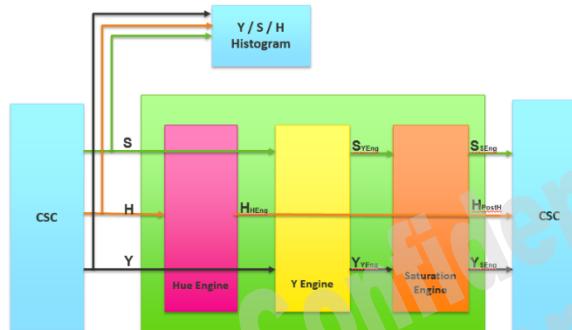


Figure 6-92 Color Processor

6.19.4 Register Definition

For register details, please refer to Chapter 4.10.9 of “MT2712 IVI Application Processor Registers”.

6.19.5 Programming Guide

6.19.5.1 Hue Adjustment

There are 20 hue phases for partial hue adjustment.

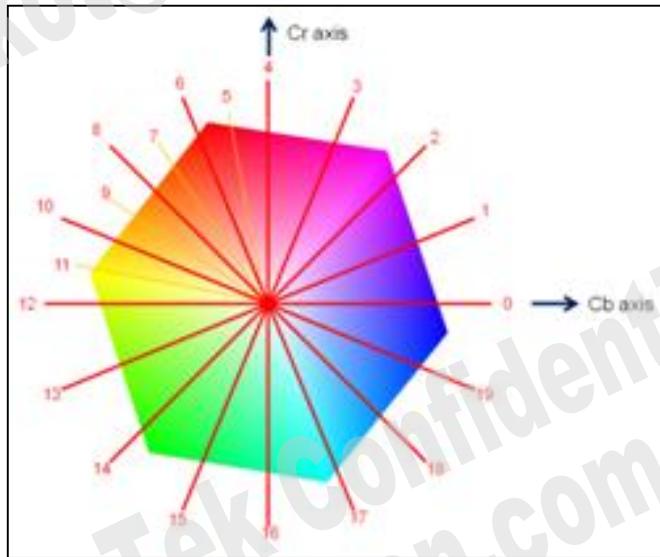


Figure 6-93 Hue Phase Distribution

Partial hue adjustment is achieved by HUE_TO_HUE_W_* registers. An example of partial hue adjustment is shown in Figure 6-94.

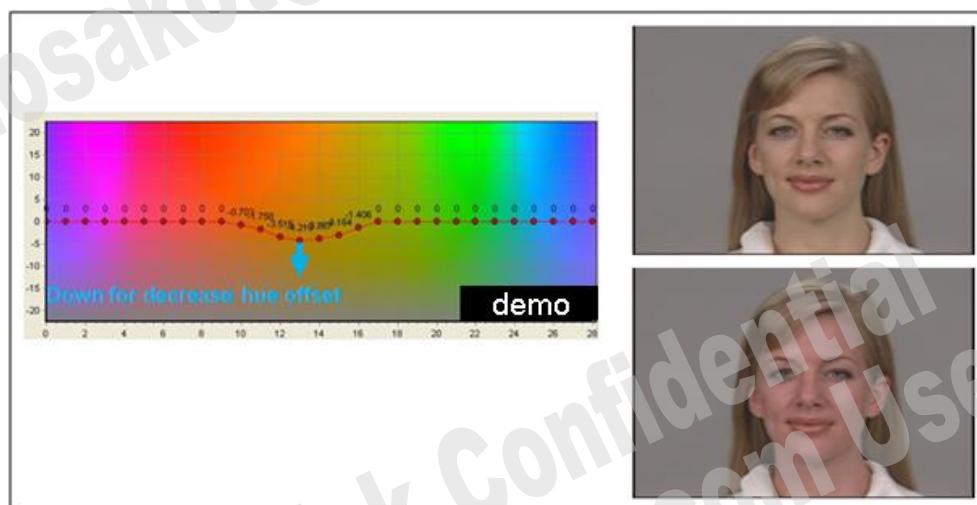


Figure 6-94 Partial Hue Adjustment Example

6.19.5.2 Luma Adjustment

Global contrast and brightness adjustment are also provided for intuitive luma adjustment. The registers are G_CONTRAST and G_BRIGHTNESS.

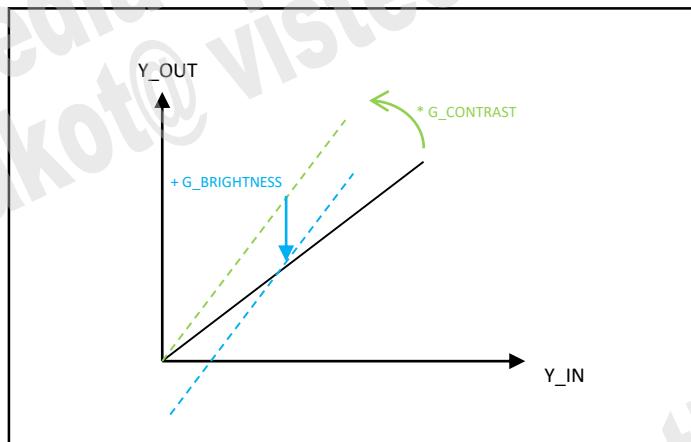


Figure 6-95 Contrast and Brightness Adjustment Example

6.19.5.3 Saturation Adjustment

In human vision system, luminance increasing results in pale images, even though the saturation is not decreased. Therefore, the color processor provides a “Chroma Boost” mechanism to detect the luminance increasing amount and decides the corresponding saturation boost level. The boost adjustment example is illustrated in Figure 6-96.

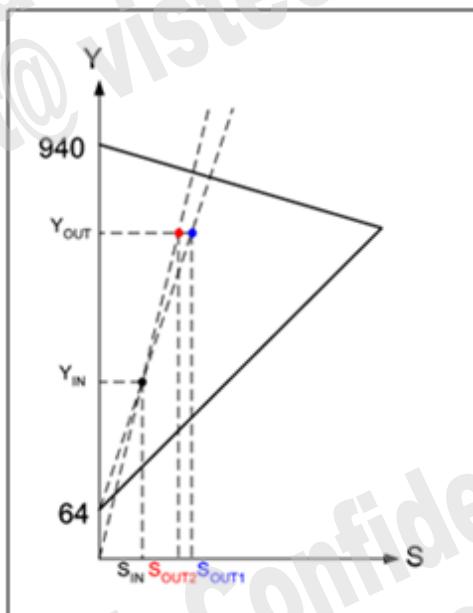


Figure 6-96 Chroma Boost Adjustment Example

For saturation adjustment, the color processor provides a flexible partial saturation adjustment. Similar to Partial Hue, the hue plan is equally divided into 20 hue phases and saturation corresponding to each hue phase can be customized by three gains and two turning points. The adjustments for each hue phase are controlled by PARTIAL_SAT_GAIN*, PARTIAL_SAT_POINT*. Figure 6-97 shows the saturation adjustment in one hue phase.

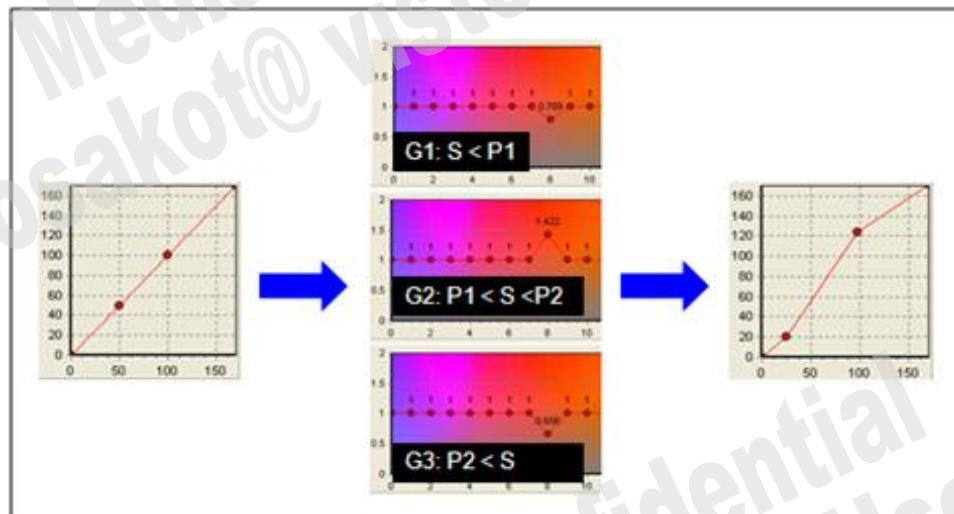


Figure 6-97 Example of Partial Saturation for One Hue Phase

A global saturation gain is provided for intuitive saturation adjustment. The control register is G_SATURATION (see Figure 6-98).

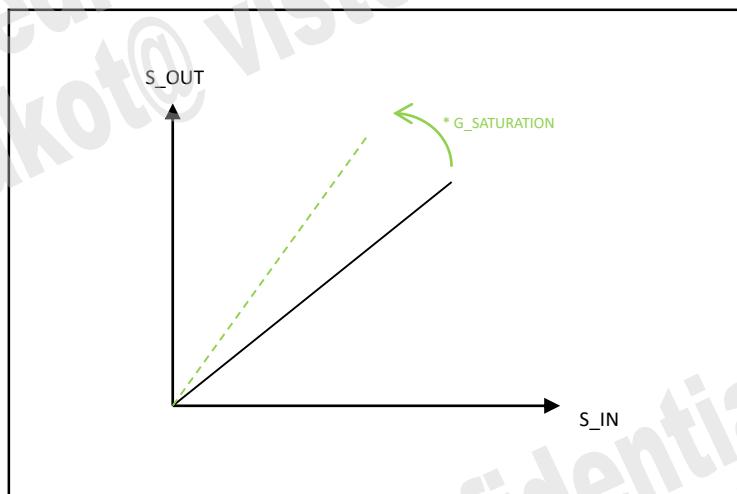


Figure 6-98 Illustration of Global Saturation Adjustment

6.20 Display Adaptive Ambient Light Controller

6.20.1 Introduction

The Display Adaptive Ambient Light (DISP_AAL) engine is composed of three sub-functions: color correction, gamma correction and AAL processor. The color correction and gamma correction change the overall mixture of RGB colors to fit the characteristics of target panel. The AAL processor is composed of content adaptive and ambient light adaptive luminance controller, and it is responsible for backlight power saving and sunlight visibility improvement.

6.20.2 Features

- Color correction
 - Fixed-coefficient inverse gamma table with wide-gamut support
 - Programmable 3×3 matrix
 - Fixed-coefficient gamma table
- Gamma correction
 - 10-bit gamma table with 512 entries
 - Non-block gamma Look-up Table (LUT) programming
- AAL processor
 - 33-bin weighted histogram
 - Dark Region Enhancement (DRE) for sunlight visibility
 - Content Adaptive Backlight Controller (CABC) compensation for backlight power saving

6.20.3 Block Diagram

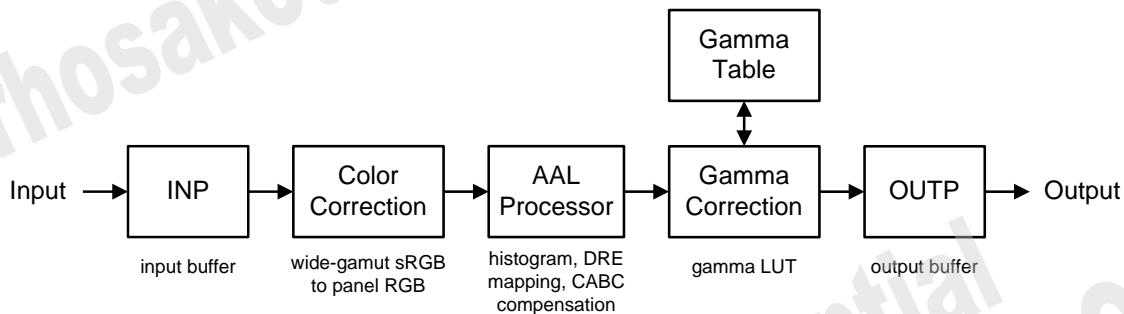


Figure 6-99 AAL Engine Block Diagram

Figure 6-100 is the block diagram of color correction core.

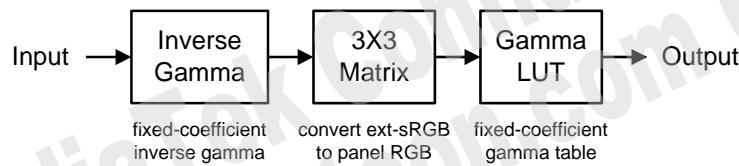


Figure 6-100 Color Correction Block Diagram

Figure 6-101 is the block diagram of AAL processor.

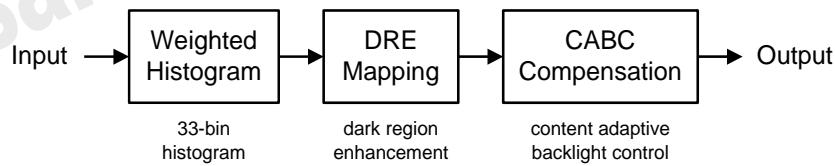


Figure 6-101 AAL Processor Block Diagram

6.20.4 Color Correction

In order to display accurate image colors, the Liquid Crystal Display (LCD) panel needs to match the standard sRGB color gamut. However, most LCD panels only display 65 to 80 percent of the sRGB color gamut. The OLED panels, on the other hand, can display over 130 percent of sRGB color gamut. With color correction, users can reproduce correct color on panels with different color gamuts.

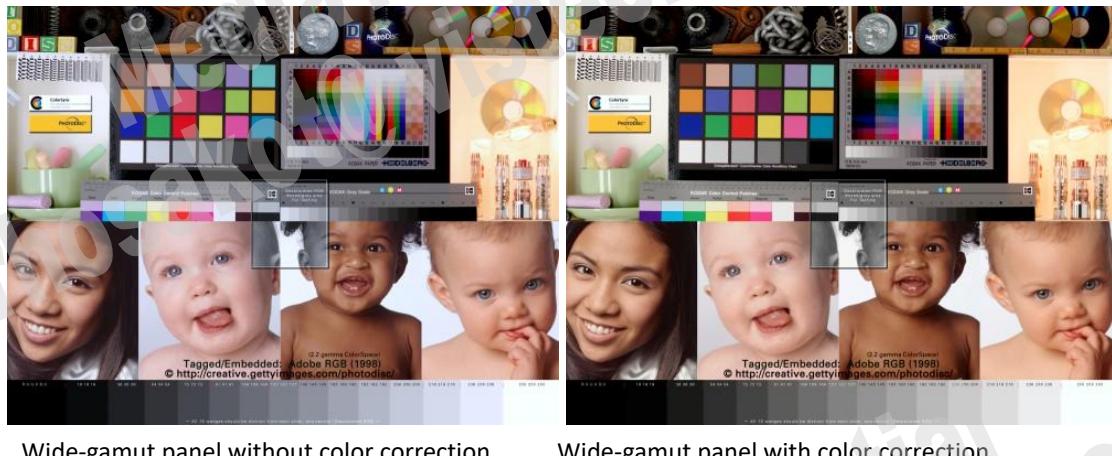


Figure 6-102 Visual Effect of Color Correction

6.20.5 Gamma Correction

For accurate image reproduction, the transfer function of the panel should have a standard 2.20 gamma value. The gamma correction consists of three programmable look-up tables for RGB colors. It applies arbitrary mapping curve to compensate the incorrect transfer function of the panel.

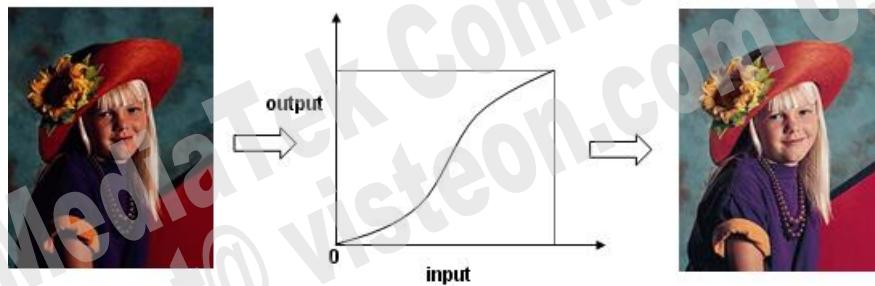


Figure 6-103 Visual Effect of Gamma Correction

6.20.6 DRE

DRE improves the visibility under sunlight.



Figure 6-104 Visual Effect of DRE

6.20.7 Register Definition

For register details, please refer to Chapter 4.10.10 of “MT2712 IVI Application Processor Registers”.

6.20.8 Programming Guide

1. Get MMSYS mutex

MediaTek Proprietary

and Confidential

© 2021 MediaTek Inc. All rights reserved.

Unauthorized reproduction or disclosure of this document, in
whole or in part, is strictly prohibited.

Page 500 of 576

2. Set AAL_EN = 1
3. Release MMSYS mutex

6.21 Display GAMMA Processing Engine

6.21.1 Introduction

The Display GAMMA (DISP_GAMMA) processing engine is composed of three sub-functions: color correction, gamma correction and dithering. The color correction and gamma correction change the overall mixture of RGB colors to fit the characteristics of target panel. Dithering, the last stage of image post processing, randomizes quantization errors.

6.21.2 Features

The processing engine has the following features:

- Color correction
 - Fixed-coefficient inverse gamma table with wide-gamut support
 - Programmable 3×3 matrix
 - Fixed-coefficient gamma table
- Gamma correction
 - 10-bit gamma table with 512 entries
 - Non-block gamma Look-up Table (LUT) programming
- Dithering
 - Ordered dithering
 - Error diffusion (a.k.a. error dispersion) dithering
 - Linear Feedback Shift Register (LFSR) dithering
 - Rounding

6.21.3 Block Diagram

Figure 6-105 illustrates the GAMMA engine block diagram.

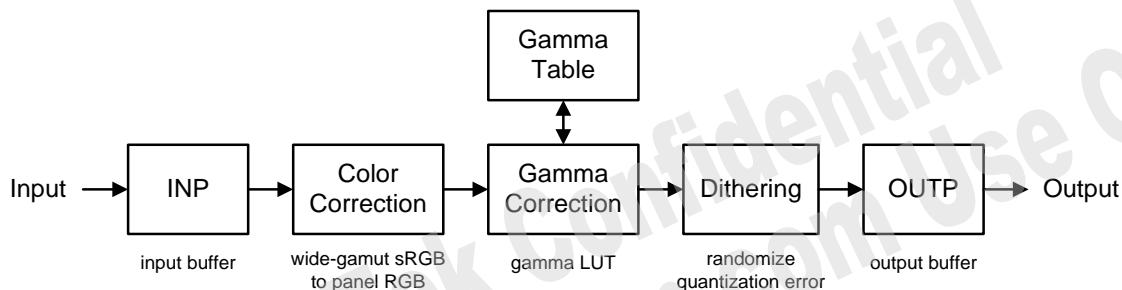


Figure 6-105 GAMMA Engine Block Diagram

Figure 6-106 shows the diagram of the color correction core.

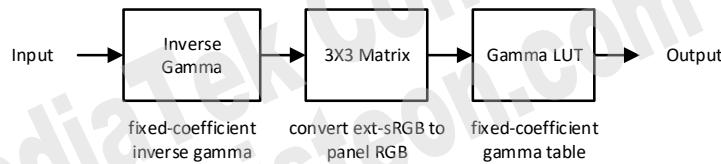


Figure 6-106 Color Correction Block Diagram

6.21.4 Color Correction

In order to display accurate image colors, the panel must match the standard sRGB color gamut. However, most LCD panels only display 65 to 80 percent of the sRGB color gamut. The OLED panels, on the other hand, display more than 130 percent of sRGB color gamut. With color correction, users can reproduce correct color on panels with different color gamuts.

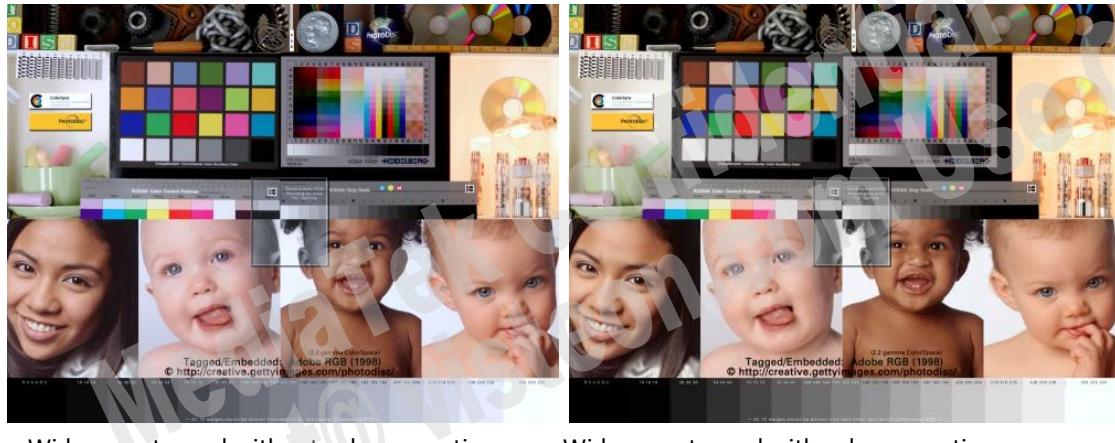


Figure 6-107 Visual Effect of Color Correction

6.21.5 Gamma Correction

For accurate image reproduction, the transfer function of the panel should have a standard 2.20 gamma value. The gamma correction consists of three programmable look-up tables for RGB colors. It applies arbitrary mapping curve to compensate for the incorrect transfer function of the panel.

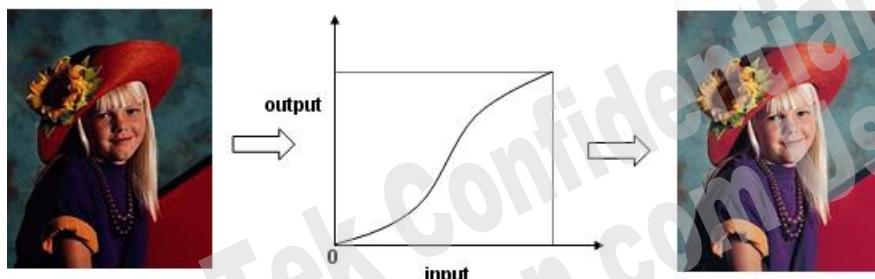


Figure 6-108 Visual Effect of Gamma Correction

6.21.6 Register Definition

For register details, please refer to Chapter 4.10.11 of “MT2712 IVI Application Processor Registers”.

6.21.7 Programming Guide

There are three possible usage scenarios for different situations, as shown in Figure 6-109.

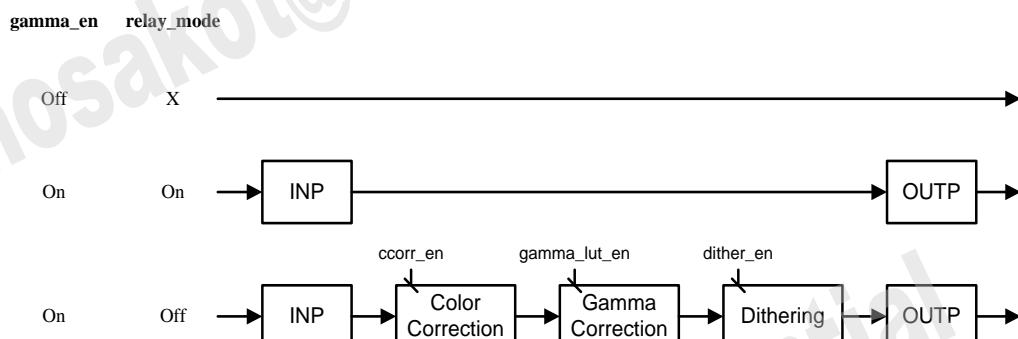


Figure 6-109 Usage Scenarios

6.22 UFOe Processor

6.22.1 Introduction

The UFOe engine in MT2712 is used for L/R mode in dual-DSI.

6.22.2 Features

- L/R mode

6.22.3 Block Diagram

The block diagram of UFOe engine is shown in Figure 6-110.

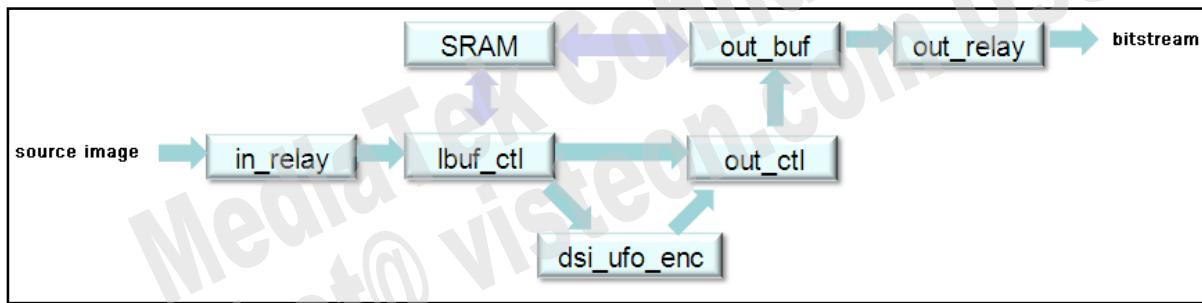


Figure 6-110 UFOe Engine Block Diagram

6.22.4 Register Definition

For register details, please refer to Chapter 4.10.12 of “MT2712 IVI Application Processor Registers”.

6.22.5 Programming Guide

Table 6-17 UFO Modes

Function	reg_lr_mode	reg_lr_overlap	reg_cfg_0b[2:1]
L/R overlap mode	1	Overlap number	X

Table 6-18 UFO Width/Height Reference in Each Mode

Function	Internal Width/Height
L/R overlap mode	$W = \text{input_width}$ $H = \text{input_height}$

Function	Additional Settings	Output Width/Height	Note
L/R overlap mode	$\text{reg_lr_overlap} = A$	$\text{Output width} = W + 2*A$ $\text{Output height} = H$	-

6.23 Display OD Engine

6.23.1 Introduction

The OD (OverDrive) engine in MT2712 only supports dithering.

6.23.2 Features

The Display OD (DISP_OD) engine supports the following features:

- Dithering

6.23.3 Block Diagram

The block diagram of DISP_OD, shown in Figure 6-111, includes DISP_OD wrapper, OD and dither.

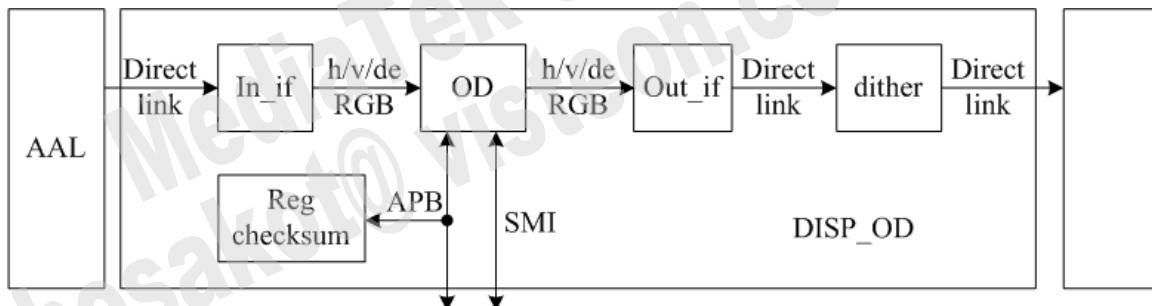


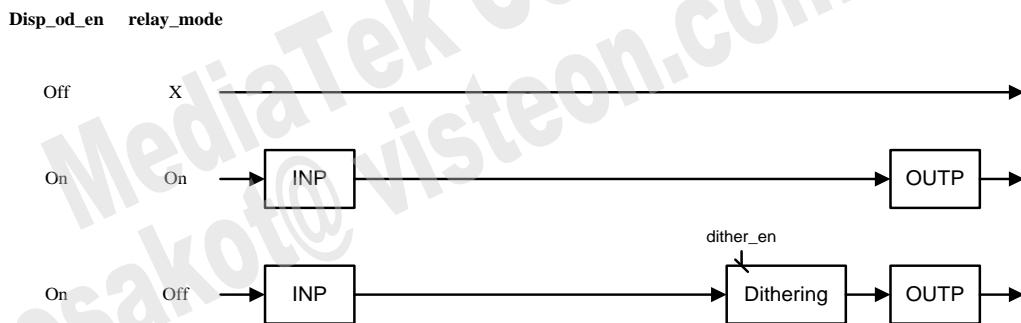
Figure 6-111 DISP_OD Engine

6.23.4 Register Definition

For register details, please refer to Chapter 4.10.13 of “MT2712 IVI Application Processor Registers”.

6.23.5 Programming Guide

There are three possible usage scenarios for DISP_OD.



6.24 Digital Parallel Interface

6.24.1 Introduction

The Digital Parallel Interface (DPI) controller provides data for the LVDS module.

6.24.2 Features

The DPI controller supports the following features:

- Programmable 2D/3D, progressive/interlaced timing generator
- Fixed-coefficient color space transform

- RGB 8-bit/YUV444 8-bit/YUV422 8-bit, 10-bit, 12-bit output data format
- YC MUX (CCIR656-like) output format
- Secure display
- 3-tap chroma Low-Pass Filter (LPF)
- Internal pattern generator

6.24.3 DPI Block Diagram

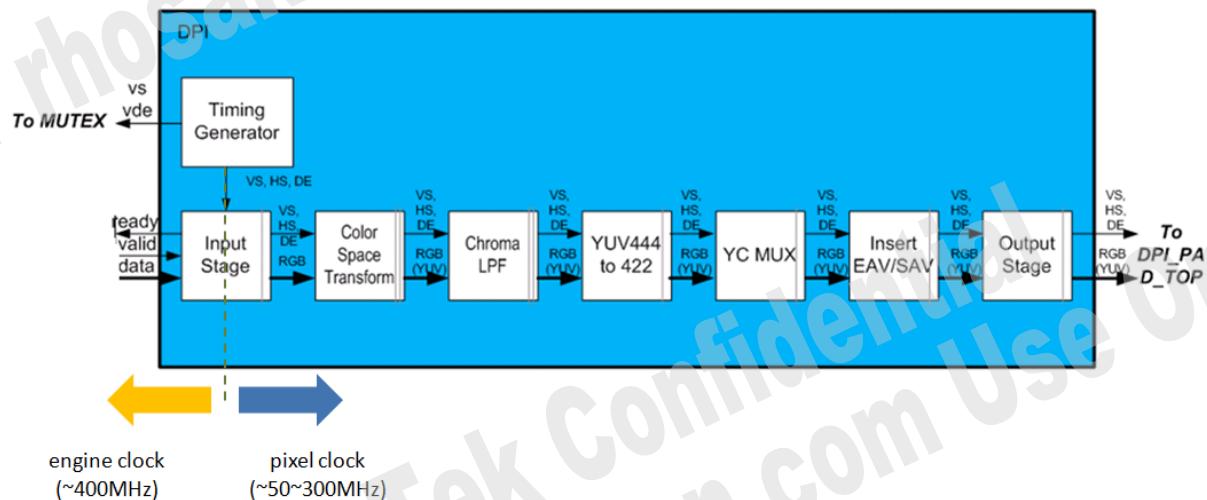


Figure 6-112 DPI Engine Block Diagram

6.24.4 Register Definition

For register details, please refer to Chapter 4.10.14 of “MT2712 IVI Application Processor Registers”.

6.24.5 Programming Guide

Figure 6-113 shows the DPI programming flow diagram. First, configure each timing register based on the target frame timing. Then, reset and enable DPI.

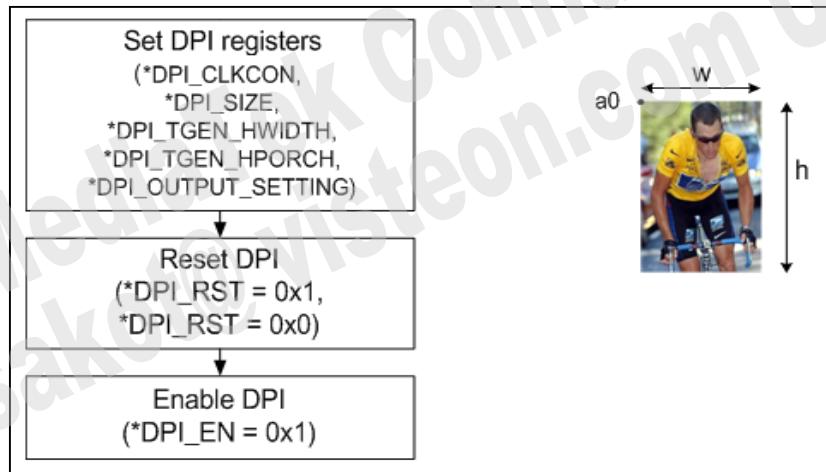


Figure 6-113 Programming Flow Diagram

6.25 Display Serial Interface

6.25.1 Introduction

Display Serial Interface (DSI) is based on the MIPI Alliance Specification, supporting high-speed serial data transfer between host processor and peripheral devices such as display modules. DSI supports both video mode and command mode data transfer defined in MIPI specification, and it also provides bidirectional transmission in low-power mode to receive messages from the peripheral. DSI should work with MIPI_TX_Config module to obtain its engine clock to analog Digital Physical (DPHY) macro, and it should work with DMA engines in the previous stage of Display (DISP) path to read frame pixels from memory.

6.25.2 Features

The DSI engine has the following features for display serial interface:

- one clock lane and up to four data lanes
- Throughput up to 1.0 Gbps for one data lane
- Bidirectional data transmission in low-power mode in data lane 0
- Unidirectional data transmission in high-speed mode in data lane 0~3
- 128-entry command queue for command transmission
- Support three types of video modes: sync-event, sync-pulse, burst mode
- Pixel format of RGB565/RGB666/loosely RGB666/RGB888
- Support non-continuous high-speed transmission in both clock/data lanes
- Support command mode frame transmission
- Support peripheral Tearing Effect (TE) and external TE signal detection
- Support limited high-speed residual packet transmission during video mode's blanking period
- Support ultra-low power mode control

- Dual DSI mode allows transmission with two clock lanes and eight data lanes

6.25.3 Block Diagram

Figure 6-114 illustrates the dual DSI modules in Multimedia System (MMSYS) and their relationship.

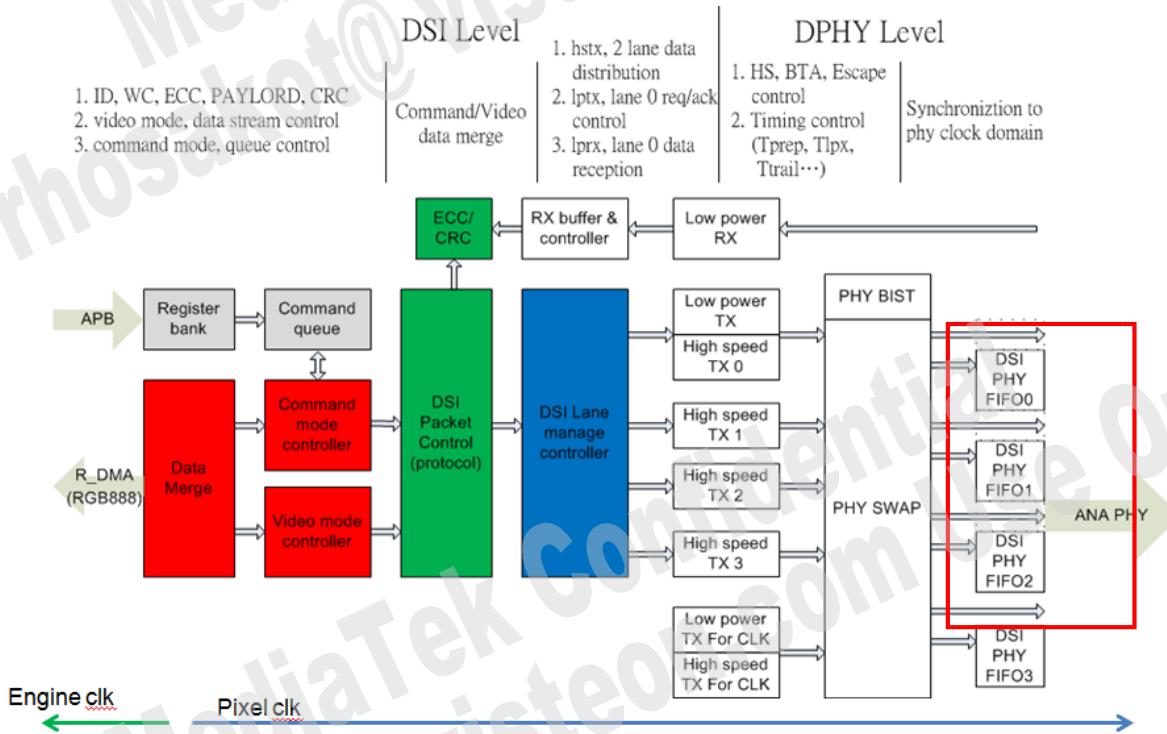


Figure 6-114 Block Diagram of DSI Modules in MMSYS

The frame pixels read by OVL / DISP_RDMA are sent to DSI. The DSIO_SEL and DS1_SEL modules are to select the input source of frame pixels, which allow DSI modules to obtain data from the same source frame through dispatch of DISP_SPLIT module. The dual DSI modules can operate simultaneously to perform an 8-lane transmission with either even-odd or left-right frame data partition.

6.25.4 DSI AC Timing

6.25.4.1 DSI Low-Power Driver Characteristics

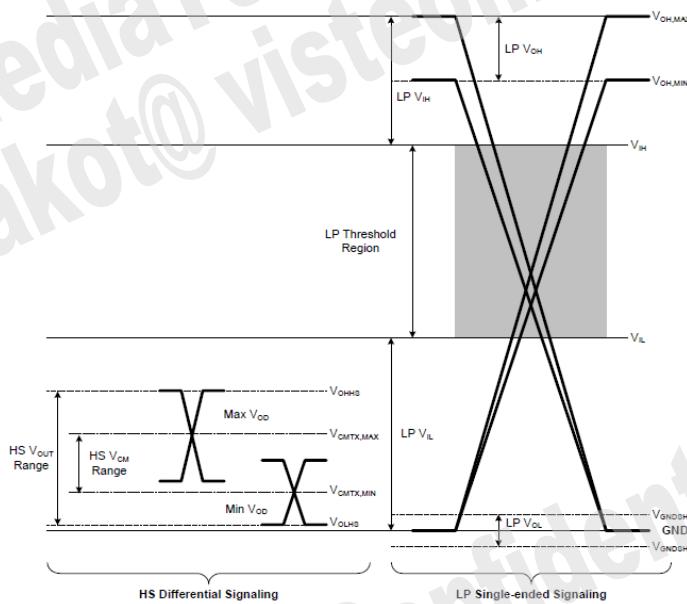


Figure 6-115 D-PHY Signaling Levels

Table 6-19 DSI LP Driver Characteristics

Symbol	Description	Low Power		Unit.
		Min.	Max.	
$V_{O_{LP}}$	Thevenin output high level	1.1	1.3	V
$V_{O_{LP}}$	Thevenin output low level	-50	50	mV
$Z_{O_{LP}}$	Output impedance of LP transmitter	110	-	Ohm
$T_{R_{LP}}/T_{F_{LP}}$	15%-85% rise time and fall time	-	25	ns
$SR@C_L=0pF$	Slew rate @ CLOAD = 0pF	-	500	mV/ns
$SR@C_L=5pF$	Slew rate @ CLOAD = 5pF	-	300	mV/ns
$SR@C_L=20pF$	Slew rate @ CLOAD = 20pF	-	250	mV/ns
$SR@C_L=70pF$	Slew rate @ CLOAD = 70pF	-	150	mV/ns
$SR(\text{rising edge}, 700mV\sim930mV)$	Slew rate @ CLOAD = 0 to 70pF (Rising Edge Only) when the output voltage is between 700 mV and 930 mV.	$30-0.075*(V-700)$	-	mV/ns
$SR(\text{rising edge}, 400mV\sim700mV)$	Slew rate @ CLOAD = 0 to 70pF (Rising Edge Only) when the output voltage is between 400 mV and 700 mV.	30	-	mV/ns
$SR(\text{falling edge}, 400mV\sim930mV)$	Slew rate @ CLOAD = 0 to 70pF (Falling Edge Only)	30	-	mV/ns

Symbol	Description	Low Power		Unit.
		Min.	Max.	
C _{LOAD}	Load capacitance	0	70	pF

6.25.4.2 DSI High-Speed Driver Characteristics – DPHY

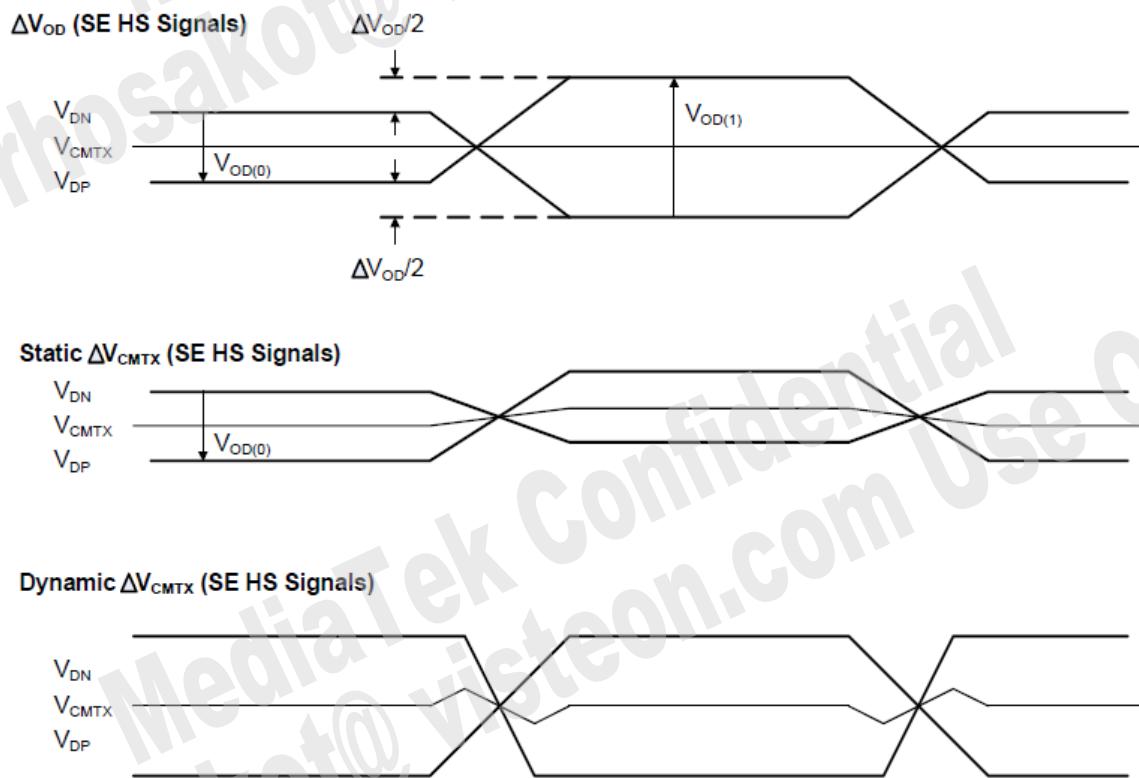


Figure 6-116 Possible ΔV_{CMTX} and ΔV_{OD} Distortions of the Single-ended HS Signals

Table 6-20 DSI High-Speed Driver Characteristics DPHY

Symbol	Description	Full Speed		Unit.
		Min.	Max.	
Data rate		80	1500	Mbps
Driver Parameters				
V _{OHHHS}	HS output high voltage	-	360	mV
V _{OD}	HS transmit differential voltage	140	270	mV
Δ V _{OD}	VOD mismatch when output is Differential-1 or Differential-0	-	14	mV
V _{CMTX}	HS transmit static common mode voltage	150	250	mV
ΔV _{CMTX(1,0)}	VCMTX mismatch when output is differential-1 or Differential-0	-	5	mV

Symbol	Description	Full Speed		Unit.
		Min.	Max.	
Z_{os}	Single ended output impedance	40	62.5	Ohm
ΔZ_{os}	Single ended output impedance mismatch	-	10	%
$\Delta V_{CMTX(HF)}$	Common-level variations above 450MHz	-	15	mV _{rms}
$\Delta V_{CMTX(LF)}$	Common-level variation between 50-450MHz	-	25	mV _{peak}

6.25.4.3 CSI High-Speed Receiver Characteristics – DPHY

6.25.4.3.1 High-Speed Receiver Timing

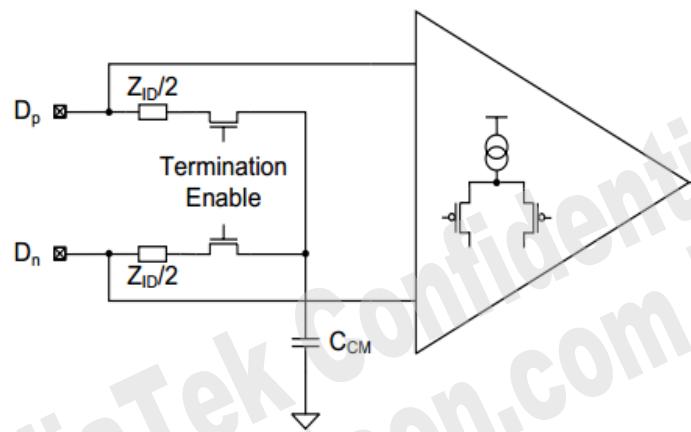


Figure 6-117 HS Receiver Implementation Example

6.25.4.3.2 Low-Power Receiver Timing

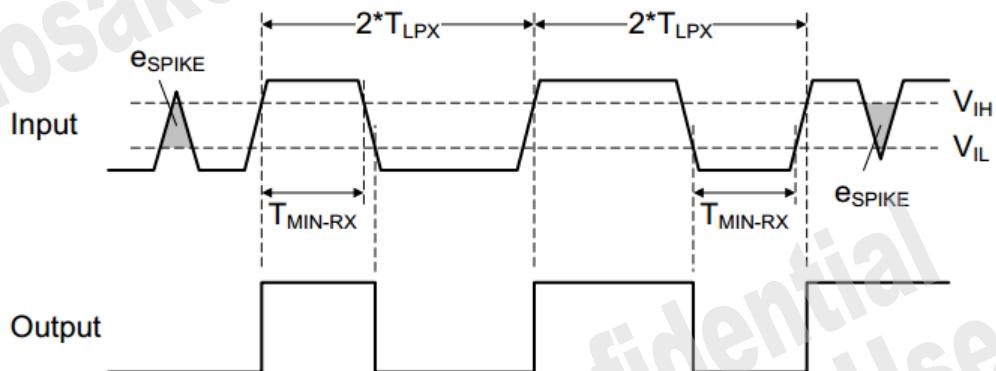


Figure 6-118 Input Glitch Rejection of Low-Power Receivers

6.25.4.3.3 High-Speed Clock Timing

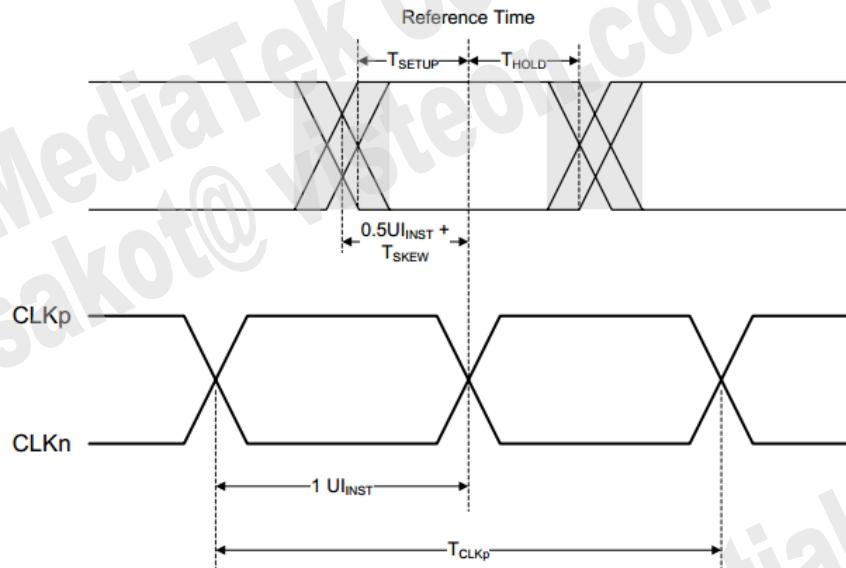


Figure 6-119 Data to Clock Timing Definitions

Table 6-21 Full Speed Electrical and Timing Information

Symbol	Description	Full Speed		Unit
		Min.	Max.	
V _{IDTH}	Differential input high threshold	-	70	mV
V _{IDTL}	Differential input low threshold	-70	-	mV
V _{IHHS}	Single-ended input high voltage	-	460	mV
V _{ILHS}	Single-ended input low voltage	-40	-	mV
V _{CMRXDC}	Common-mode voltage HS receive mode	70	330	mV
Z _D	Differential input impedance	80	125	Ohm
T _{SETUP[RX]}	RX Data to Clock Setup Time Tolerance	0.15(DATA RATE <= 1G) 0.2(1G < DATA RATE <= 1.5G)	-	UI
T _{HOLD[RX]}	RX Data to Clock Hold Time Tolerance	0.15(DATA RATE <= 1G) 0.2(1G < DATA RATE <= 1.5G)	-	UI
ΔV _{CMRX(HF)}	Common-mode interference beyond 450 MHz	-	100	mV/pp
ΔV _{CMRX(LF)}	Common-mode interference 50 MHz – 450 MHz	-50	50	mVpeak
C _{CM}	Common-mode termination	-	60	pF

6.25.4.4 CSI Low-Power Receiver Characteristics – DPHY

Table 6-22 CSI Low-Power Receiver Characteristics DPHY

Symbol	Description	Low Power		Unit
		Min.	Max.	
V _{IL}	Logic 0 input voltage, not in ULP State	-	550	mV
V _{IH}	Logic 1 input voltage	880	-	mV
V _{HYST}	Input hysteresis	25	-	mV
V _{IHCD}	Logic 1 contention threshold	450	-	mV
V _{ILCD}	Logic 0 contention threshold	-	200	mV
e _{spike}	Input pulse rejection	-	300	V.ps
T _{MIN-RX}	Minimum pulse width response	20	-	ns
V _{INT}	Peak interference amplitude	-	200	mV
f _{INT}	Interference frequency	450	-	MHz

6.25.5 Register Definition

For register details, please refer to Chapter 4.9.15 of “MT2712 IVI Application Processor Registers”.

6.25.6 Programming Guide

6.25.6.1 Clock Control

After enabling PLL clock from MIPI DPHY macro and Clock Gating (CG) cells in MMSYS, a primary module clock enable should be set to turn-on for the entire design (see Table 6-23). To disable clock for DSI, unset it. For more details on MIPI DPHY PLL clock control, refer to the functional specification of MIPI_TX_Config module.

Table 6-23 Sequence to Enable Module Clock

Step	Description	R/W	Register [bit]	Value
1	Enable primary module clock	W	DSI_COM_CON [1]	1

6.25.6.2 DSI HS Clock Control

After enabling the module clock and properly setting up related registers, it is necessary to enable the clock lane before starting high-speed data transmission by setting up the DSI_PHY_LCCON register. Please follow the steps shown in Table 6-24 to turn on the high-speed clock.

Table 6-24 Sequence to Enable High-Speed Clock

Step	Description	R/W	Register [bit]	Value
1	Enable DSI high-speed clock	W	DSI_PHY_LCCON [0]	1

To disable the high-speed clock, follow the steps in Table 6-25. Note that the high-speed clock should be turned off before entering ultra-low power mode.

Table 6-25 Sequence to Exit Ultra-low Power Mode on Clock Lane

Step	Description	R/W	Register [bit]	Value
1	Disable DSI high-speed clock	W	DSI_PHY_LCCON [0]	0

6.25.6.3 DSI ULPS Enter Control

In certain conditions, the DISP subsystem may be powered off after DSI enters the Ultra-low Power State (ULPS) to reduce more power consumption. ULPS is a protocol defined in MIPI specification and forces Liquid Crystal Monitor (LCM) to enter the standby scenario to save power. The sleep-in control in this module is shown in Table 6-26.

Table 6-26 Sequence of Sleep-in Control (Entering Ultra-low Power Mode)

Step	Description	R/W	Register [bit]	Value
1	Disable DSI high-speed clock	W	DSI_PHY_LCCON [0]	0
2	Data lane enter ultra-low power mode	W	DSI_PHY_LD0CON [1]	1
3	Clock lane enter ultra-low power mode	W	DSI_PHY_LCCON [1]	1

6.25.6.4 DSI ULPS Exit Control

To wake up DSI and LCM from ULPM, executing an “ULPS-exit” procedure defined by MIPI specification is required. This procedure performs a protocol to get DP/DN signals from LP-00 through LP-10 to LP-11, where the LP-10 period should not be less than 1 ms. The procedure can be easily controlled by the DSI sleep-out sequence with an interrupt indication in Table 6-27.

Table 6-27 Sequence of Sleep-out Control (Exiting Ultra-low Power Mode)

Step	Description	R/W	Register [bit]	Value
1	Enable sleep-out interrupt	W	DSI_INTEN [6]	1
2	Configure corresponding cycle counts for 1ms LP-10 period	W	DSI_PHY_TIMECON4	0x22E09
3	Recover lane number (4-lane)	W	DSI_TXRX_CON [5:2]	0xF
4	Sleep-out start	W	DSI_START [2]	1
5	Issue interrupt and sleep-out done	R	DSI_INTSTA [6]	1
6	Clear interrupt	W	DSI_INTSTA [6]	1

6.25.6.5 DPHY Timing Control

All of the timing parameters defined in MIPI DPHY should be properly written in the DSI registers for correct timing control. The written value is based on the DSI internal clock cycle period which is related to DPHY PLL clock settings through the MIPI_TX_Config module.

For example, the timing parameter $T_{HS-PREPARE}$ must be between $40\text{ ns} + 4*\text{UI}$ and $85\text{ ns} + 6*\text{UI}$, where the Unit Interval (UI) means time interval, equal to the duration of any HS state on clock lane. If the clock lane is set to

500 MHz frequency, and the bit rate is set to 1 Gbps, the UI should be 1 ns. In other words, the value of $T_{HS-PREP}$ must be between 44 ~ 91 ns. The internal DSI clock is 4x divided by clock lane, or 125 MHz. To satisfy $T_{HS-PREP}$, the register value DA_HS_PREP should be 6 ~ 11 (see Table 6-28).

Table 6-28 DPHY Timing Parameter Register Settings

	Timing Specification	Absolute Time for UI: 1 ns	DA_HS_PREP Value
$T_{HS-PREP}$	40 ns + 4*UI~85 ns + 6*UI	44~91 ns	6~11

Note that for different bit rate requirements, the UI values are various. For more precise timing control, select DPHY clock that is as fast as possible. However, the faster DPHY clock is set, the more the power waste. A suitable clock is beneficial to the optimization of system power consumption.

The registers of timing parameters for data lane and clock lane are illustrated in Figure 6-120 and Figure 6-121 respectively. The registers for Bus Turnaround (BTA) timing are illustrated in Figure 6-122.

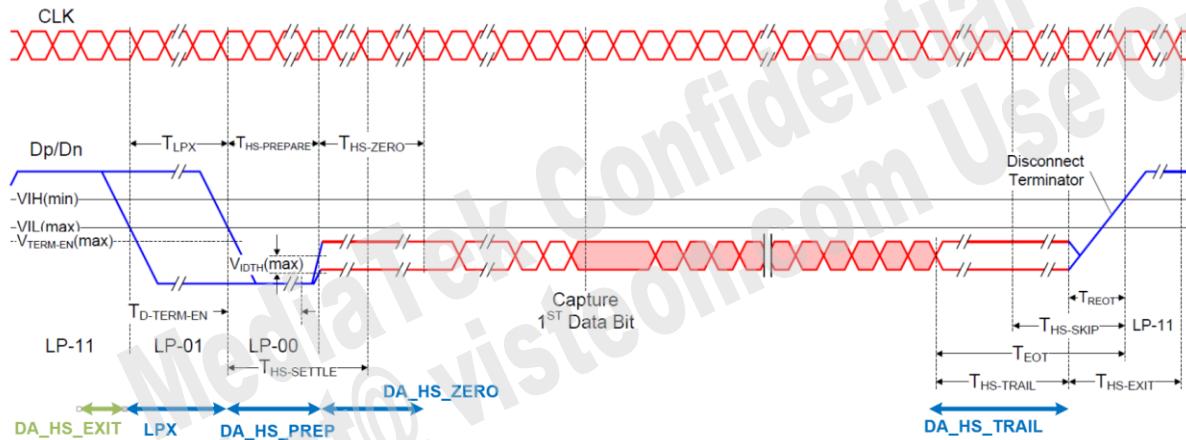


Figure 6-120 Registers for Data Lane Timing Parameters

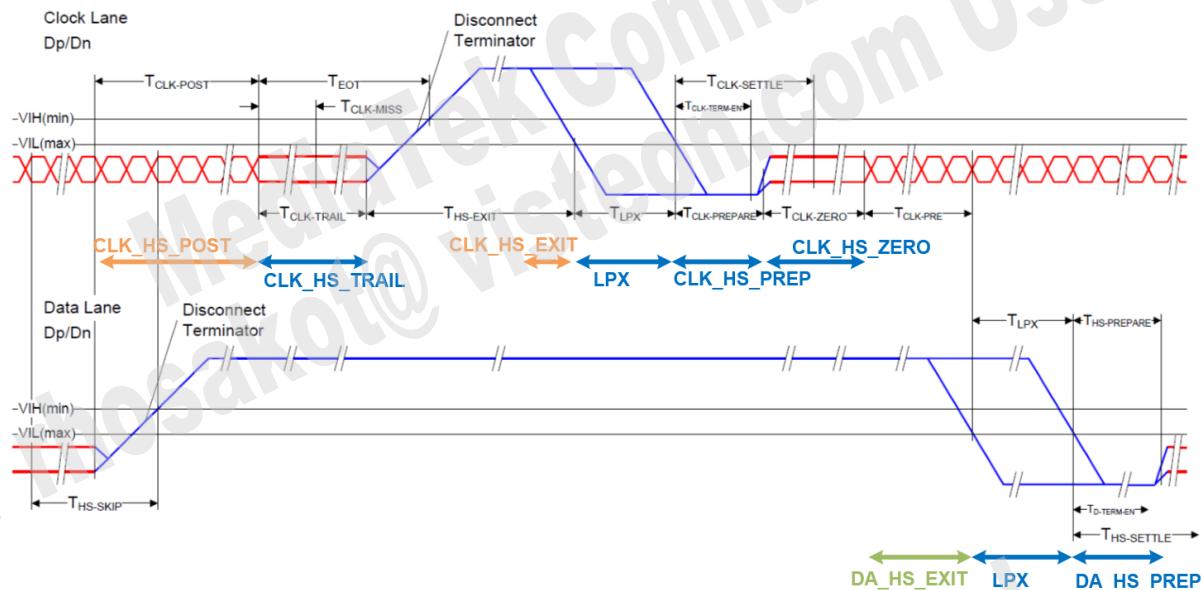


Figure 6-121 Registers for Clock Lane Timing Parameters

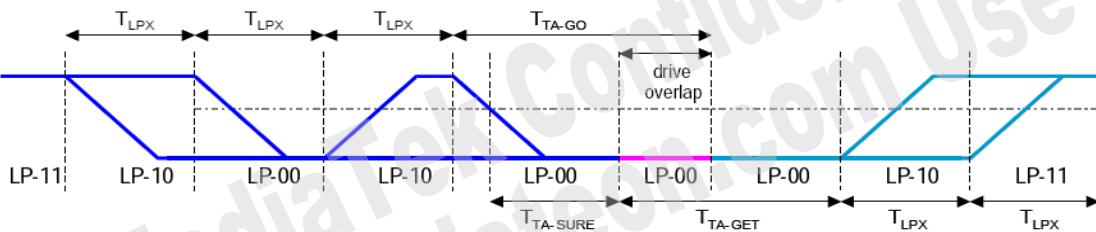


Figure 6-122 Register for BTA Timing Parameters

6.25.6.6 Command Mode

DSI supports command mode transmission by writing commands to a dedicated command queue. By configuring commands and triggering DSI, the transmission can be executed sequentially.

6.25.7 Command Queue

DSI has a dedicated command queue that is 32-bit wide and up to 128-entry deep, as shown in Figure 6-123. To simplify the settings for transmitting a packet in the command mode, the command queue is designed to categorize all possible transmission types and commands into four primary instructions and to unify all DSI specification commands into one or several 32-bit wide instructions.

Figure 6-123 also illustrates a 32-bit instruction structure with the instruction format of CONFIG byte.

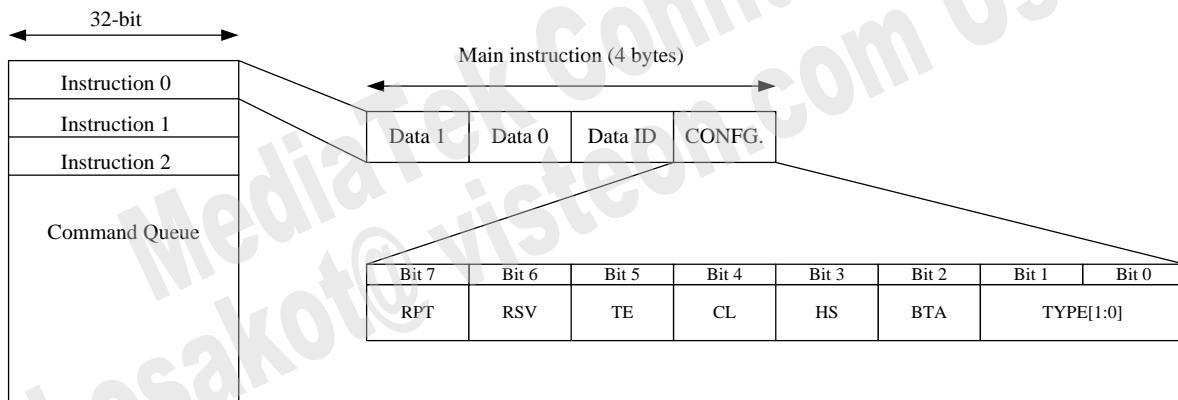


Figure 6-123 DS1 Command Queue Instruction Format

Table 6-29 shows the descriptions of the CONFIG byte to an instruction. For convenience's sake, virtual channel 0 is used for all packets in the following examples. Software programmers should take charge of the real virtual channel numbers to the slave devices.

Table 6-29 Configuration Field Description of Main Instruction

	Value	Function	Description
Type[1:0]	00	-	Used for DSI short packet read/write command
	01	-	Used for DSI frame buffer write command (long packet)
	10	-	Used for DSI generic long packet write command
	11	-	Used for DSI frame buffer read command (short packet)
BTA	0	Off	Turns around the DSI link after the DSI command is transmitted
	1	On	
HS	0	Off	Enables HS TX transmission for this packet; otherwise transmit packet via LP TX
	1	On	
CL	0	8-bit	Selects command length for frame buffer read/write instruction. Only effective for type 1 and type 3 instructions.
	1	16-bit	
TE	0	Off	Enables TE request. Will only turn around the DSI link without any packet transmission
	1	On	
Resv	-	-	Reserved for further use
Resv	-	-	Reserved for further use

6.25.7.1 Type-0 Instruction

Type-0 instruction is used to transmit short packets. Table 6-30 lists the formats of type-0 instruction where (Data ID + Data 0 + Data 1) is constructed by a DSI short packet command (without Error Correction Code (ECC)).

Table 6-30 Type-0 Instruction Format

Byte 3	Byte 2	Byte 1	Byte 0
Data 1	Data 0	Data ID	CONFIG.

Suppose users are to send “Turn On Peripheral” and “Color Mode On” commands which are transmitted via Low-Power Transmitter (LPTX) and High-Speed Transmitter (HSTX) respectively and request slave’s response after the second command is finished, the descriptions can be translated into two 32-bit instructions and achieved by the steps illustrated in Table 6-31.

Table 6-31 Type-0 TX Example

Step	Description	R/W	Register [bit]	Value
1	Fill command queue entry-0	W	DSI_CMDQ_0	0x0000120C
2	Fill command queue entry-1	W	DSI_CMDQ_1	0x00003200
3	Set command count	W	DSI_CMDQ_CON	0x2
4	Start command	W	DSI_START [0]	0x1
5	Issue interrupt and receive slave response	R	DSI_INTSTA [0]	0x1
6	Clear interrupt status	W	DSI_INTSTA [0]	0x1
7	Read trigger status (Acknowledge)	R	DSI_RX_TRIG_STA [3:0]	0x4
8	Respond with read acknowledgement to module and go to next commands in queue	W	DSI_RX_RACK [0]	0x1
9	Issue interrupt and command done	R	DSI_INTSTA [1]	0x1
10	Clear interrupt status	W	DSI_INTSTA [1]	0x1

6.25.7.2 Type-1 Instruction

Type-1 command is used to write data into the frame buffer. As shown in Table 6-32, there are 4 bytes constructing this type of instruction where mem_start_0 and mem_start_1 can be generic commands defined by slave vendors or Display Command Set (DCS) commands. mem_start_1 is optional, e.g. the memory start/continue command can be single-byte as DCS defines. It depends on the CL bit of the CONFIG. Byte to indicate whether the DSI controller sends mem_start_1 or not. Since the length of frame buffer to be updated is not a constant, this type of instruction may send several long packets to the slave. The payload data and length of each packet (excluding mem_start_0 and mem_start_1) are prepared by the RDMA controller which couples the output of image data path or layer overlay result to the DSI controller. For the first packet, mem_start_0 and mem_start_1 (if CL = 1) are used as the parameters to inform the slave that the host is starting to write the frame buffer. For the remaining packets, the register value MEM_CONTI[15:0] will be used as the parameters to inform the slave to write these data following the last pixel of the previous packet. For more flexibility, mem_start_0, mem_start_1, MEM_CONTI[15:0] and CL are all programmable. However, it consumes only one entry of command queue.

Users need to set up two registers to define the packet length and packet count for a frame-based type-1 transmission. Frame width in bytes should be set to DSI_PSCON, and frame height in lines should be set to DSI_VACT_NL, respectively. These two registers are used in both video and command mode frame data transmission.

Table 6-32 Type-1 Instruction Format

Byte 3	Byte 2	Byte 1	Byte 0
Mem start 1 (optional)	Mem start 0	Data ID	CONFIG

Please follow the example in Table 6-33 to write the frame buffer via DCS command in the HS TX mode.

Table 6-33 Type-1 TX Example

Step	Description	R/W	Register [bit]	Value
1	Fill command queue entry-0	W	DSI_CMDQ_0	0x002C3909
2	Set command count	W	DSI_CMDQ_CON	0x1
3	Set memory continue command	W	DSI_MEM_CONTI [15:0]	0x3C
4	Start command	W	DSI_START [0]	0x1
5	Issue interrupt and command done	R	DSI_INTSTA [1]	0x1
6	Clear interrupt status	W	DSI_INTSTA [1]	0x1

6.25.7.3 Type-2 Instruction

Type-2 instruction is used to send a long packet. As shown in Table 6-34, this type of main instruction requires several sub-instructions which do not have CONFIG. To send a type-2 command, users need to write a CONFIG with TYPE = 2 and packet header information (Data ID + 2 byte word count) to entry 0, and write a series of data bytes in size of word count to the following entries, excluding ECC and checksum. The bytes in following entries will be treated as long packet data instead of the next instruction, until the word count size is reached. The command queue count should be set as the number of multiple entries used.

The type-2 command is sent in LPTX mode due to memory latency in reading sub-instruction data. Besides, type-2 command should be sent individually without the next instruction followed. For the 32-entry command queue, the maximum word count for long packet is 124 bytes.

Table 6-34 Type-2 Instruction Format

Byte 3	Byte 2	Byte 1	Byte 0
WC 1	WC 0	Data ID	CONFIG.
Data 3	Data 2	Data 1	Data 0
-	-	Data WC-1	Data WC-2

Table 6-35 provides the example of sending three parameters (0x33, 0x22, 0x11) by a generic long packet command with 3-byte word count.

Table 6-35 Type-2 TX Example

Step	Description	R/W	Register [bit]	Value
1	Fill command queue entry-0 (data ID and word count)	W	DSI_CMDQ_0	0x00032902
2	Fill command queue entry-1 (parameters)	W	DSI_CMDQ_1	0x00112233
3	Set command count	W	DSI_CMDQ_CON	0x2

Step	Description	R/W	Register [bit]	Value
4	Start command	W	DSI_START [0]	0x1
5	Issue interrupt and command done	R	DSI_INTSTA [1]	0x1
6	Clear interrupt status	W	DSI_INTSTA [1]	0x1

Note that the RPT bit of CONFIG is designed for this type of instruction. It is useful for the NULL packet or blanking packet. For example, if a null packet is to be sent, only the main instruction (entry-0 of command queue) will be needed, and the following payload data will be sent as “0”.

6.25.7.4 Type-3 Instruction

Type-3 instruction is used for reading frame buffer. As shown in Table 6-36, the format is the same as that of type-1. When this instruction is executed, the host will first send a short packet with memory start parameter given in byte 2 and byte 3 and automatically issue the next packet by memory continue parameters programmed in MEM_CONTI[15:0]. The number of total packets required to be sent depends on the FRM_BC and “maximum return packet size”. For example, if 1,024 bytes are to be read from the frame buffer in the slave, and the “maximum return packet size” is set to “4”, there will be another 255 short packets with memory continue parameters to be sent successively, after the first short packet described in main instruction is sent.

Table 6-36 Type-3 Instruction Format

Byte 3	Byte 2	Byte 1	Byte 0
Mem start 1 (optional)	Mem start 0	Data ID	CONFIG.

See Table 6-37 for the example of using type-3 instruction to perform the frame buffer read.

Table 6-37 Type-3 TX Example

Step	Description	R/W	Register [bit]	Value
1	Fill command queue entry-0	W	DSI_CMDQ_0	0x002E0603
2	Set command count	W	DSI_CMDQ_CON	0x1
3	Set memory read continue command	W	DSI_MEM_CONTI[15:0]	0x3E
4	Start command	W	DSI_START [0]	0x1
5	Issue interrupt and receive slave response	R	DSI_INTSTA [0]	0x1
6	Read receive data bytes	R	DSI_RX_DATA03	-
7	Clear interrupt status	W	DSI_INTSTA [0]	0x1
8	Respond with read acknowledgement to module and go to next commands in queue	W	DSI_RX_RACK [0]	0x1
9	Issue interrupt and command done	R	DSI_INTSTA [1]	0x1
10	Clear interrupt status	W	DSI_INTSTA [1]	0x1

6.25.7.5 Command Mode Status Debugging Register

Sometimes an improper configuration setting to DSI, DISP paths and MUX may cause hanging issues on DSI because command mode control needs to wait for memory data from DMA in previous stages. As a result, a

debugging register DSI_STATE_DBG6 is designed for users to observe internal states of DSI controller in order to reduce debugging efforts. Table 6-38 shows the status mapping to the register bit [14:0]. Users can check the on-hot bit to find incorrect settings.

Table 6-38 Command Mode Status in Debugging Register

Bit	Value	Description
0	0x0001	Idle (wait command)
1	0x0002	Reads command queue for packet header
2	0x0004	Sends type-0 command
3	0x0008	Waits for data from the previous module to send type-1 command
4	0x0010	Sends type-1 command
5	0x0020	Sends type-2 command
6	0x0040	Reads command queue for packet data
7	0x0080	Sends type-3 command
8	0x0100	Sends BTA
9	0x0200	Waits for RX-read data
10	0x0400	Waits for software RACK for RX-read data
11	0x0800	Waits for peripheral TE
12	0x1000	Gets TE signaling
13	0x2000	Waits for software RACK for peripheral TE
14	0x4000	Waits for external TE

6.25.7.6 Video Mode

DSI supports video mode traffic sequences, including sync pulse mode, sync event mode and burst mode. To facilitate the translation of the parameters of packets, see the timing diagrams in the following sections for the corresponding register settings.

6.25.7.7 Sync-pulse Mode

A non-burst sync-pulse mode enables the peripheral to accurately reconstruct the original video timing, including sync pulse widths. A timing diagram for sync-pulse mode is shown in Figure 6-124, which also shows registers mappings to lines of VSA/VBP/VACT/VFP periods.

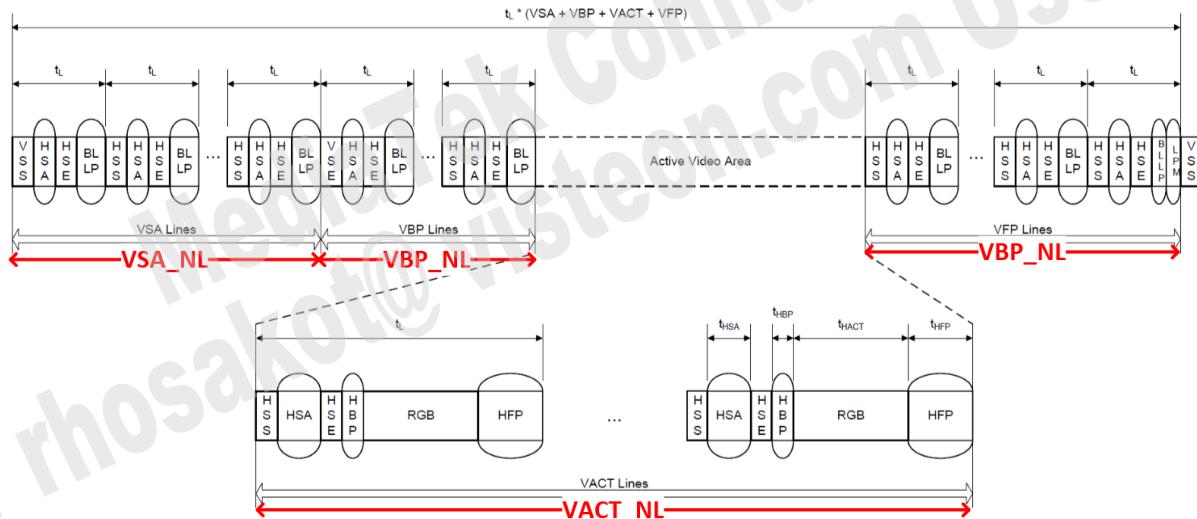


Figure 6-124 Non-burst Transmission: Sync-pulse Mode

A detailed timing diagram for one line period is shown in Figure 6-125. In this diagram, users must base on real timing parameters of HPW/HBP/HFP and pixel format to figure out the correct value of registers. In addition, a slight adjustment on HFP_WC should also be performed because it needs to leave some space time for HS preparation time due to non-continuous data lane transmission. The formulas for these registers are shown in Figure 6-126.

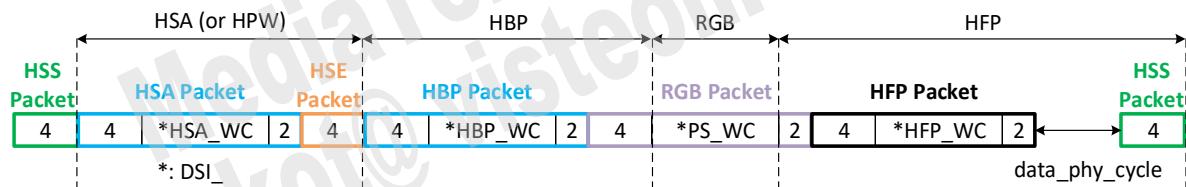


Figure 6-125 Sync-pulse Mode Line Period

$DSI_HSA_WC = HSA \times BPP - 10$
$DSI_HBP_WC = HBP \times BPP - 10$
$DSI_HFP_WC = HFP \times BPP - 12 - data_phy_cycle \times lane_num$
$data_phy_cycle = 1 + DA_HS_EXIT + LPX + DA_HS_PREP + DA_HS_ZERO + 1$
$DSI_PS_WC = active_pixel \times BPP$

Figure 6-126 Sync-pulse Mode Word-count Parameters

6.25.7.8 Sync-event Mode

A non-burst sync-event mode is similar to the pulse-sync mode, but accurate reconstruction of sync pulse widths is not required. Therefore, a single sync event is substituted. The timing diagram is shown in Figure 6-127.

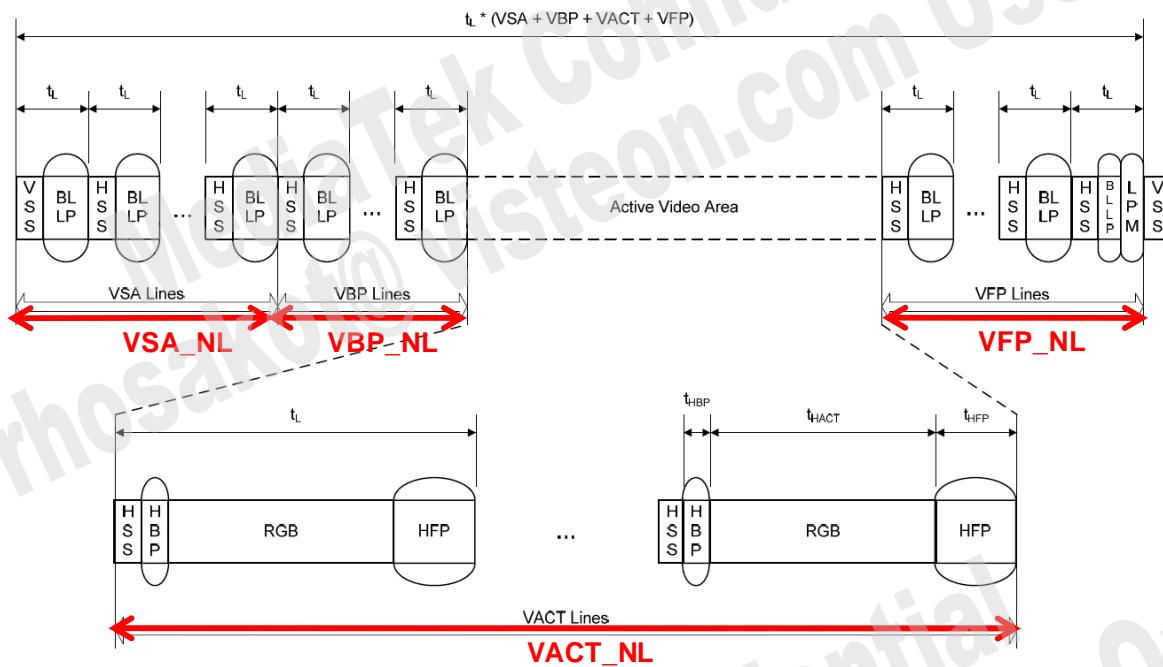


Figure 6-127 Non-burst Transmission: Sync-event Mode

A detailed timing diagram for one line period is shown in Figure 6-128, and the register settings of word count is listed in Figure 6-129.

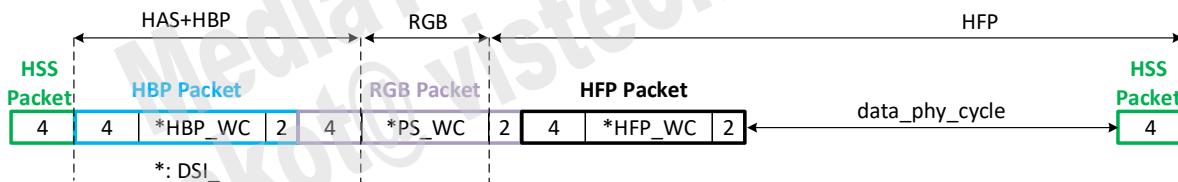


Figure 6-128 Sync-event Mode Line Period

DSI_HSA_WC = Don't care.
DSI_HBP_WC = (HBP+HSA) x BPP -10
DSI_HFP_WC = HFP x BPP -12 - data_phy_cycle x lane_num
data_phy_cycle = 1 + DA_HS_EXIT + LX + DA_HS_PREP + DA_HS_ZERO + 1
DSI_PS_WC = active_pixel x BPP

Figure 6-129 Sync-event Mode Word-count Parameters

6.25.7.9 Burst Mode

The burst mode allows RGB pixel packets to be time-compressed, leaving more time in a scan line for LP mode to save power or for multiplexing other transmissions onto DSI link. The timing diagram is shown in Figure 6-130.

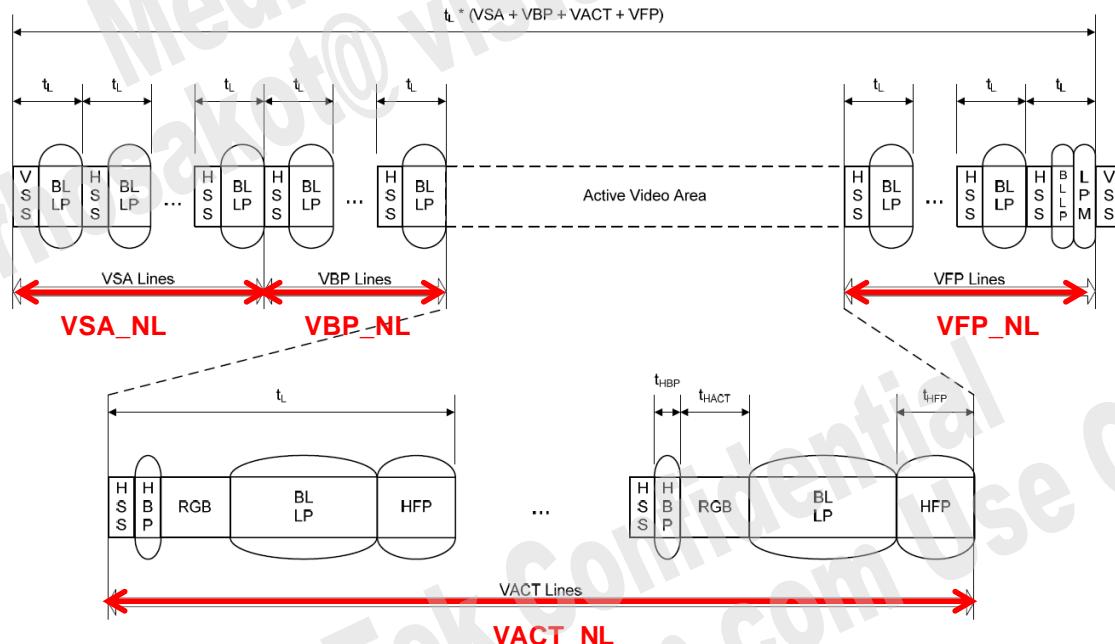


Figure 6-130 Burst Mode Transmission

Detailed timing diagram for one line period of burst mode is almost the same as that of sync-event mode shown in Figure 6-128 and Figure 6-129, except that the BLLC_WC register should be set.

6.25.8 Clock Lane Non-continuous Low-power Control

Clock lane non-continuous low-power control is supported in the version of DSI, which allows clock lane to return to idle state after a command transmission or during blanking packet period in the video mode. Figure 6-131 shows the blanking packet period distributed in a video mode transmission, where the host transfers blanking packets or goes back to idle state.

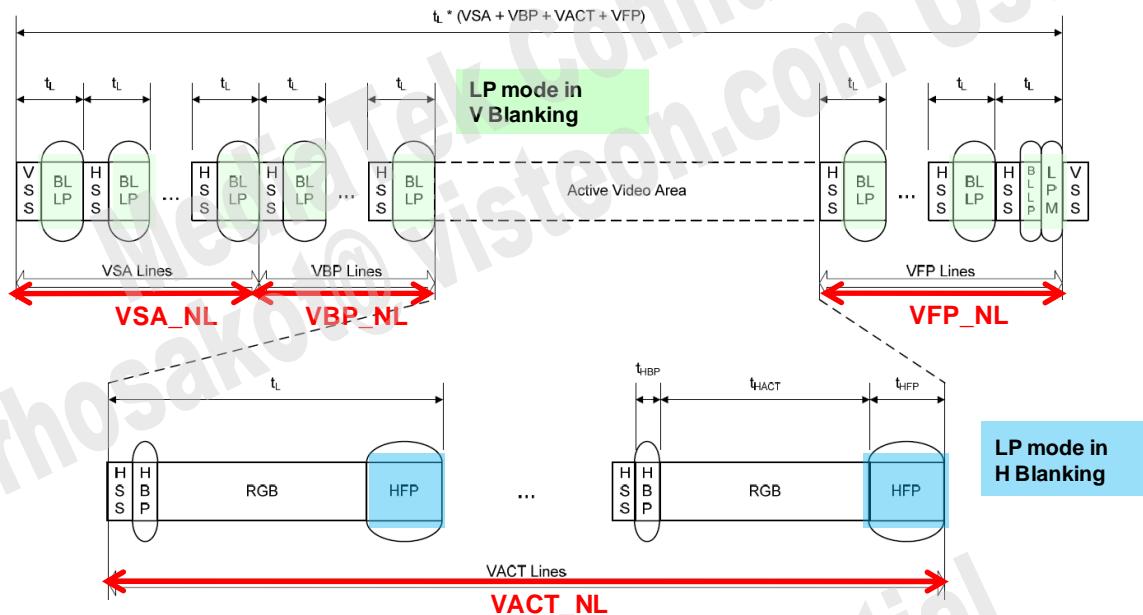


Figure 6-131 Blanking Packet in Video Mode Transmission

Figure 6-132 is the timing diagram of vertical blanking low-power control. This mechanism is through the register of DSI_HSTX_CKLP_WC, whose value is listed in Figure 6-132. In most cases, vertical blanking power-saving is suggested. If users would like to have more precise low-power control, the value set for VACT-line horizontal blanking low-power is acceptable. However, the timing should be calculated carefully from affecting the line period, since the clock/data lanes need to take a little time to go from the idle state to the high-speed mode before the next HS transmission.

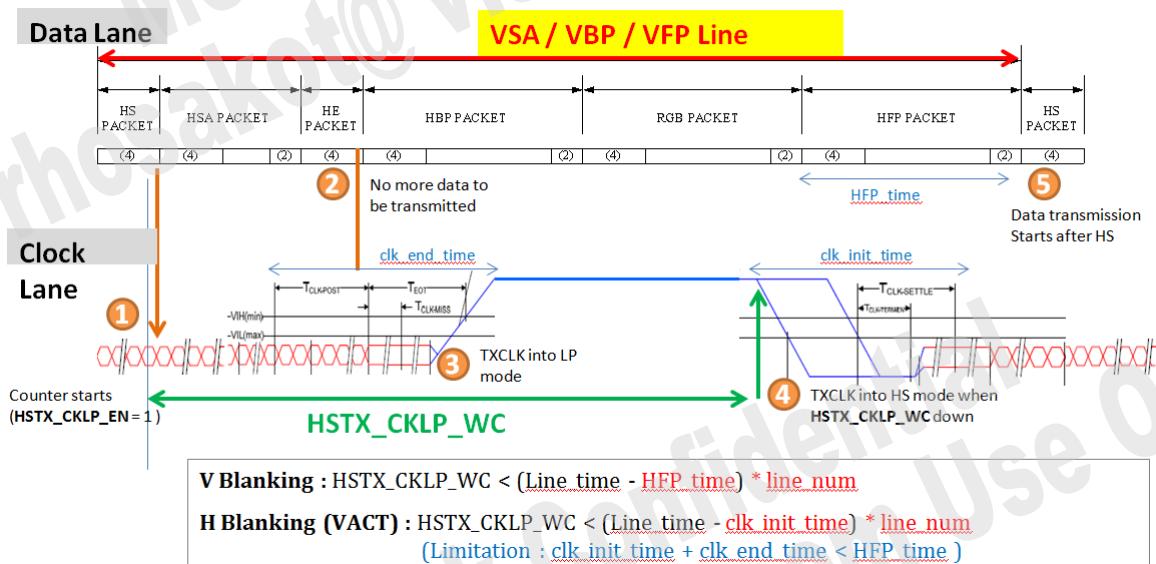


Figure 6-132 Low-power Counter Settings for Video Mode Low-power Control

6.25.8.1 Auto-calculation for Non-continuous Clock Word Counter

Because the calculation of word counter to control non-continuous clock recover is complicated, an auto-calculation register is provided in this version of design to simplify this flow. Users can set HSTX_CKLP_WC_AUTO in register DSI_HSTX_CKLP_WC to 1 to allow module to automatically figure out the required values instead of through manual calculation. In this mode, the value of HSTX_CKLP_WC is ignored.

6.25.8.2 Peripheral TE Detection

6.25.8.2.1 TE Signaling

DSI is capable of receiving peripheral Tearing Effect (TE) signals via BTA process. Before starting to receive TE signals from the peripheral, make sure a DCS command of “set_tear_on” is sent and register configuration of TE is enabled in the peripheral to avoid TE hanging issue. Table 6-39 is an example showing how to trigger TE commands by command queue.

Table 6-39 Example of TE Signaling Detection

Step	Description	R/W	Register [bit]	Value
1	Fill command queue: DCS “set_tear_on”	W	DSI_CMDQ_0	0x00351500
2	Fill command queue : get TE	W	DSI_CMDQ_1	0x00000020
3	Fill command queue : type-1 command	W	DSI_CMDQ_2	0x002C3909
4	Set command count	W	DSI_CMDQ_CON	0x3
5	Set memory write continue command	W	DSI_MEM_CONTI[15:0]	0x3C
6	Start command	W	DSI_START [0]	0x1
7	Issue interrupt and receive TE	R	DSI_INTSTA [2]	0x1
8	Read trigger status (TE)	R	DSI_RX_TRIG_STA [3:0]	0x2
9	Clear interrupt status	W	DSI_INTSTA [2]	0x1
10	Respond read acknowledgement to module and go to next commands in queue	W	DSI_RX_RACK [0]	0x1
11	Issue interrupt and command done	R	DSI_INTSTA [1]	0x1
12	Clear interrupt status	W	DSI_INTSTA [1]	0x1

6.25.8.2.2 External TE Pin

In certain cases, external TE pin instead of TE signals may be used for some reasons. DSI supports such mechanism to issue external TE interrupt signals. Refer to the sequences shown in Table 6-40 for detailed control information.

Table 6-40 Example of External TE Pin Detection

Step	Description	R/W	Register [bit]	Value
1	Enable external TE interrupt	W	DSI_INTEN [4]	0x1
2	Enable external TE	W	DSI_TXRX_CON [10]	0x1
3	Select external TE polarity (active-high)	W	DSI_TXRX_CON [9]	0x0
4	Issue interrupt and receive external TE	R	DSI_INTSTA [4]	0x1

Step	Description	R/W	Register [bit]	Value
5	Clear interrupt status	W	DSI_INTSTA [4]	0x1

6.25.8.2.3 Video Mode Extra Packet Transmission

DSI supports sending out an extra short/long packet during video mode transmission. This operation must use a set of special registers instead of the original command queue. The single extra packet is sent once and immediately when users trigger the VM_CMD_START bit of register DSI_START. Refer to the registers with prefix of "DSI_VM_CMD" for more detailed descriptions.

6.25.8.2.4 Short Packet

To transmit an extra short packet in the video mode, follow the example illustrated in Table 6-41.

Table 6-41 Example of Short Packet Transmission in Video Mode

Step	Description	R/W	Register [bit]	Value
1	Enable vm-cmd function	W	DSI_VM_CMD_CON [0]	0x1
2	Select short packet to be sent	W	DSI_VM_CMD_CON [1]	0x0
3	Enable sending period (VBP, VFP)	W	DSI_VM_CMD_CON [5:3]	0x6
4	Fill short packet ID and bytes 0~1	W	DSI_VM_CMD_CON [31:8]	0x001535
5	Enable vm-cmd done interrupt	W	DSI_INTEN [5]	0x1
6	Start vm-cmd transmission	W	DSI_START [16]	0x1
7	Issue interrupt and vm-cmd transmission done	R	DSI_INTSTA [5]	0x1
8	Clear interrupt status	W	DSI_INTSTA [5]	0x1

6.25.8.2.5 Long Packet

To transmit an extra long packet in the video mode, follow the example illustrated in Table 6-42.

Table 6-42 Example of Long Packet Transmission in Video Mode

Step	Description	R/W	Register [bit]	Value
1	Enable vm-cmd function	W	DSI_VM_CMD_CON [0]	0x1
2	Select long packet to be sent	W	DSI_VM_CMD_CON [1]	0x1
3	Enable sending period (VSA, VBP, VFP)	W	DSI_VM_CMD_CON [5:3]	0x7
4	Fill long packet ID and word count	W	DSI_VM_CMD_CON [31:8]	0x000739
5	Fill long packet data byte 0~3	W	DSI_VM_CMD_DATA0	0x332211FF
6	Fill long packet data byte 4~6	W	DSI_VM_CMD_DATA1	0x00665544
7	Enable vm-cmd done interrupt	W	DSI_INTEN [5]	0x1
8	Start vm-cmd transmission	W	DSI_START [16]	0x1
9	Issue interrupt and vm-cmd transmission done	R	DSI_INTSTA [5]	0x1
10	Clear interrupt status	W	DSI_INTSTA [5]	0x1

6.25.8.3 Dual DSI Display

Dual DSI mode is supported in this version to transmit a frame with eight data lanes and two clock lanes. To enable this feature, users should set up DSI0/DSI1 clocks to the same frequency, enable dual DSI mode in both DSI0 and DSI1, and trigger start bit to one of the modules for simultaneous control. The sequence is shown in Table 6-43.

Table 6-43 Dual DSI Mode Control Sequence

Step	Description	R/W	Module: Register [bit]	Value
1	Configure MIPI_TX0_CONFIG to enable DSI0 clock	-	MIPI_TX0_CONFIG	-
2	Configure MIPI_TX1_CONFIG to enable DSI1 clock	-	MIPI_TX1_CONFIG	-
3	Enable dual DSI mode for DSI0	W	DSI0 : DSI_COM_CON [4]	0x1
4	Enable dual DSI mode for DSI1	W	DSI1 : DSI_COM_CON [4]	0x1
5	Start transmission (either DSI0 or DSI1)	W	DSI0 : DSI_START	0x1

6.26 MIPI TX Configuration Module

6.26.1 Introduction

The MIPI TX configuration module (MIPI_TX_CONFIG) is used to control MIPI TX related registers for the MIPI DPHY macro. This analog macro includes design of SDM Phase-Locked Loop (PLL), bandgap, Low-Dropout (LDO) core, lane control, General Purpose Input (GPI) pads, and output enable, etc. The analog macro provides MIPI DSI clock lane with up to 750 MHz at a data rate of 1.0 Gbps and provides primary clocks to DSI module.

6.26.2 Features

The following functions of the MIPI DPHY can be controlled by this module.

- Bandgap control
- LDO core power and configuration
- SDM PLL configuration
- SSC control
- Analog function related settings and status
- Output pads control and electrical features
- GPI pads control
- Software-control mode for each lane
- Lane swap for clock and data lanes

6.26.3 Block Diagram

The block diagram of MIPI_TX_CONFIG module is shown in Figure 6-133.

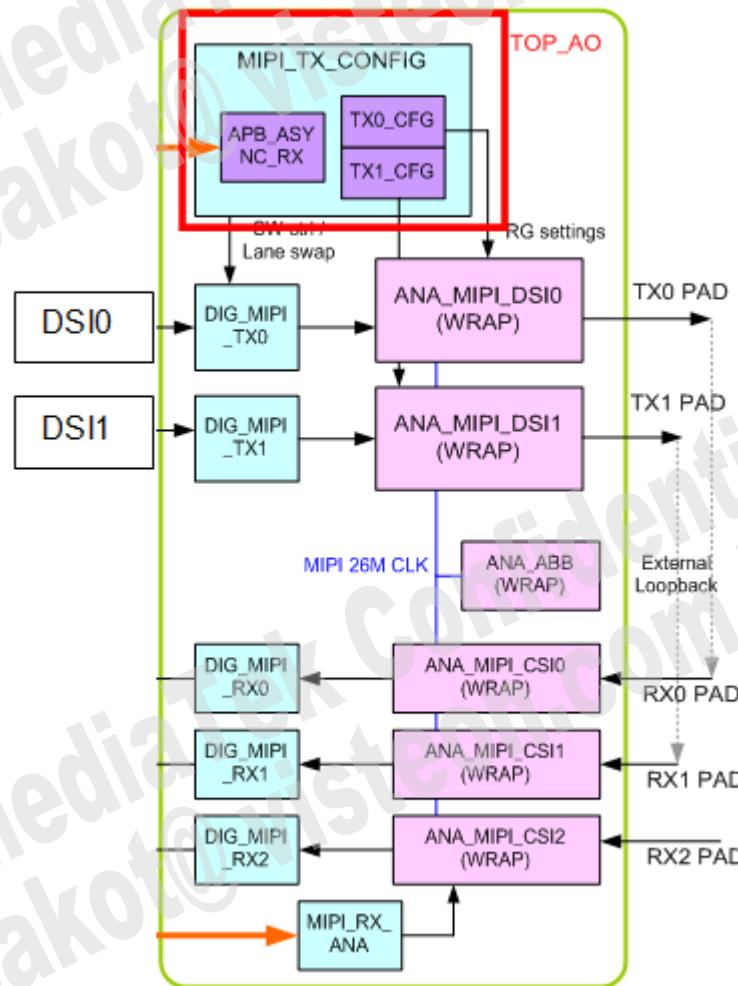


Figure 6-133 MIPI_TX_CONFIG Module's Block Diagram

6.26.4 Register Definition

For register details, please refer to Chapter 4.10.16 of “MT2712 IVI Application Processor Registers”.

6.27 DISP LVDS Encoder

6.27.1 Introduction

Display Low-Voltage Differential Signaling (LVDS) Encoder is used to generate LVDS format data for LCM interface, whether a dual-link or single-link LVDS.

6.27.2 Features

The LVDS encoder supports the following features:

- Dual Link & Single Link
- Max. resolution 1920×1080
- VESA standard 8-bit & 6-bit
- DISM standard 8-bit & 6-bit
- Built-in test pattern
- Bit select for RGB
- Channel swap

6.27.3 Block Diagram

Figure 6-134 is the block diagram of LVDS encoder in the path of mmsys.

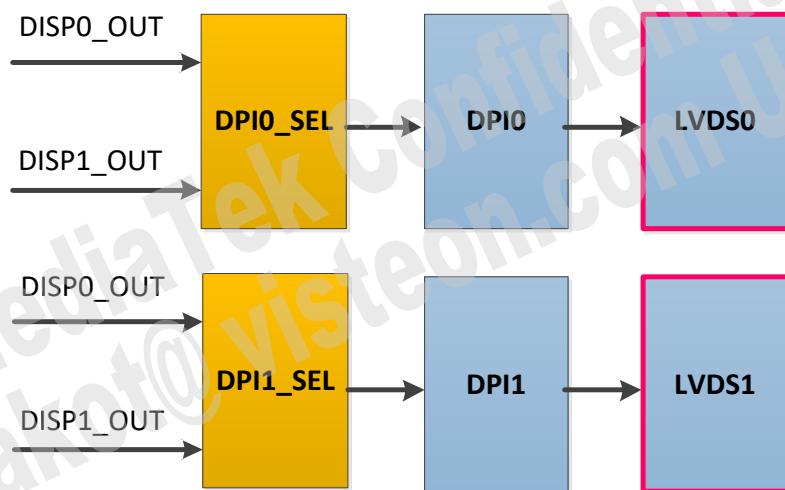


Figure 6-134 LVDS in MMSYS Path Block Diagram

Figure 6-135 is the block diagram of LVDS encoder's internal structure.

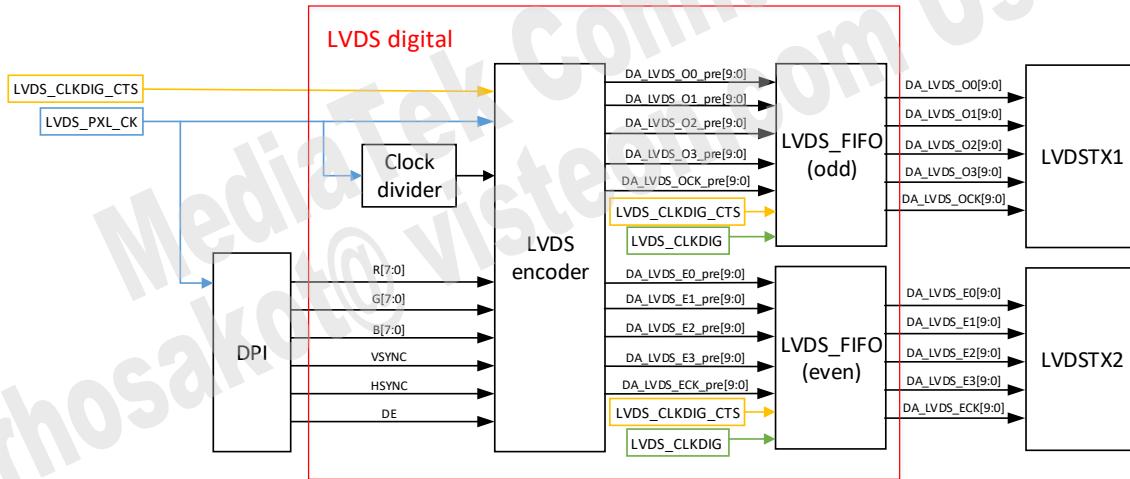


Figure 6-135 LVDS Encoder Block Diagram

Figure 6-136 is the block diagram of LVDS encoder datapath.

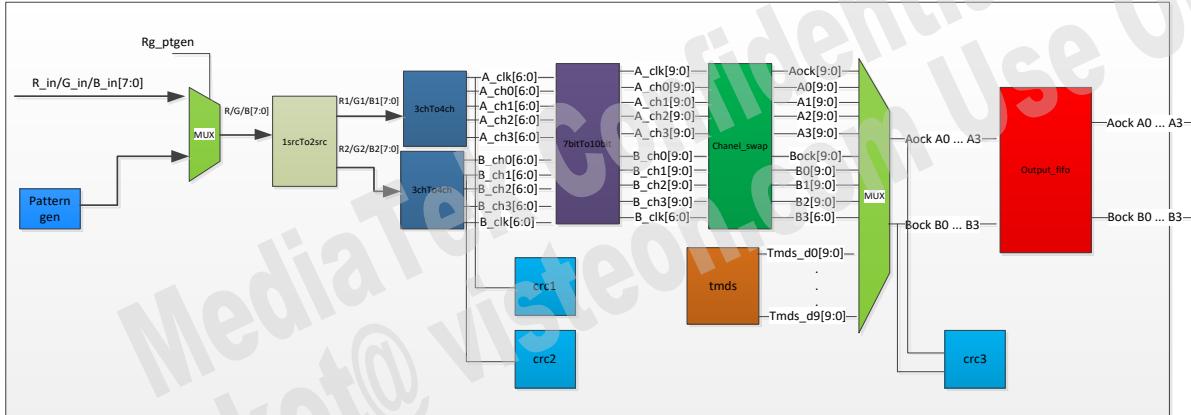


Figure 6-136 Block Diagram of LVDS Encoder Datapath

6.27.4 LVDS AC Timing

6.27.4.1 Introduction

- Low voltage differential signaling (LVDS) for high-speed data transmission.
- Key blocks Serialize & Driver & Bias & LDO & VOPLL.

6.27.4.2 Functional Specifications

Refer to the table below for the functional specifications of LVDS.

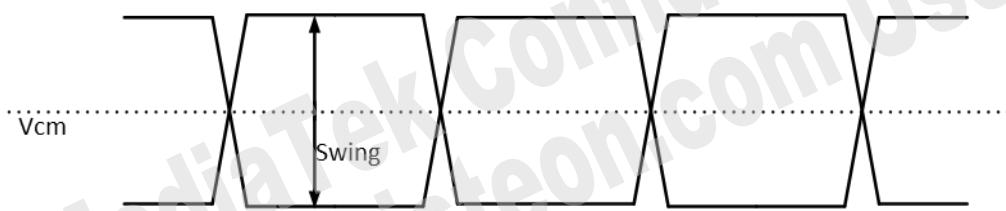


Figure 6-137 LVDS Waveform

Table 6-44 LVDS Specifications

Parameter	Min.	Typ.	Max.	Unit	Note
*IP Specification					
TX output swing	200	-	400	mV	Single-ended & differential is 400mV
Output Vcm	1.08	1.2	1.32	V	-
Pixel clock (single-link)	0.56	75	85	MHz	-
Pixel clock (dual-link)	1.12	150	170	MHz	CKDIG is kept the same as single link
Date rate per channel	3.92	525	595	Mb/s	-
Data rate (single-link)	3.92*4	525*4	595*4	Mb/s	-
Data rate (dual-link)	3.92*8	525*8	595*8	Mb/s	-
Data jitter (peak to peak)	-	-	300	ps	-
Test condition	Rload =100 Ohm, Cload=2pF				

6.27.5 LVDS Clock

There are two PLLs in the whole LVDS clock path, namely ANA_PLLGP (LVDSPLL) and ANA_MIPILVDS (VOPLL) as shown in Figure 6-138. LVDSPLL is the clock source of VOPLL that generates three clocks, LVDS_PXL_CK, LVDS_CLKDIG, and LVDS_CLKDIG_CTS, for LVDS interfaces.

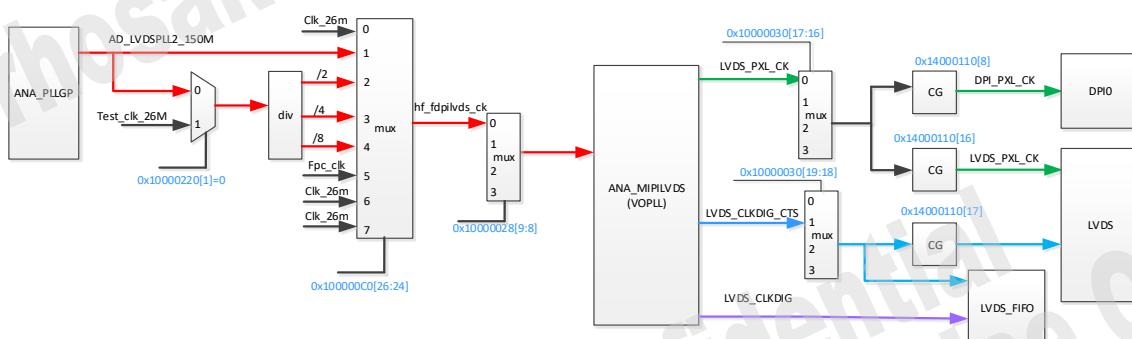


Figure 6-138 Overview of LVDS Clock Path

Figure 6-139 shows the block diagram of LVDSPLL. The frequency of voltage controlled oscillator ($F_{VCO_LVDSPLL}$) can be obtained from a reference input clock ($F_{REF_LVDSPLL}$) multiplied by a feedback divider (FBKDIV) whose ratio of the feedback divider is controlled by LVDSPLL_SDM_PCW as shown in the equation (1). It is a non-integer

value, and LVDSPLL_SDM_PCW[30:24] is integer portion of the value and LVDSPLL_SDM_PCW[23:0] is fractional portion of the value. Take 63.51 as an example, LVDSPLL_SDM_PCW[30:24] is 63 (integer) and LVDSPLL_SDM_PCW[23:0] is 51 (decimal). The recommended range of feedback divider should be set in between 15.0 to 123.0. The recommended operating frequency of VCO ($F_{VCO_LVDSPLL}$) is from 1.5 GHz to 3 GHz.

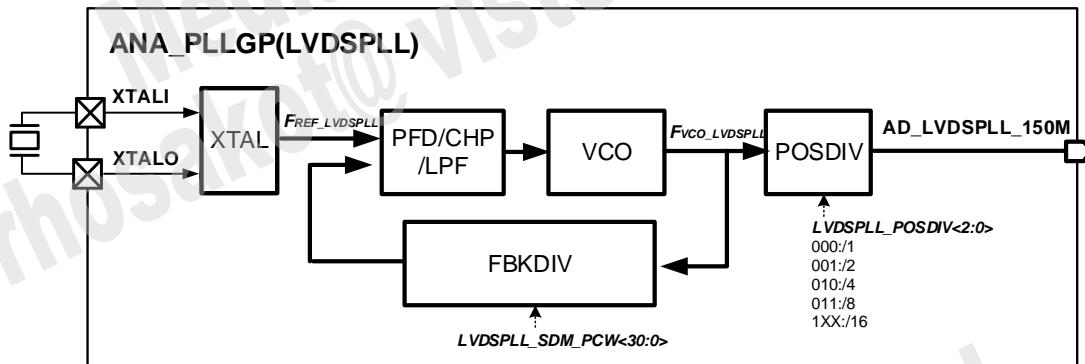


Figure 6-139 Block Diagram of LVDSPLL

- $F_{VCO_LVDSPLL} = F_{REF_LVDSPLL} \times \frac{LVDSPLL_SDM_PCW[30 : 0]}{2^{24}}$ (1)

The VOPLL provides clock signals and pixel clock for LVDS interface. MT2712 has two independent VOPLLs that are supplied to LVDS0/1 and LVDS2/3 separately. The block diagram of VOPLL is shown in Figure 6-140, and it contains a PLL followed with several clock dividers and multiplexers. LVDS_PXL_CK is the source of LVDS pixel clock. LVDS_CLKDIG and LVDS_CLKDIG_CTS provide the clocks with identical frequency for LVDS module and LVDS FIFO as shown in Figure 6-135 and Figure 6-138. In equation (2) and (3), the frequency of LVDS_CLKDIG_CTS has to meet the conditions with pixel clock (LVDS_PXL_CK) according to either single link or dual link.

- Single link: $LVDS_PXL_CK \times 7 = 10 \times LVDS_CLKDIG_CTS$ (2)

- Dual link: $LVDS_PXL_CK \times 7 = 20 \times LVDS_CLKDIG_CTS$ (3)

The following two equations illustrate how to calculate the frequency of output clock (F_{OUT_VOPLL}) and the frequency of voltage controlled oscillator (F_{VCO_VOPLL}) for VOPLL. It is recommended to supply a clock signal within a range of 15 MHz to 30 MHz as reference clock (F_{REF_VOPLL}) and the frequency of VCO (F_{VCO_VOPLL}) should operate at between 1.5 GHz to 3 GHz. In addition, the sum of TX_DIV1 and TX_DIV2 should be equal to or greater than 2.

$$F_{VCO_VOPLL} = F_{REF_VOPLL} \times \frac{RG_VPLL_FBKDIV[6:0] \times RG_VPLL_FBKSEL[1:0] \times RG_VPLL_VCO_DIV_SEL}{RG_VPLL_PREDIV[1:0]} \quad (4)$$

$$F_{OUT_VOPLL} = F_{REF_VOPLL} \times \frac{RG_VPLL_FBKDIV[6:0] \times RG_VPLL_FBKSEL[1:0]}{RG_VPLL_POSDIV[2:0] \times RG_VPLL_PREDIV[1:0]} \quad (5)$$

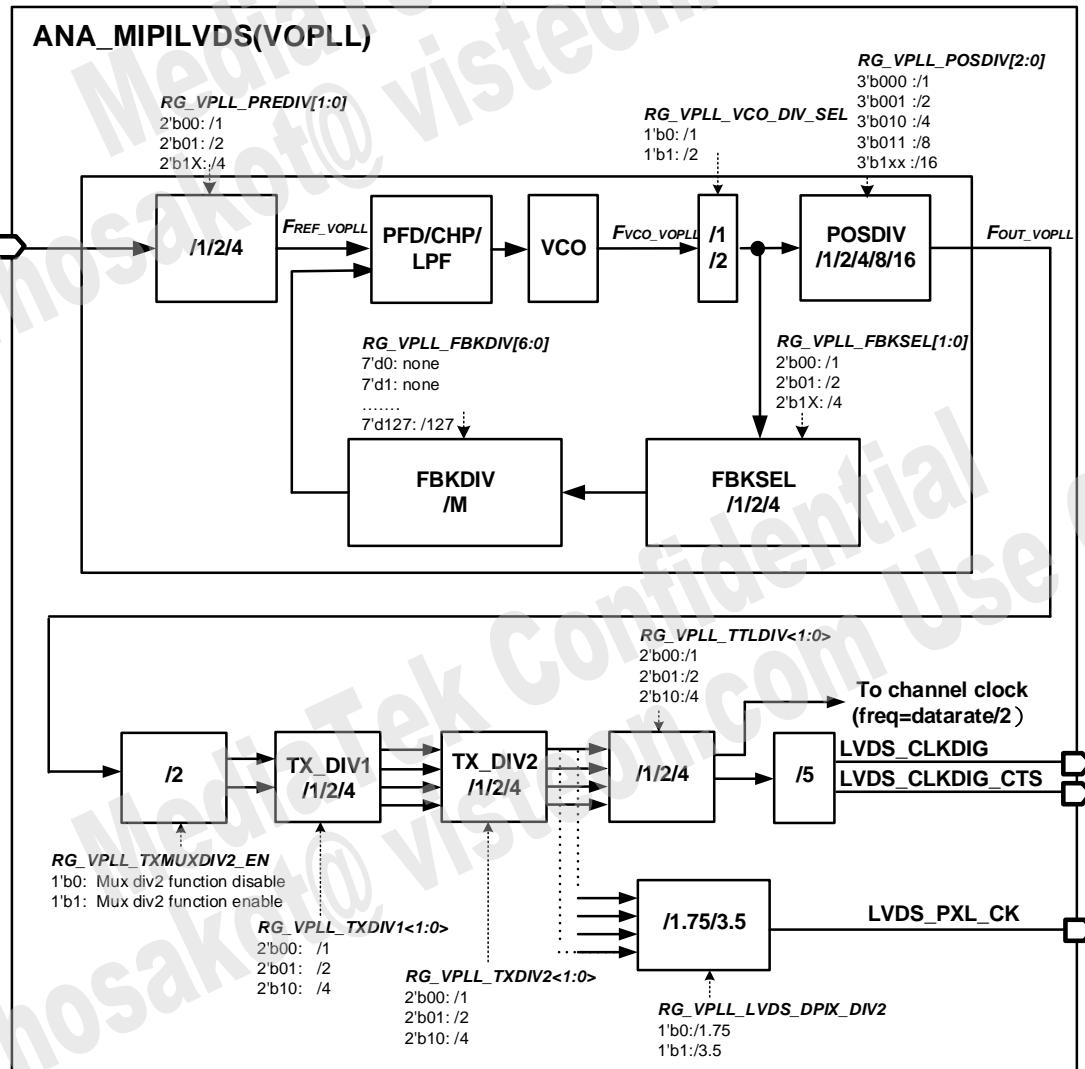


Figure 6-140 Clock Path of VOPLL

6.27.6 LVDS Encoder Standard

LVDS encoder has two standards: VESA and DISIM.

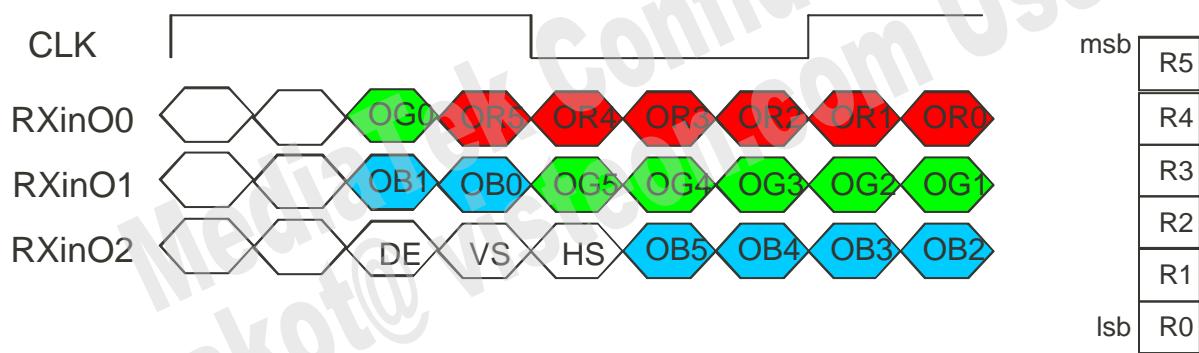


Figure 6-141 Single Link 6-bit VESA Standard

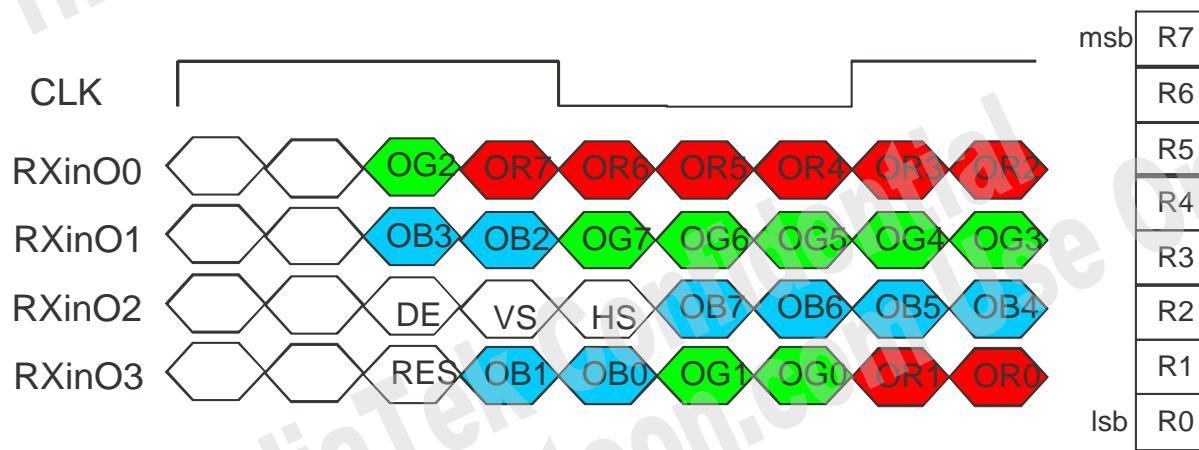


Figure 6-142 Single Link 8-bit DISM Standard

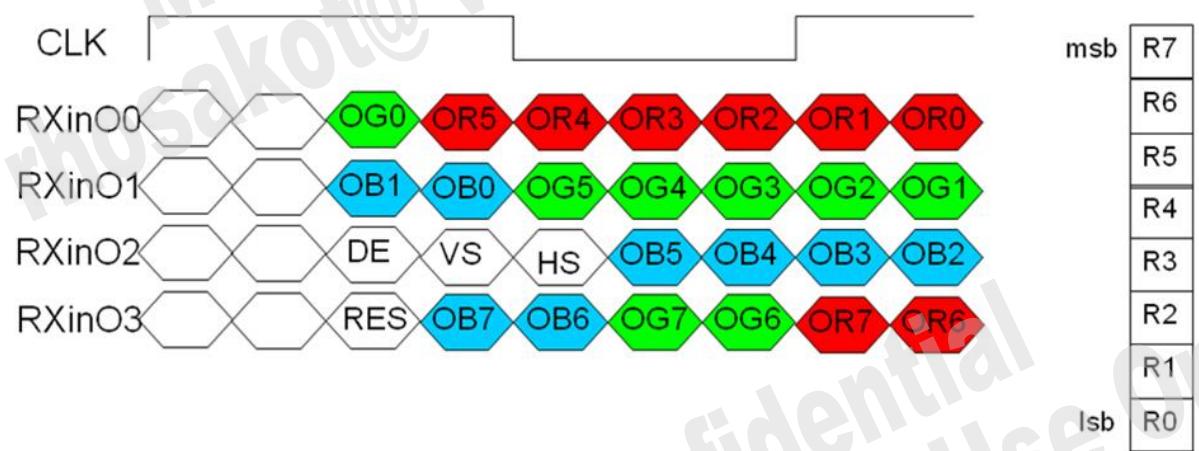


Figure 6-143 Single Link 8-bit VESA Standard

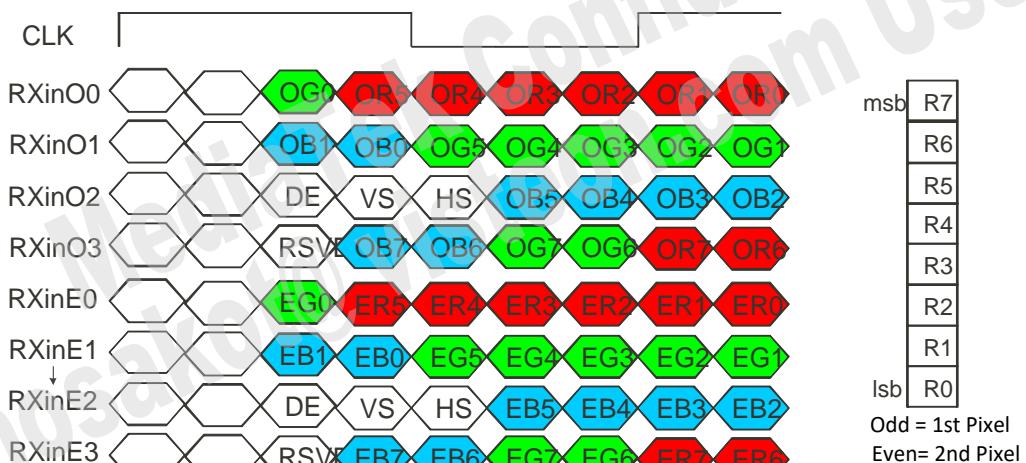


Figure 6-144 Dual Link 8-bit VESA Standard

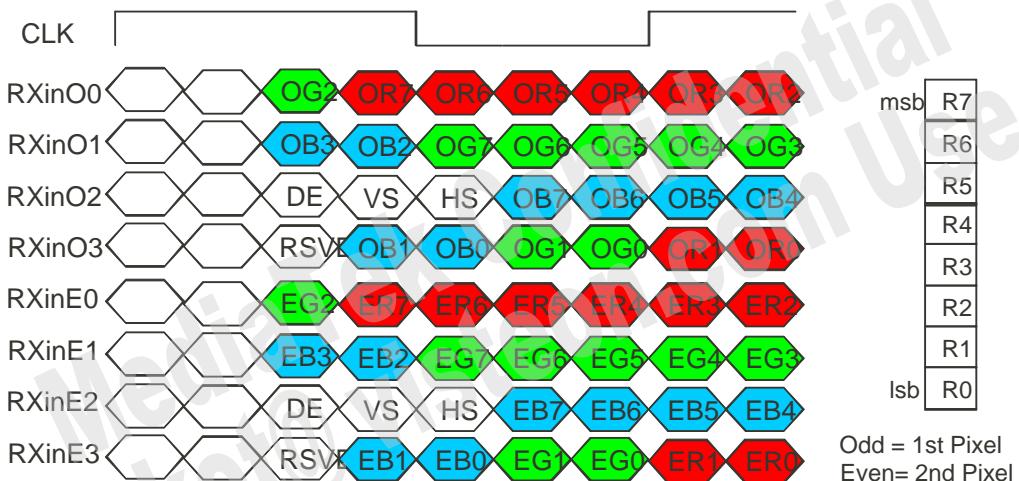


Figure 6-145 Dual Link 8-bit DISM Standard

6.27.7 DC/AC Parameters

Table 6-45 DC/AC Parameters

LVDS Electrical	Min.	Typ.	Max.	Unit
TX Output Swing ¹	200	-	400	mV
Output VCM	1.08	1.2	1.32	V
Pixel Clock (single-link)	0.56	75	85	MHz
Pixel Clock (dual-link) ²	1.12	150	170	MHz
Date rate per channel	3.92	525	595	Mb/s

¹ Single-ended and differential voltage is 400 mV.

² CKDIG is kept the same as single link.

LVDS Electrical	Min.	Typ.	Max.	Unit
Total Data rate(single-link)	3.92*4	525*4	595*4	Mb/s
Total Data rate (dual-link)	3.92*8	525*8	595*8	Mb/s
Data Jitter (peak to peak)	-	-	300	ps

Test conditions: R_{load} is 100 ohm and C_{load} is 2 pF.

6.27.8 Chanel Swap

The LVDS encoder has two types of channel swaps: rgb input channel swap and LVDS output channel swap.

RG_PANEL_IN_B/ RG_PANEL_IN_G/RG_PANEL_IN_R are for LVDS input RGB channel swap.

RG_LLV(0-3)_SEL/RG_LLV_(5-8)_SEL/RG_LLV_CK0_SEL/RG_LLV_CK1_SEL are for LVDS output channel swap.

Refer to the register definition for details.

6.27.9 Register Definition

For register details, please refer to Section 4.9.17 of “MT2712 IVI Application Processor Registers”.

6.27.10 Programming Guide

The LVDS encoder is located after dpi0 in Display System (Dispsys), and the dpi0 is located after rdma0 and rdma1. The main usage is that rdma0 or rdma1 reads source from DRAM, and sends it to dpi0 in a fixed timing which is decided by LVDS panel size, like 1280×800. LVDS encoder will encode dpi0 output RGB into VESA or DISM format, and then send it to analog macro LVDS PHY.

Before using LVDS encoder, analog macro needs to first set LVDS PLL, VPLL and LVDSTX.

After clock setting is completed, rdma0 or rdma1, dpi0 and LVDS encoder should be set up. For the LVDS usage, please refer to the register definition.

6.28 DISP_PWM Generator

6.28.1 Introduction

The Display Pulse-width Modulation (DISP_PWM) generator provides PWM signals for the Liquid Crystal Display (LCD).

6.28.2 Features

The module has the following features:

- DISP_PWM IO level: 1.8 V or 3.3 V
- Operating clock: 26 MHz (default) or 104 MHz
- DISP_PWM output frequency range: 6.2 Hz~13 MHz (26 MHz as operating clock) or 24.8 Hz~52 MHz (104 MHz as operating clock)
- DISP_PWM duty cycle range: 0%~100%; 1024 steps
- Gradual PWM control

6.28.3 Block Diagram



Figure 6-146 DISP_PWM Block Diagram

6.28.4 Register Definition

For register details, please refer to Chapter 4.10.18 of “MT2712 IVI Application Processor Registers”.

6.28.5 Programming Guide

Following is the programming flow:

1. Turn on DISP_PWM’s operating clock.
2. Get Multimedia System (MMSYS) mutex.
3. Set up DISP_PWM_CON_0 and DISP_PWM_CON_1.
4. Write DISP_PWM_EN = 1.
5. Release MMSYS mutex.

6.29 Display Mutex

6.29.1 Introduction

Display Mutex (DISP_MUTEX) is used to synchronize the start trigger signal of all submodules in display path. Because the display path is flexible and multi-tasking, there are up to 10 mutex cores in parallel.

6.29.2 Features

DISP_MUTEX supports the following features:

- Support up to 10 mutex cores in parallel.
- Each submodule can be assigned to any of the 10 mutex cores through setting registers.
- The start trigger signal for each mutex can come from SW driven or display interface (DSI, DPI) driven, depending on user’s selection.

6.29.3 Block Diagram

DISP_MUTEX is composed of multiple symmetric mutex_cores. Each mutex_core contains enough control_points to issue the trigger pulses to all the function modules simultaneously. All the control_points

inside a single mutex_core share the same start trigger pulse. As to where the start pulse comes from, users can select between display panel interface and software.

Through the register setting, users can select which mutex_core the trigger pulse of each function module comes from. Figure 6-147 illustrates the DISP_MUTEX architecture.

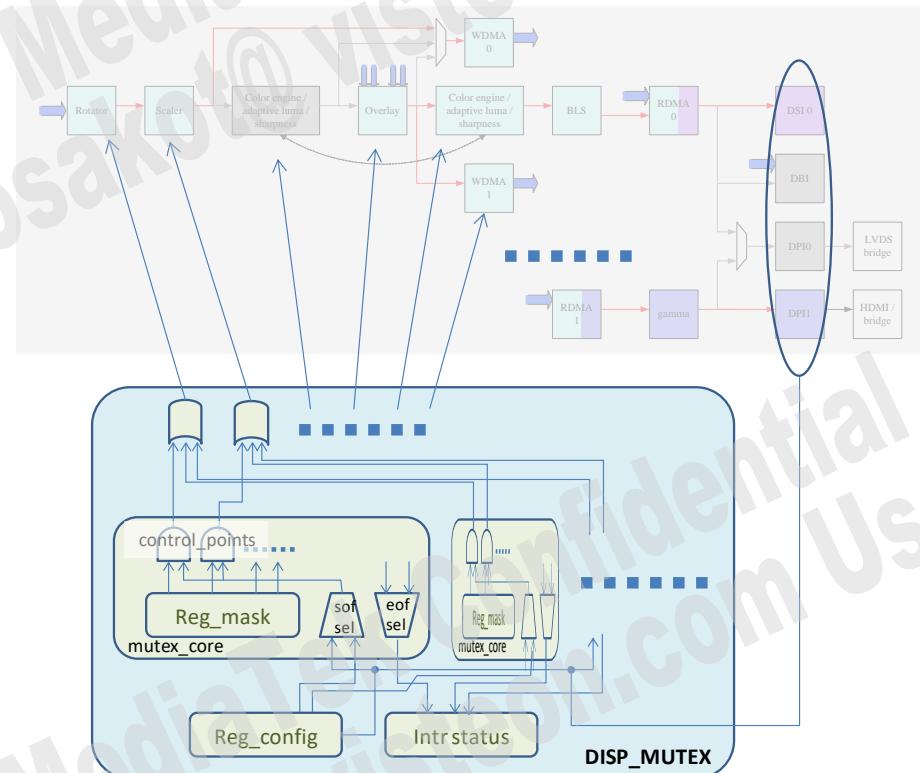


Figure 6-147 DISP_MUTEX Block Diagram

6.29.4 Register Definition

For register details, please refer to Chapter 4.10.19 of “MT2712 IVI Application Processor Registers”.

6.29.5 Programming Guide

6.29.5.1 Software Flow

The start pulse can come from display panel interface or software, depending on user’s selection. It is called single mode when the start pulse comes from software, and refresh mode when it comes from display interface. Figure 6-148 and Figure 6-149 show the programming sequences for these two modes respectively.

- Single mode
 - One software (SW) trigger only processes one frame
 - Memory in, memory out path (not always being in single mode)

- Memory in and direct link to command mode display output, e.g. DBI and DSI command mode
- Refresh mode
 - Process frame-by-frame after start
 - Memory in and direct link to video mode display output, e.g. DSI video mode, DPI, LVDS, HDMI, TV

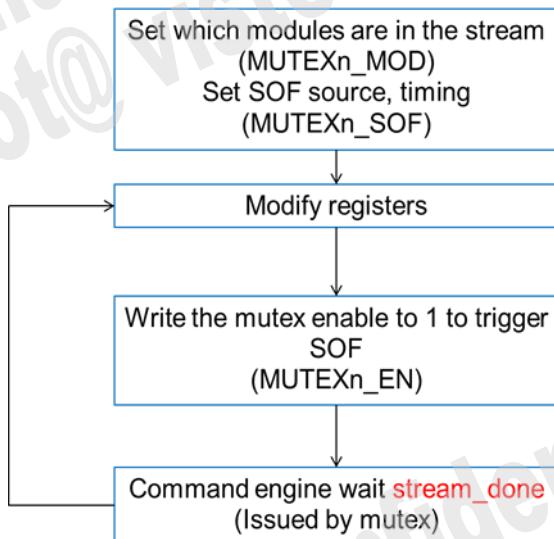


Figure 6-148 Refresh Mode Programming Sequence

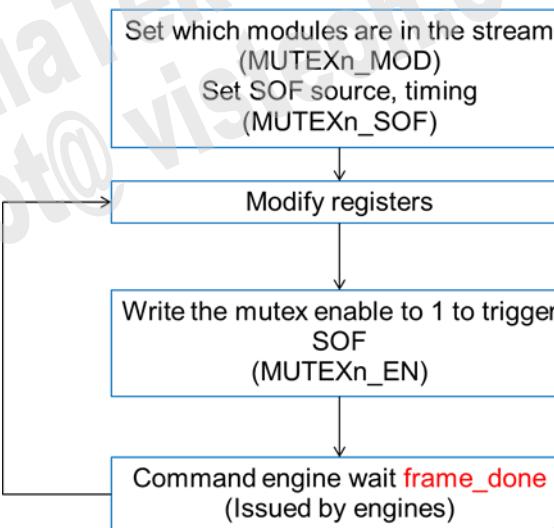


Figure 6-149 Single Mode Programming Sequence

- In the single mode, enable (DISP_MUTEXn_EN) will be de-asserted automatically after SOF is issued.
- Reset
 - Enable (DISP_MUTEXn_EN) will be automatically de-asserted when SW resets mutex (DISP_MUTEXn_RST)

- To reset stream
 - Reset mutex, and then reset engines
- Always use reset to disable mutex
- Interrupt source
 - Start of Frame (SOF)
 - Stream done
 - End of Frame (EOF) – DISP_RDMA underrun

6.30 Smart Multimedia Interface

6.30.1 Introduction

Smart Multimedia Interface (SMI) is a MediaTek's proprietary interface, which is used in multimedia systems. The SMI bus fabric deals with the complex bus interconnection and memory access transaction in a high-performance multimedia-rich system. The SMI bus fabric is separated into two parts for hierarchical arbitration. The SMI local arbiter is used for the first level arbitration for part of multimedia engines. The second level arbiter needs to access grants from the first level arbiter.

6.30.2 Features

SMI_COMMON supports the following features:

- Auto clock gating for power reduction
- Arbitration among requests from local arbiters to EMI
- Bandwidth/outstanding limiter
- Performance monitor
- Command throttling for reducing latency

SMI_LARB supports the following features:

- First level arbitration of multimedia engines
- GMC/SMI protocol to AXI protocol conversion
- Bandwidth regulation for each master
- Performance monitor enables performance index measurement

6.30.3 Block Diagram

The basic block diagram of SMI local arbiter is shown in Figure 6-150.

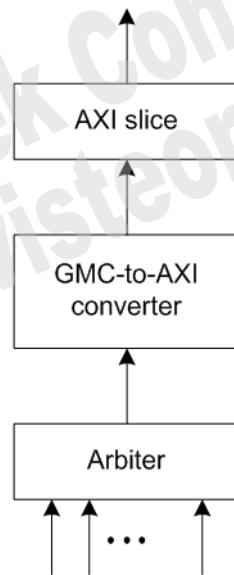


Figure 6-150 SMI Local Arbiter Block Diagram

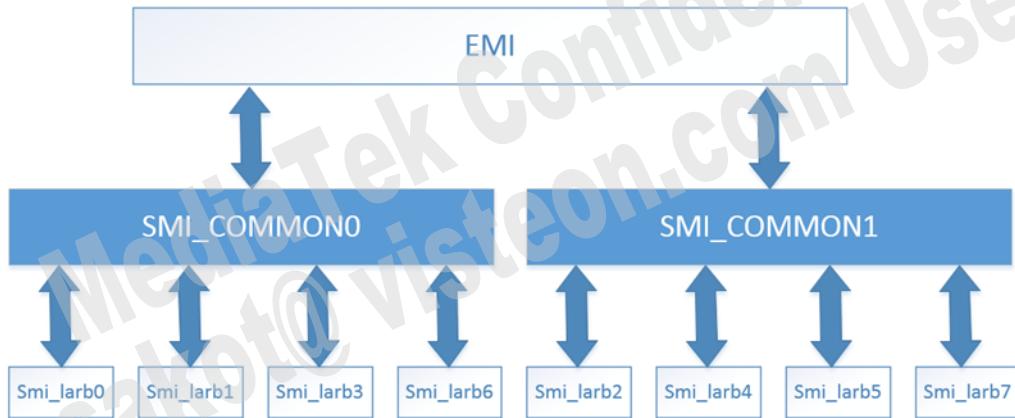


Figure 6-151 SMI_COMMON and Neighboring Blocks

6.30.4 Register Definition

For register details, please refer to Chapter 4.10.20 of "MT2712 IVI Application Processor Registers".

6.31 Multimedia Memory Management Unit (MM_IOMMU)

6.31.1 Introduction

The Multimedia (MM) engine usually require a certain amount of contiguous memory region. In order to avoid memory fragmentation, the Operating System (OS) needs to perform memory copy to align the requested ranges with one another. However, excessive memory copy may greatly degrade the system performance. For

some performance-critical engines, a common method to guarantee sufficient memory available for them is to statically reserve the physical memory. This solution is straightforward, but expensive since a large amount of memory is reserved for a certain engine without sharing with the others.

The MM_IOMMU is therefore designed to solve the fragmentation problem by paging the memory space, and to reduce the cost of static memory reservation. Each page is in the size of 4 KB, 64 KB, 1 MB, or 16 MB and can inherently be self-aligned without requiring any memory copies. In terms of memory usage, each requester is serviced on a dynamic memory allocation basis handled by the OS. With the choice of various page sizes, chances for internal fragmentation happening are lower and the flexibility of virtual memory usage is higher. The more frequently larger page sizes are used, the fewer the page table entries (PTEs) are required. Thus, fewer page table walks are taking place. This means the system performance may be increased due to fewer DRAM accesses. However, for a low-cost smartphone solution, it is always that the smaller the physical memory size, the better. With a smaller unit of page size, e.g. 4 KB, there will be ample page numbers available in the system. Each engine will have even chances to obtain the pages from the OS. The choice between different page sizes remains a trade-off, and is highly dependent on the DRAM specification. In this system, a smart software memory allocator is implemented, which chooses the best-fit sizes to make up the fewest PTEs. For example, for a requested range of 17 MB and 68 KB, the allocator will try to request for a 16 MB super-section, a 1 MB section, a 64 KB large page, plus a 4 KB page as the optimal solution. The major function of MM_IOMMU is to translate a virtual address into a physical one by performing page table look-up. By adopting the MM_IOMMU, the memory space is a virtualized and contiguous address, seen by the MM engines. Along with the page table managed by the OS, it no longer needs to statically reserve physical memory on system-up. The physical memory is allocated dynamically in the unit of 4 KB, 64 KB, 1 MB, or 16 MB by the OS. From the viewpoint of system cost, the physical memory size required on system-up is greatly reduced.

The MM_IOMMU supports two-level translation. The first layer is in a coarse-grained 1 MB section or 16 MB super-section, while the second layer is in a fine-grained 4 KB page or 64 KB large page. The major benefit of two-level translation is that the storage required by second-level page table itself is allocated dynamically in the size of 4 KB page frame after system-up. Only 16 KB storage for first-level page table should be statically reserved on system-up. Compared with one-level translation solution with 4 GB virtual address space, using two-level translation enables around additional 4 MB physical memory to be released and shared in the system. However, a two-level translation takes two page table look-ups. This may degrade the system performance, because the penalty of two complete DRAM latencies is added to the original DRAM latency of the requester. This problem can be solved by speculative prediction and caching, which will be mentioned in the later sections.

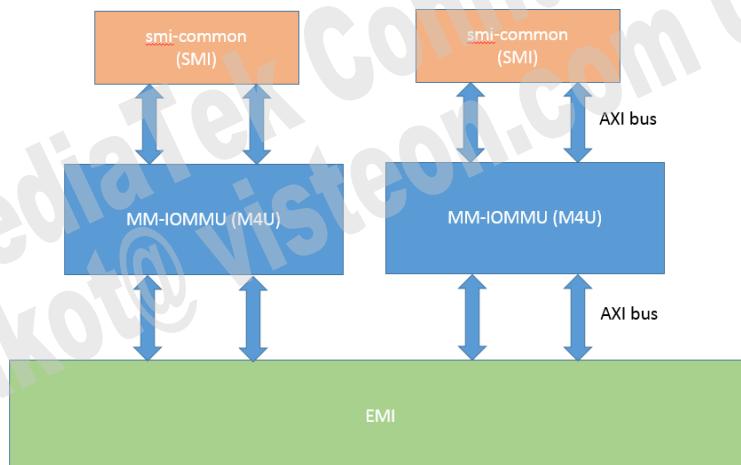


Figure 6-152 Simple System Architecture

In the system architecture, two MM_IOMMUs in total are implemented. In the figure above, one MM_IOMMU is placed between the Smart Multimedia Interface (SMI) and the External Memory Interface (EMI). The other is placed between the Peripheral Devices and EMI/CCI. The philosophy of this architecture is mainly considering the performance and cost aspects. Considering the latency penalty induced by page table look-up all the way from MM_IOMMU to EMI, which is called page table walk, the latency of an engine command is potentially doubled EMI latency for 1-level translation and tripled for 2-level translation. Therefore, it is reasonable to put the MM_IOMMU closer to the EMI to reduce additional EMI latency added to the overall latency of the actual engine command.

Since all MM engines, including the display engines, the image engines, the audio engines and the video engines, need the MM_IOMMU to perform address translation, the MM_IOMMU plays a critical role in MM performance, especially for hard real-time MM engines. The translation look-aside buffer (TLB), a major component in the MM_IOMMU, caches the page numbers to accelerate the translation process. The inherent hit rate of the TLB is quite high in that many MM engines' memory access behaviors are sequential in address. Thus, according to the temporal and spatial locality principles, the miss-rate of MM_IOMMU is expected to be very low. Besides having cache for page tables, the MM_IOMMU performs auto-prefetching to lower the impact of page table walks as well. Again, since many MM engines' behaviors can be predicted perfectly, the MM_IOMMU can simply perform "prefetching one page ahead" to further enhance the hit rate of the TLB. In addition, a novel non-blocking TLB architecture is proposed and implemented inside the MM_IOMMU to provide out-of-order and outstanding capabilities. With all these improvements, most of the latency overhead of two-level translation can be hidden and thus the performance overhead of page table look-up is nearly negligible.

6.31.2 Features

- Translates 32-bit virtual address into 33-bit physical address

- Two-level address translation in 4 KB, 64 KB, 1 MB, or 16 MB sizes
- Adopts a novel non-blocking cache architecture (miss-under-miss supported)
 - 16-entry Reservation Station (RS)
 - 16* 128-byte write data buffer
 - Multiple outstanding capability for a certain engine port
 - Out-of-order capability between different engine ports (different AXI IDs)
- Two-layer TLB structure
 - Banked-layer1 and unified-layer2 for multi-AXI channels support
 - Main TLB (L1): 16-entry and 48-entry fully associative caches with low cost Pseudo Least Recently Used (PLRU) replacement policy for layer1 and layer2 descriptors respectively
 - Prefetch TLB (L2): A 32 KB 256-set 4-way set-associative cache with LRU replacement policy
 - Victim TLB: A 4-entry fully associative cache with LRU replacement policy
- Entry lock feature is supported in the main TLB for critical pages
 - Partial range invalidation and full range invalidation are supported in the main TLBs, prefetch TLB and victim TLB
 - L2 TLB triggers an interrupt, once it finishes invalidation
- 8 sets of sequential ID are supported in each main TLB
 - Single-entry mode in the main TLB for scan-line based addressing
 - Each layer of page table in one sequential ID occupies only one main TLB entry
- Auto-prefetching feature is supported in prefetch TLB
 - Auto-prefetching for 2-layer page table to achieve zero-miss rate for scan-line based engine
 - Prefetch TLB pre-fetches the next page by performing current virtual address plus the 1 page (4 KB) by default
 - 16 sets of prefetch distance resources can be allocated for 16 IDs for +/- 1~15 pages for each main TLB.
- Security feature is supported for DRM requirements
 - Separate secure and non-secure page tables by different page table base addresses
 - Table selection is decided by the security type issued from the master engines
 - The security type of the transaction after finishing the secure table look-up is decided by the descriptor attribute
 - The security type of the transaction after finishing the non-secure table look-up is decided by the access type of the master engine
- Performance monitor, traffic monitor and debug feature support
 - Counters for miss count, auto-prefetch count, and total transaction count are available for miss-rate calculation for each ID
 - Counters in each RS entry for average translation latency calculation
 - Counters for read/write command/data statistics
 - Non-blocking, half-entry, and blocking modes are supported
 - Engine port ID log and page table layer log for translation fault and invalid physical address fault
 - Error hang feature for easier debug when exception occurs
 - Security debug mode for secure and non-secure registers isolation
 - Memory Assertion Unit (MAU) is supported for translation-level debug
- Low power and high performance TLB architecture

- With non-blocking TLB structure and broadcast mechanism, the overall SRAM read times are fewer than total translation requests for scan-line based engines.
- Fine-grained DCM is implemented: Each entry of RS, FIFO, and AXI slice has a hand-code CG.

6.31.3 Block Diagram

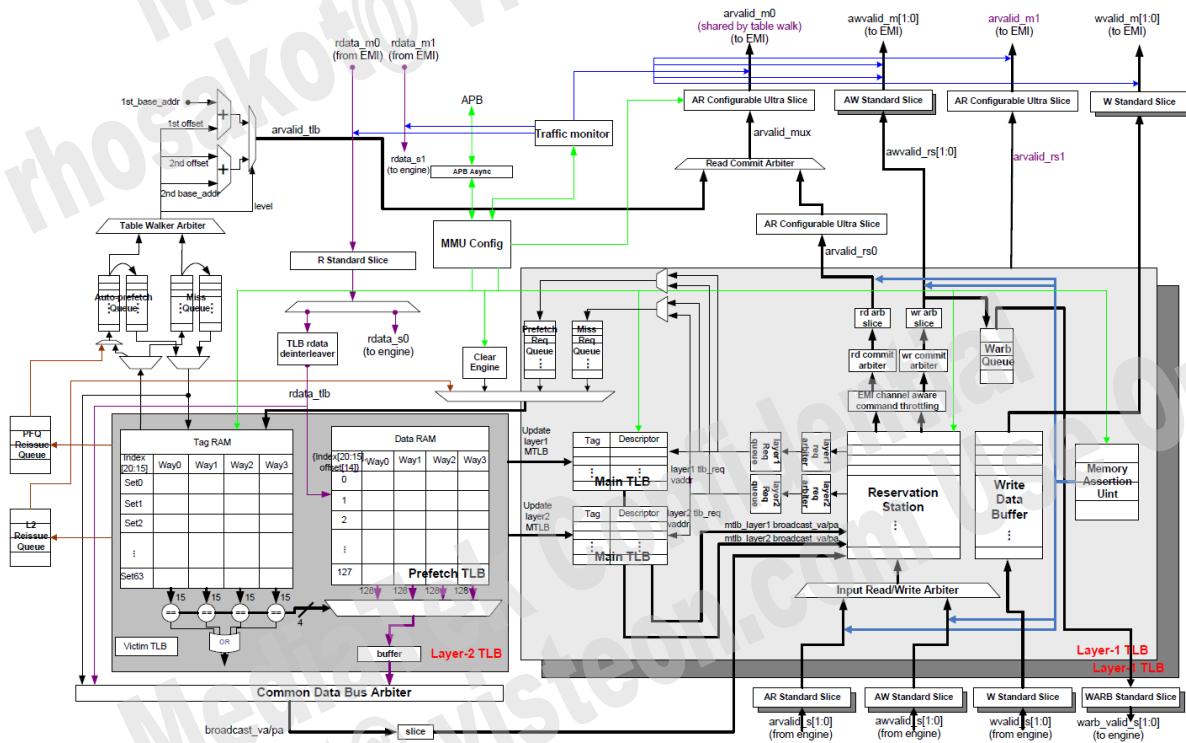


Figure 6-153 MM_IOMMU Block Diagram

6.31.4 Register Definition

For register details, please refer to Chapter 4.10.21 of “MT2712 IVI Application Processor Registers”.

6.32 Video De-interlacing

6.32.1 Introduction

The module Video De-interlacing (VDO) is used to de-interlace video.

De-interlacing: Converting an interlaced video stream to a progressive one.

The input source of de-interlacing is from DRAM. The output is sent to DISPFT and then written to DRAM through a write-channel.

Maximum operating frequency: 286 MHz

6.32.2 Features

The VDO has the following features:

- VDO supports full HD de-interlacing
- De-interlacing algorithm
- MDDI architecture
- Motion-Adaptive (MA)
 - 4-field old motion detection (SD/HD source)
 - 8-field old motion detection (SD source)
 - 3-field chroma motion detection
 - 24 division regions/region blending
- Fusion
- New pixel-based edge-preserving
- Motion/Comb count source detection (luma/chroma)
- Film mode detection (DTV)
- DRAM interface
- Block-based/scan-line mode for 420/422 (several swap modes)
- Support burst length read in scan-line mode and DTV swap mode
- Support burst length read in scan-line based packaged 10-bit data
- Horizontal sharpness
- Support H.265 interlace source
- Top/bottom field detection

6.32.3 Block Diagram

Figure 6-154 shows the block diagram of the VDO module in MT2712.

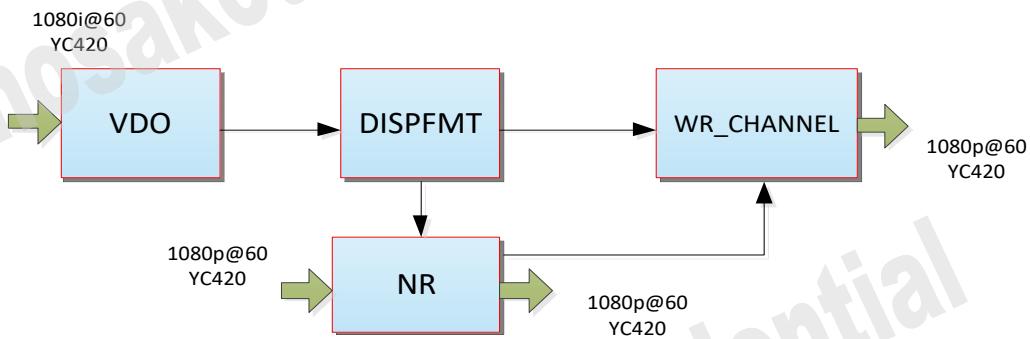


Figure 6-154 VDO Module Block Diagram

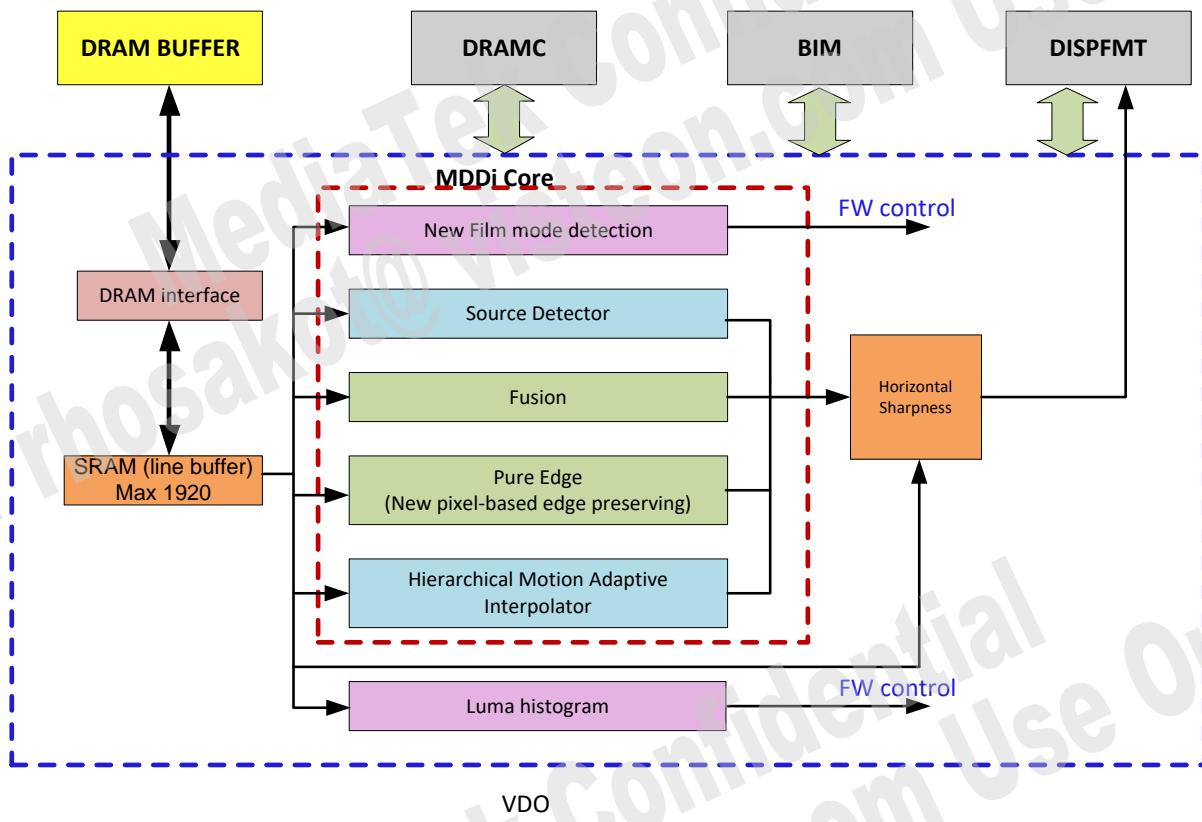


Figure 6-155 Internal Block Diagram of VDO

Media Direct De-interlacing (MDDi™) is an innovative progressive scan processing technology proposed by MediaTek. This technology utilizes three advanced techniques, which include intelligent source detector, hierarchical motion adaptive interpolator, and Pure Edge Engine (pixel-based).

Hierarchical Motion-Adaptive Interpolator analyzes the source content to distinguish the motion objects and the still objects, and then applies different interpolation methods to these objects. For motion objects, the conventional intra-field interpolation usually results in jagged edges. The patent-pending Pure Edge technique smooths out the staircase effects and provides visually pleasing images. Intelligent source detector or new film mode detection detects 3:2/2:2 pull-down sequences, and then re-assembles the correct frame source sequences. The luma histogram provides the luma information for video analysis.

6.32.4 Register Definition

For register details, please refer to Chapter 4.10.22 of “MT2712 IVI Application Processor Registers”.

6.32.5 Programming Guide

6.32.5.1 Normal Use

- Enable VDO clock
- Configure source data setting
- Configure HW SRAM utilization setting
- Configure frame buffer pointer setting
- Configure VDO mode
- Frame mode
- Field mode
- Intra mode with edge-preserving
- 4/5 field Motion-Adaptive (MA) mode
- 8-field MA mode
- Fusion
- Configure start line and sub-start line for different DI modes
- Configure VDO soft reset

6.32.5.2 Software Flow

6.32.5.2.1 Based Address

VDO: 15012xxxh

Based Address	Functional Description
150124xxh	Main control/motion detection
150127xxh	Luma-histogram
15012Fxxh	Film mode detection/CRC
150128xxh	Fusion/Pure-edge preserving

6.32.5.2.2 Engine Initialization – Software Reset

Besides the power-on reset, VDO contains the following reset:

6.32.5.2.2.1 VDO Software Reset

0x1501243c 0xFF

0x1501243c 0x0

6.32.5.2.2.2 Pulldown Counter Software Reset

0x15012460 0x1

0x15012460 0x0

6.32.5.2.2.3 Auto Reset for Each vsync

0x15012430[28] == 1'b1

6.32.5.2.3 Picture Size Settings

6.32.5.2.3.1 HBLOCK (0x15012410[7:0])

Description: The number of blocks per line stored in DRAM.

Value: Pixels per line/8, e.g. SD 720/8 = 'h5A

6.32.5.2.3.2 DW_NEED (0x15012410[15:8])/DW_NEED_HD (0x150124e0[8:0])

Description: How many double-words of pixels should be received from DRAM for SD (DW_NEED) or for HD(DW_NEED_HD).

Value: Pixels/4, e.g. SD 720/4 = 'hB4

6.32.5.2.3.3 PIC_HEIGHT (0x15012410[25:16])

Description: The height of the picture.

Value: Height of the source picture, e.g. n720P 720 = 'h2d0

6.32.5.2.3.4 HD_LINE_MODE (0x150124e0[20])

Description: Set source line as HD (line > 720).

6.32.5.2.4 Hardware SRAM

To cost down SRAM, some SRAMs are shared and merged, for which new control bit needs to be added (HD_MEM_1920).

Mode	HD_EN 0x150124e0[24]	HD_MEM 0x150124e0[22]	HD_MEM_1920 0x150124e0[21]
Source width <= 720 (field mode/frame mode/4-field mode)	1'b0	1'b0	1'b0
720 < source width <= 1920 (field mode/frame mode)	1'b1	1'b0	1'b0
720 < source width <= 1280 (4-field mode)	1'b0	1'b1	1'b0
1280 < source width <= 1920 (4-field mode)	1'b0	1'b1	1'b1

6.32.5.2.5 Frame Buffer Pointer

Pointer Y field indicates the current display field. There are two sequences of these pointers.

Five fields: (A) Z Y X W

Field	Start Address of Y	Start Address of C
W	15012480h	X
X	15012408h	1501240Ch
Y	15012400h	15012404h
Z	15012484h	150124FCh

Field	Start Address of Y	Start Address of C
A	150124ECh	X
	TOP (15012430[2] = 0)	Bottom (15012430[2] = 1)
	Addr 1 2 3	Addr 1 2 3
	A Y W	Z X
	Z X	A Y W

If Fusion is used, the A field pointer (0x150124ec) should be set as W field chroma DRAM address.

Field	Start Address of Y	Start Address of C
W	15012480h	150124ECh
X	15012408h	1501240Ch
Y	15012400h	15012404h
Z	15012484h	150124FCh
A	X	X

6.32.5.2.6 Vertical Shift

6.32.5.2.7 Start Line and Sub-Start Line

Start line and sub-start line means which source line should be displayed.

Data format: YUV420

Settings 1:

Source	Display Mode	Register	Values
SD field mode		0x15012420	0x07fb_07fa
SD 4-field mode		0x15012424	0x07fd_07fc
HD(720/1080) field mode		0x15012450	0x07fa_07f9
HD(720/1080) 4-field mode		0x15012454	0x07fe_07fd

Settings 2:

Source	Display Mode	Register	Values
SD frame mode		0x15012420	0x07fa_07f9
HD(720/1080) frame mode		0x15012424	0x07fd_07fc
		0x15012450	0x07fa_07f9
		0x15012454	0x07fd_07fc
		0x15012428	0x0000_0000
		0x1501242c	0x2060_2060

Data format: YUV422

Settings 1:

Source	Display Mode	Register	Values
SD field mode		0x15012420	0x07fb_07fa
SD 4-field mode		0x15012424	0x07fb_07fa
HD(720/1080) field mode		0x15012450	0x07fa_07f9

Source	Display Mode	Register	Values
HD(720/1080) 4-field mode		0x15012454	0x07fa_07f9
		0x15012428	0x8080_0000
		0x1501242c	0x8080_0000

Settings 2:

Source	Display Mode	Register	Values
SD frame mode HD(720/1080) frame mode		0x15012420	0x07fa_07f9
		0x15012424	0x07fa_07f9
		0x15012450	0x07fa_07f9
		0x15012454	0x07fa_07f9
		0x15012428	0x0000_0000
		0x1501242c	0x0000_0000

6.32.5.2.8 Address Swap Mode

Mode	VDO Setting	Description
SWAP_8520_0	0x1501247C[9]=0 0x1501241C[10]=0 (Hardware Default)	16*32 block 1 [No swap]
SWAP_8520_1	0x1501247C[9]=0 0x1501247C[16]=0 0x1501241C[10]=1	8520 swap mode 1
SWAP_8520_2	0x1501247C[9]=0 0x1501247C[16]=1 0x1501241C[10]=1	8520 swap mode 2
SWAP_5351_0	0x1501247C[9]=1 0x1501247C[1:0]=0	Top/Bottom fields in a MB are grouped individually
SWAP_5351_1	0x1501247C[9]=1 0x1501247C[1:0]=1	Two adjacent MBs in horizontal are exchanged
SWAP_5351_2 (support DRAM burst length)	0x1501247C[9]=1 0x1501247C[1:0]=2	Block 4 [64x32]
SWAP_5351_3	0x1501247C[9]=1 0x1501247C[1:0]=3	Block 1[16x32]

Enable DRAM burst length: 0x1501247c[14] == 1

6.32.5.2.9 De-interlacing Algorithm

De-interlacing has the following modes:

6.32.5.2.9.1 Frame Mode (Inter Mode)

15012430[1:0] = 2'b11, under this mode, top and bottom fields sources of de-interlacing are all from original source.

6.32.5.2.9.2 Field Mode (Intra Mode)

15012430[2] = 0 (top field from source, bottom field from interpolation)

15012430[2] = 1 (bottom field from source, top field from interpolation)

6.32.5.2.9.3 Intra Mode with Edge-preserving

15012478[11] = 1 turn on the intra mode with edge-preserving

15012478[11] = 0 turn off the intra mode with edge-preserving

6.32.5.2.9.4 4/5 Field Motion-Adaptive (MA) Mode (Included Chroma Motion)

Only turn on MA4F (15012488[24]). The other registers are set by hardware automatically.

1. 4-field:

-		F_PRGS 150124E0[31]	I2P_N_FRM 1501247C[21:20]	-		I_IN_P 150124E0[30]	F_L_SEL 150124E0[29]
MA4F 15012488[24]	0	0	0	PRGS	0	0	0
	1	1	1		1	0	1

2. 5-field: (The settings of 4-field above must be set too)

GET_5FLD 150124C0[4]	0		Not loading field A		
	WA_NA24 150124C0[5]	0	WA field difference is applied in 24 sub regions		
		1	WA field difference is not applied in 24 sub regions		
	MA5F 150124C0[23]	0	Disable 5-field MD		
		1	Enable 5-field MD	RBTH_WA 15012438h[2:0]	16*(WA_TH / WY_TH - 1/2)
				WA_VMV_FCH 150124A4h[19:12]	Threshold for WA further check

3. Chroma Motion-detection

CRM_3FMD (0x150124D0h) can enable chroma motion detection. Chroma motion detection is implemented by 3 fields.

CRM_3FMD 150124D0h[0]	0	Disable chroma motion detection		
	1	Enable chroma motion detection		Loading chroma Z field PTR_ZF_C (150124FCh)
	CRM_FDIFF 150124D0h[1]	0	Compared with CRM_DIFF(150124D0h[15:8])	
		1	Compared with blending TH when saw tooth is found. Otherwise, use CRM_DIFF	
	CRM_EXP_OFF 150124D0h[2]	0	chroma motion expansion	
		1	Disable chroma motion expansion	
	C_INTER_X 150124D0h[3]	0	Chroma inter mode uses X & Z field	
		1	Chroma inter mode uses X field only	

		C_VT_121 150124D0h[4]	0	Chroma VT 1111 mode
			1	Chroma VT 121 mode

6.32.5.2.9.5 8-field Motion-Adaptive (MA) Mode

Register:

R/W Pointer	Description		
MBAVG1 15012458h	DRAM read pointer for 6-field MD		
MBAVG2 1501245Ch	DRAM read pointer for 8-field MD		
MBAVG3 150124D8h	DRAM write pointer for 6/8-field MD		

6/8-field MD is based on 4-field MD. 6-field MD is enabled when both MA4F & MA6F are set to 1. 8-field MD is enabled when MA4F, MA6F & MA8F_OR are all set to 1.

Motion-Adaptive 4/6/8-field De-interlacing							
MA4F 15012488h[24]	0	Disable 4-field MD					
	1	MA6F 15012488h[31]	0	4-field MD	MA8F_OR 150124C0h[25]	0	6-field MD
			1			1	8-field MD

Enable 8-field MD, the results of 4-field MD will be written to DRAM. To protect other DRAM space from abnormal writing, PRT_WR_STA and PROT_WR_END can be set to 1; the protect space is programmable by PRT_STA_ADDR and PRT_END_ADDR.

Protect other DRAM Space from Mis-writing		
PROT_WR_STA 1501247Ch[12]	0	Protect DRAM space before VDO start address is disabled
	1	Protect DRAM space before VDO start address (DW): PRT_STA_ADDR (15012F54h)
PROT_WR_END 1501247Ch[13]	0	Protect DRAM space after VDO end address is disabled
	1	Protect DRAM space after VDO end address (DW): PRT_END_ADDR (150124DCh)

6.32.5.2.9.6 Fusion

Fusion local motion information (4 bit/per 8 pixels) will be written to DRAM and then be read from DRAM.

Therefore, the based DRAM address needs to be set. The maximum DRAM space needed is as follows:

Maximum DRAM space (1080p): $1920/8 * 4\text{bit} = 960 \text{bit} \approx [960/128] \approx 8 \text{OW}$ (8 *16 Bytes)

One frame totally needs: $8 * 540 = 4320 \text{OW}$ (4320 * 16 Bytes)

It also sets the DRAM resign by low bound and high bound to avoid mis-writing other DRAM address.

Related Register:

0x15012800[0] fusion_en

0x15012F40[23] fusion_output_data_en (default = 1)
0x15012870[29] en_Imr: enable DRAM read operation
0x15012870[28] en_Imw: enable DRAM write operation
0x15012870[25:0] fusion_flag_addr_base, the value should be DRAM logical address/16.
(e.g. if DRAM is 0xC4000000, the value is 0x400000)
0x15012874[25:0] da_flag_waddr_hi_limit (high bound)
0x15012874[25:0] da_flag_waddr_lo_limit (low bound)

6.33 Display Format

6.33.1 Introduction

Display Format (DISPFMT) is used to generate timing for VDO.

6.33.2 Features

DISPFMT has the following features:

- Generate timing
 - Configurable timing generate
 - Generate timing control signal to VDO, Noise Reduction (NR)
- 422 to 444
 - Repeat
 - Average
 - Adaptive copy between left and right pixels for the luma key
- Other
 - Background, border colors

6.33.3 Block Diagram

Figure 6-156 is the block diagram of DISPFMT in MT2712.

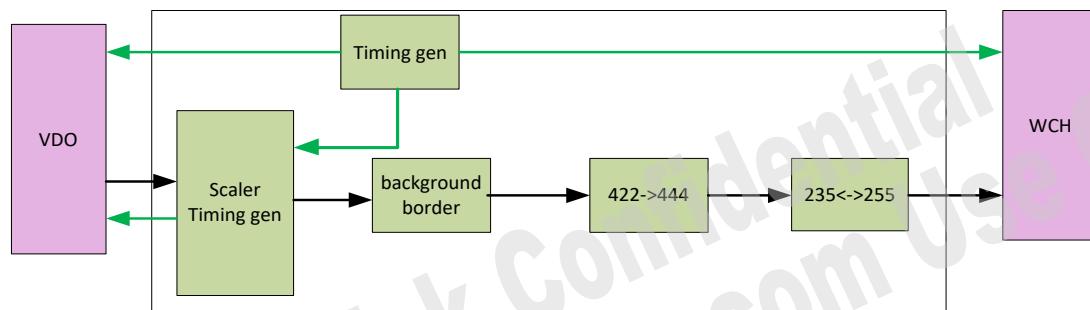


Figure 6-156 DISPFMT Block Diagram

6.33.4 Register Definition

For register details, please refer to Chapter 4.10.23 of “MT2712 IVI Application Processor Registers”.

6.33.5 Programming Guide

6.33.5.1 Normal Use

1. Enable DISPFMT clock
2. Configure DISPFMT output timing
 - 480p/576p
 - 720p
 - 1080p
3. Configure VDO related register
4. Configure DISPFMT soft reset

6.33.5.2 Software Flow

1. Power on DISPFMT/VDO clock
2. Set output timing setting
15011094 [31:30]: set tv_type; 10 for PAL timing, 00 for NTSC timing
150110D4: user defines V_total and H_total
150110a0: set horizontal output active region
150110a4: set odd field vertical output active region
150110a8: set even field vertical output active region
3. Set VDO setting
4. Soft reset
150110ac 0x400
150110ac 0x000

6.34 Noise Reduction (NR)

6.34.1 Introduction

Random noise is an inherent problem with all recorded images. The result is often called picture grain. Not only does noise get introduced during post-production editing or the final stage of video compression, but it is also present at the source in the form of film grain or imaging-sensor noise. Noise-reduction algorithms can minimize the grain in a picture.

The simplest approach to noise reduction is to use a spatial filter that removes high-frequency data. In this approach, only a single frame is evaluated at any given time, and parts of the image that are one or two pixels

in size are nearly eliminated. This does remove the noise, but it also degrades the image quality because there is no way to differentiate between noise and detail. This approach can also cause an artificial appearance in which people look like their skin is made of plastic. This represents the most widely used noise-reduction approach.

A temporal filter takes advantage of the fact that noise is a random element of the image that changes over time. Instead of simply evaluating individual frames, a temporal noise filter evaluates several frames at once. By identifying the differences between two frames and then removing that data from the final image, visible noise can be reduced very effectively. If there are no objects in motion, this is a virtually perfect noise-reduction technique that preserves most of the details. This approach is used by many high-end competitors. However, a problem arises if there are moving objects in the image, which also cause differences from one frame to the next; of course, these differences should be retained. If moving objects are not distinguished from noise, a ghosting or smearing effect is seen.

The Noise Reduction (NR) module processing uses a per-pixel motion-adaptive and noise-adaptive temporal filter to avoid the artificial appearance and artifacts associated with conventional noise filters. To preserve maximal details, moving pixels do not undergo unnecessary noise processing. In static areas, the strength of noise reduction is determined on a per-pixel basis, depending on the level of noise in the surrounding pixels as well as in previous frames, allowing the filter to adapt to the amount of noise in the image at any given time. The final result is a natural-looking picture with minimal noise and grain, and maximal preservation of fine details.

6.34.2 Features

- Mosquito Noise Reduction (MNR)
- Block Noise Reduction (BNR)
- Spatial Noise Reduction (SNR)
- Temporal Based Frame Noise Reduction (TNR)
- Standalone mode (DRAM in DRAM out)
- Non-standalone mode

6.34.3 NR Block Diagram

The NR block diagram is shown in Figure 6-157.

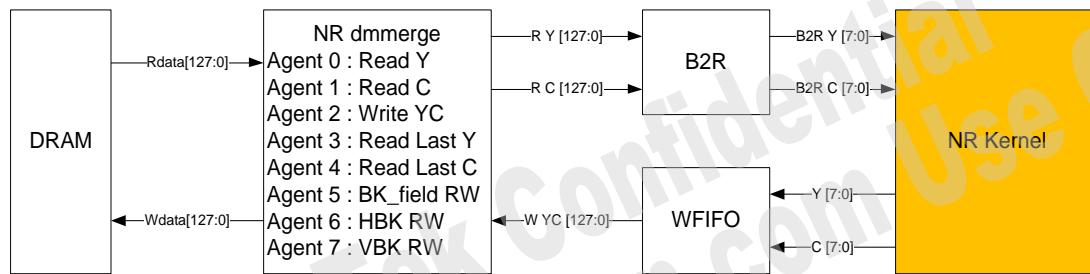


Figure 6-157 NR Block Diagram

6.34.4 Register Definition

For register details, please refer to Chapter 4.10.24 of “MT2712 IVI Application Processor Registers”.

6.34.5 Programming Guide

6.34.5.1 Standalone Operation Mode Configuration Flow

1. Configure NR DRAM read agent
2. Configure NR DRAM write agent
3. Configure NR misc control
4. Configure NR processing parameter
5. Initiate NR processing
6. Wait for NR interrupt assertion
7. Allocate the NR DRAM frame buffer
8. Set the format of NR DRAM frame buffer

6.34.5.2 Picture Size Configuration

In standalone operation mode, NR modules can process pictures with different sizes.

Related registers:

- 0x1501301c bit[9:0] HD_HDEW: horizontal active size in 2 pixel units.
- 0x1501301c bit[30:20] HD_VDEW: vertical active size.

Configuration value of HD_VDEW:

- In frame mode, the frame vertical active size is used to configure HD_VDEW.
- In field mode, the field vertical active size is used to configure HD_VDEW.

Take SD picture as an example. The active size of a SD picture is 720 x 480. As for frame mode, HD_VDEW should be configured as 480. As for field mode, HD_VDEW should be configured as 240.

6.34.5.3 Address Swap Mode

The decoded video data can be stored in DRAM in a different format which is called address swap mode. The NR module can support all of these data format variations in NR standalone operation mode. The correct register settings for different address swap modes are shown in Table 6-46 and Table 6-47.

Table 6-46 Related Register Fields for Frame Buffer Address Swap Control

Address	Bit Fields	Description
0x1501302C	[22:20]	HD_ADDR_SWAP
0x15013400	[6: 4]	WFIFO_ADDR_SWAP

Table 6-47 Corresponding Register Fields Configuration Values for Address Swap Mode

Address Swap Mode	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
HD_ADDR_SWAP	3'b000	3'b001	3'b010	3'b011	3'b100	3'b101	3'b110

6.34.5.4 DRAM Burst Read

The DRAM controller supports DRAM burst read function. Up to 64-byte data can be fetched from an external DRAM in one DRAM protocol command. Enabling this function can help to increase DRAM bandwidth utilization. The NR module supports the DRAM burst read function under certain constraint.

- 0x1501302C[23]: HD_BURST_READ_EN, configured as 1
- 0x1501302C[22:20]: HD_ADDR_SWAP, configured as 3'b010
- 0x15013028[6:0]: HD_LINE_OFFSET

Note: Only under the specified address swap mode, the DRAM burst read function can be enabled.

6.34.5.5 Non-Standalone Operation Mode Configuration Flow

1. Generate NR video timing
2. Switch NR operation mode
3. Switch NR module's clock source
4. Configure NR module control:
 - NR video input source format is 4:2:2
 - NR video input source is from VDO
 - Disable Bypass mode
 - Enable/Disable NR processing

6.34.5.6 Clock Source Switch

When the NR module operates in the non-standalone mode, the clock source for NR module must be switched to DI clock. 0x15010018[0]=1 means NR clock is the clock of DI.

When the NR module operates in the standalone mode, the clock source for NR module must be selected by 0x10000090[26:24].

6.34.5.7 Video Source Switch

When the NR module operates in the non-standalone mode, the video input source must be switched to DI. If 0x15014000[28] NR_SRC_SEL = 0, select DI's output as NR processing video source.

6.34.5.8 Display Formatter (DISPFMT_DI) Timing Configuration

Timing control signals for NR are generated by DISPFMT_DI. These timing control signals define the window of active video of DI's output. NR will only process the pixels in the active window.

6.34.5.9 NR Checksum

The NR module provides checksum capability to help verify the system correctness and reduce debugging effort.

6.34.5.10 DRAM Clock Domain Checksum

The DRAM clock domain checksum will return a 128-bit checksum value which is the checksum of the DRAM read/write data of whole frame. The Luma component and Chroma component checksums are separated.

To enable DRAM clock domain checksum, users only have to set checksum enable switch as 0'b1, and configure the shift mode. The checksum value will be available in the status register after the NR operation is done.

6.35 TV Decoder

6.35.1 Introduction

The TV Decoder (TVD) is a multi-standard TV signal decoder.

6.35.2 Features

The module supports the following features:

- Accept NTSC, PAL (B, D, G, H, I, M, N, Nc), SECAM (B, D, G, K, K1, L), NTSC-4.43, and PAL-60 composite video
- One fully differential CMOS analog preprocessing channel with clamping and Automatic Gain Control (AGC) for best S/N performance
- High-speed oversampling 10-bit A/D converter
- Capabilities for locking weak, noisy, or unstable signals
- Single 27-MHz reference clock for all standards
- No need of line-locked clock source
- Automatic detection and switching among NTSC, PAL and SECAM standards
- 2D/3D comb filter

6.35.3 Block Diagram

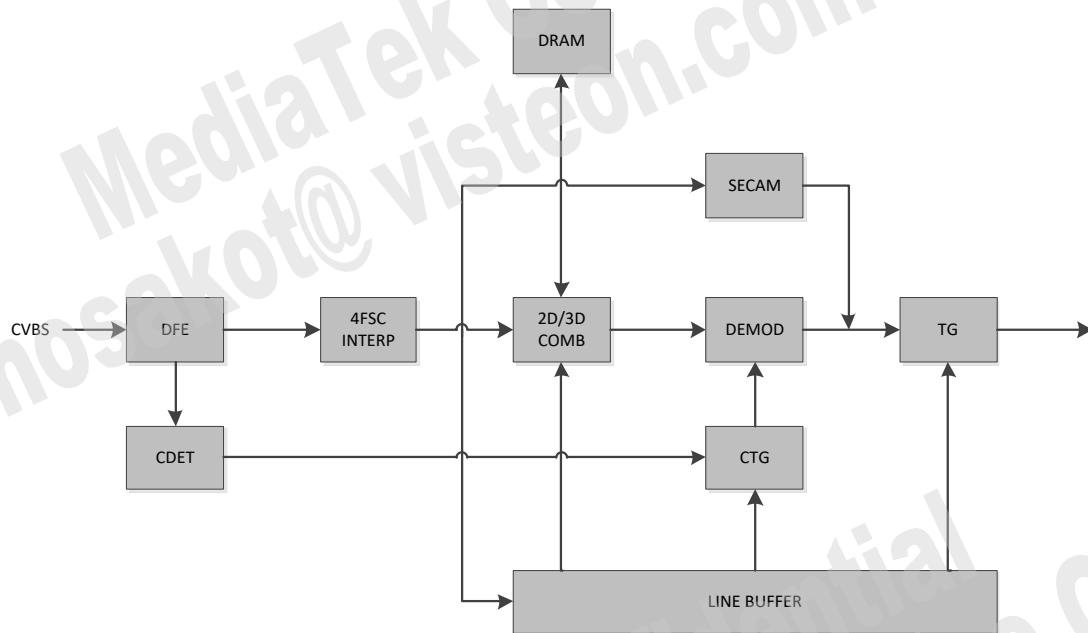


Figure 6-158 TVD Block Diagram

6.35.4 Programming Guide

6.35.4.1 Normal Flow

- Program CVBS ADC register
- Enable clock and reset
- Check TVD VPRES status
- Check mode status, set different settings for PAL/NTSC/SECAM
- Tune color process setting

6.35.4.2 TVD Reset

TVD SW reset register bit is 0x15010130[13], when 0x15010130[13]==1'b0, all TVD logics will be reset.

6.35.4.3 TVD Interrupt

SW can read TVD interrupt status from 0x15010000[6], then clears the status by 0x15010004[6].

TVD interrupt has two sub-interrupts as mentioned below:

- INTR_VPRES_TVD: TVD3D video on/off switching, write 1 to clear 0x15015000[0]: INTR_VPRES_TVD status register bit 0x1501540c[0]: Interrupt mask for INTR_VPRES_TVD

- INTR_MODE_TVD: TVD3D mode switching, write 1 to clear 0x15015000[1]: INTR_MODE_TVD status register bit 0x1501540c[1]: Interrupt mask for INTR_MODE_TVD

6.35.4.4 Debug Mode

- 0x150155ac[18]: Enable TVD internal pattern generator for testing
- 0x150155a0[27]: Enable Gen-lock debugging mode

6.35.4.5 Some Useful Status

- 0x15015080[12]: VPRES status
- 0x15015080[30:28]: Mode status
- 0x15015080[31]: Color kill status
- 0x15015044[25:16]: Sync level
- 0x15015044[9:0]: Blank level
- 0x1501509c[8]: Burst lock status
- 0x15015088[3]: V lock status
- 0x15015088[1]: H lock status

6.35.5 Register Definition

For register details, please refer to Chapter 4.10.25 of “MT2712 IVI Application Processor Registers”.

6.36 DISP_MONITOR

6.36.1 Introduction

The Display Monitor (DISP_MONITOR) is mounted at Digital Serial Interface (DSI)/Low-Voltage Differential Signaling (LVDS) input port. The IP monitors DSI/LVDS input data inside the interesting region and computes the Cyclic Redundancy Check (CRC). The CRC is then compared with the golden value. The comparison result indicates whether this display’s content is identical with expected.

The system supports four independent Display Monitors; each is configurable to monitor any display path.

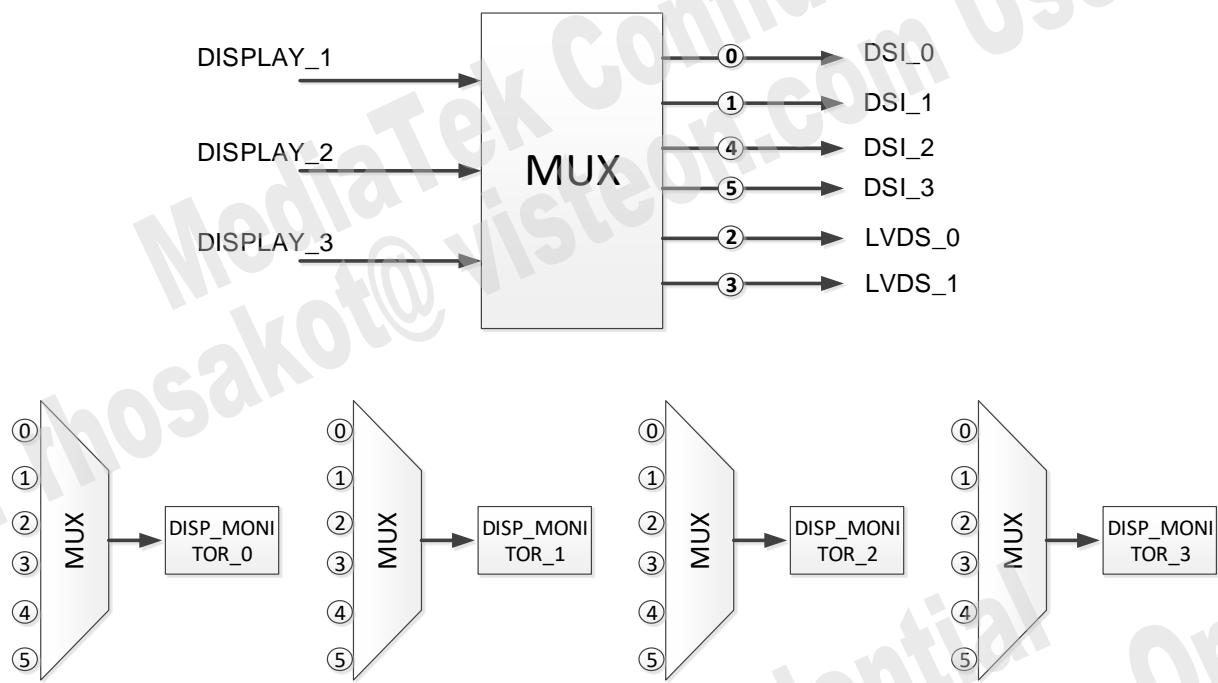


Figure 6-159 DISP_MONITOR Overview

6.36.2 Features

DISP_MONITOR supports the following features:

- 24-bit pixel data input
- CRC calculation of max. 12 independent regions
- All the regions are programmable for location and size

6.36.3 Block Diagram

Figure 6-160 depicts the block diagram of DISP_MONITOR.

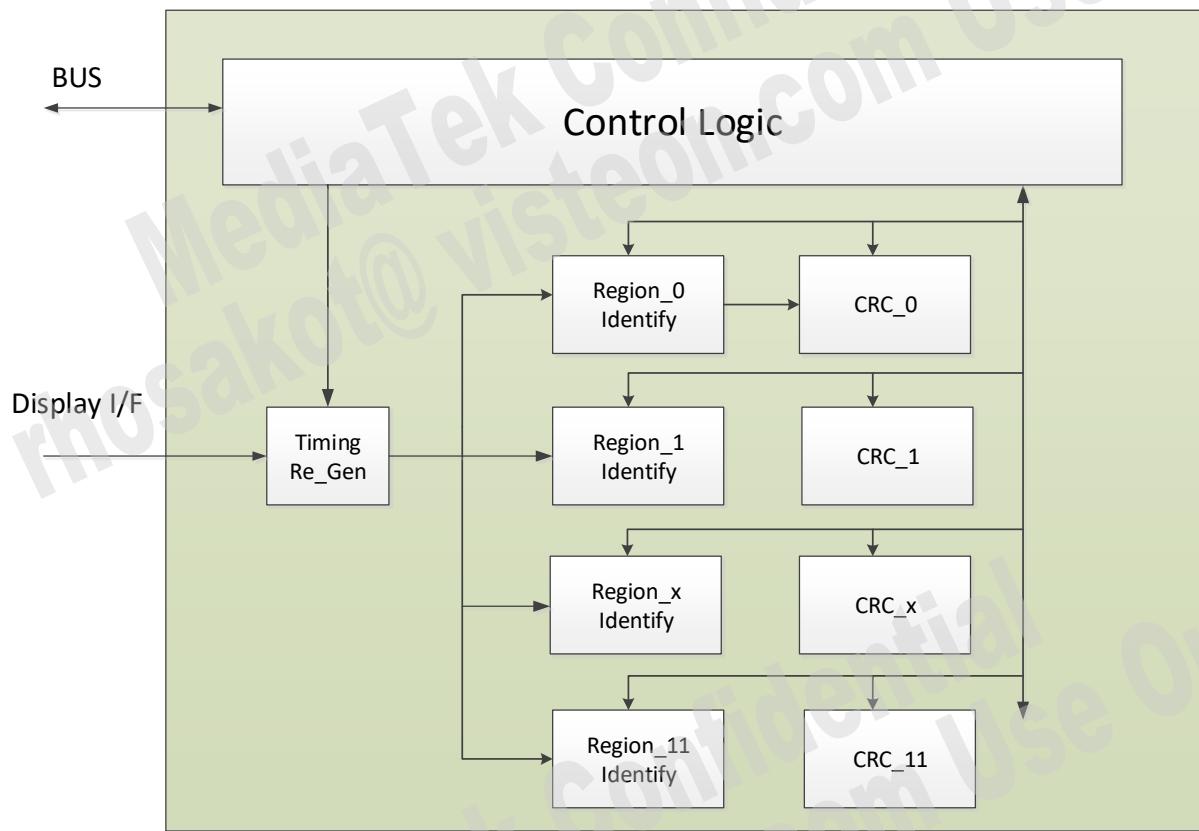


Figure 6-160 DISP_MONITOR Function Blocks

Timing Re_gen: Generate the timing information (such as Horizontal/Vertical (H/V) counter) according to the input data and resolution.

Region_x Identify ("x" indicates any value from 0 to 11): Determine whether the coming data is inside the region.

CRC_x ("x" indicates any value from 0 to 11): Compute the CRC value for active data (inside the region).

Control Logic: Control the IP behaviors, such as configuring the parameter, recording the status, enabling or disabling monitor, controlling interrupt, etc.

6.36.4 Register Definition

For register details, please refer to Chapter 4.9.26 of "MT2712 IVI Application Processor Registers".

6.36.5 Programming Guide

6.36.5.1 Programming Flow

Following shows the basic flow to configure one monitor:

Step 1: Disable clock gate and release reset.

- Step 2: Select source engine.
 Step 3: Configure source size.
 Step 4: Configure region parameters.
 Step 5: Enable region.
 Step 6: Configure Interrupt Request (IRQ).
 Step 7: Enable engine.
 Step 8: IRQ handle or poll the ready status.
 Step 9: Read the region's CRC and go on ...

6.36.6 IRQ Guide

Figure 6-161 shows the IRQ status distribution of DISP_MONITOR.

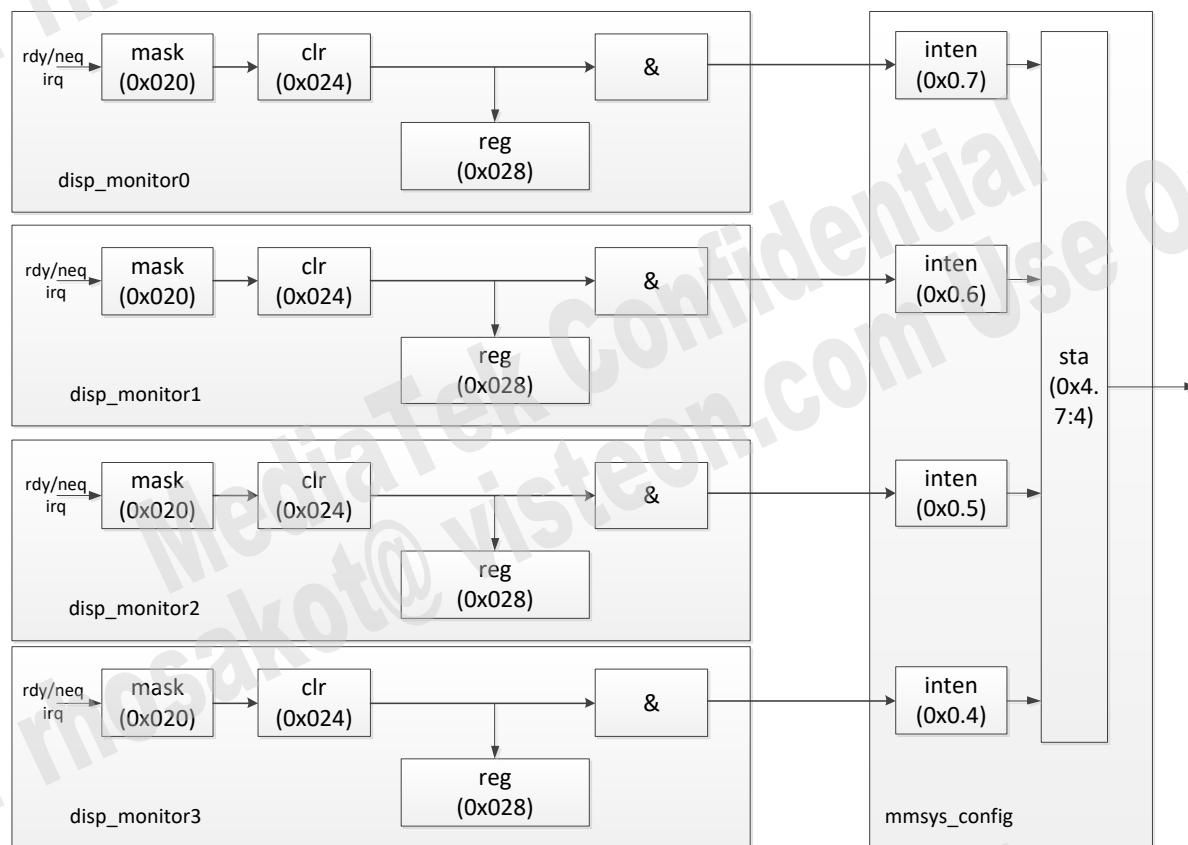


Figure 6-161 IRQ Status

Following is an example handling flow once IRQ has happened:

1. Find the IRQ source engine (DISP_MONITOR 0/1/2/3).
2. Find the sub IRQ source engine (region x in DISP_MONITOR).
3. Clear the sub IRQ status and go to Interrupt Service Request (ISR).
4. Clear the top IRQ status.
5. Exit the IRQ handle routine.

Document Revision History

Version	Date	Author	Description
2.01	2018-02-01	Dan Liu	<ul style="list-style-type: none"> • Change Peak DMIPS number of 12K to 9.9K • Change 2712M little Cores A35 of 3x 1.2GHz to 2x 1.2GHz • Delete 2712T • Update 1x eMMC 5.1
2.02	2018-02-23	Dan Liu	<ul style="list-style-type: none"> • Update section 2.6.1 Package Outlines
2.03	2018-05-23	Dan Liu	<ul style="list-style-type: none"> • Update Figure 2-6 Power On/Off Sequence with XTAL • Correct typo of color depth of RGB8888 in section 1.3.4 • Update Figure 6-9 Waveform of CASE2 Format in I2S-IQ Interface • Update Figure 6-10 Waveform of CASE3 Format in I2S-IQ Interface • Update USB BC1.2 spec • Update Display spec
2.04	2018-06-11	Dan Liu	<ul style="list-style-type: none"> • Update Table 2-13 XTAL Specifications • Update section 5.14.2: Features • Update Figure 5-67 Output Waveform Timing Programmable Parameters
2.05	2018-07-24	Dan Liu	<ul style="list-style-type: none"> • Update Figure 2-8 Power On/Off Sequence with XTAL
2.06	2018-9-03	Dan Liu	<ul style="list-style-type: none"> • Update section 5.5.2 Features • Update section 5.5.3 Block Diagram • Update section 5.5.7 Register Map
2.07	2018-10-15	Dan Liu	<ul style="list-style-type: none"> • Update Table 1-1 Scalable Platform Solution • Update chapter 1.3.2 Memory
2.08	2018-11-19	Garlic Tseng	<ul style="list-style-type: none"> • Remove section 1.3.5 audio codec software part
2.09	2018-12-20	Sophia	<ul style="list-style-type: none"> • Move Table 1-1 Scalable Platform Solution to section 2.7 • Update Figure 2-10 Top Mark of MT2712
2.10	2019-01-08	Sophia	<ul style="list-style-type: none"> • Modify table format of section 2.2.4 • Add section heading 2.2.4.3 1.8 IO
2.11	2019-02-12	Sophia	<ul style="list-style-type: none"> • Update section 2.5.2 Analog Block Features • Add section 2.5.7 AACD
2.12	2019-03-12	Sophia	<ul style="list-style-type: none"> • Fix a typo for VCC18IO_RTC digital power in section 2.1.3 Detailed Pin Description • Update Table 2-5 Absolute Maximum Ratings • Modify the caption of Table 2-5 Absolute Maximum Ratings • Update Table 2-34 Scalable Platform Solution

Version	Date	Author	Description
			<ul style="list-style-type: none"> Update Table 5-13 MT2712 Pins Reset Status & Driving: separate 18OD33 IO driving strength to 1.8v mode and 3.3V mode, add the 3.3V mode driving guideline Update section 5.2.6.4 Driving Strength Select: add 18OD33 IO driving strength application note Add description in section 5.9.1 Introduction and section 5.9.2 Features Update Figure 5-50 SPI Slave Block Diagram
2.13	2019-05-05	Sophia	<ul style="list-style-type: none"> Correct MSDC IO driving typo in section 5.2 GPIO Controller Add the features "Support mode A/mode B sync period 32/64 bck cycle" and "I2S/EIAJ format LRCK width support 32/64 bck cycle" to PCM part in section 6.1.2 Features Update the Rpu and Rpd in MSDC 1.8 IO table in section 2.2.4.5 Update the note of MSDC 1.8 IO table in section 2.2.4.5 Update MT2712P operation voltage specification in and add the warning to Table 2-6 Recommended Operating Conditions for Power Supply Correct the typos in Table 2-5 Absolute Maximum Ratings and Table 2-6 Recommended Operating Conditions for Power Supply: DDRV_BRDDR → DDRV_ARDDR
2.14	2019-06-04	Sophia	<ul style="list-style-type: none"> Update XO input AC specifications in Table 3-1 XO Specifications Update Table 6-37 DC/AC Parameters Add the Abbreviations
2.15	2019-07-01	Sophia	<ul style="list-style-type: none"> BT601/656 interface for internal use only
2.16	2019-09-20	Sophia	<ul style="list-style-type: none"> Add Section 6.1.5 Audio Clock Structure Add Section 6.1.8 Programming Guide Add Table 3-26 RTC Clock Specification and Section 3.7.5 Programming Guide Modify legacy Interface -> ONFI 1.0 interface in Section 5.12 NAND Flash Interface Change "NAND Flash Controller" into "NFI" in Section 5.12 NAND Flash Interface Add "NDQS" PIN to Section 5.12.3 Block Diagram Add 5.12.6 Programming Guide Replace "w RS" with "Repeated Start" in Section 5.15.6 Transfer Format Support Add note to Section 5.15.8 Programming Guide Add Section 5.16.6 Clock Relationship Add Section 5.16.8 Programming Guide Revise the pin numbers of UART ports in Section 5.5.2 Features

Version	Date	Author	Description
			<ul style="list-style-type: none">• Add "Step 16" to Table 5-36 USB Host Initialization Flow• Add clock domain information to Figure 5-15 USB Host Architecture• Add Section 6.13.5 Programming Guide• Change "Support DISM 8bit mode" to "Support DISM 8-bit and 6-bit mode" in Section 6.27.2 Features• Revise the description mistake in the caption of Figure 6-142 Dual Link 8-bit DISM Standard• Add Section 6.27.8 Chanel Swap• Add Section 5.1.7 Memory Map for 4G DRAM Support• Fix connection error in Figure 4-10 Infra Bus Fabric• Keep dither function in and remove OD descriptions from Section 6.23 Display OD Engine• Remove compression description from Section 6.22 UFOe Processor• Remove PAD related functional description from Section 6.24 Digital Parallel Interface• Remove UFO compression description from Section 6.25.2 Features and Section 6.25.3 Block Diagram• Add operation frequency range and IO voltage to Section 6.28.2 Features• List the registers in Section 6.18.5 Programming Guide• Add Section 6.19.5 Programming Guide
2.16	2019-09-20	Sophia	<ul style="list-style-type: none">• Remove redundant symbols at the top of diagram in Figure 6-47 DISP Functional Blocks of Multimedia Partition• Remove connections from DISP FMT to NR and from NR to WR_CHANNEL in Figure 6-48 IMGSYS_TOP Functional Blocks• Update the programming guide and waveform of mutex in Section 6.10.5 Programming Guide• Add Figure 6-53 DISP_MONITOR Clock Diagram and its note• Add Section 4.4.5 Programming Guide• Update the frequencies in Table 4-6 Clock Frequency of Major AP System Bus Fabric• Add Section 4.2.8 Programming Guide• Add description to Section 4.3.5 Programming Guide• Remove unused circuit from Figure 6-45 IMG_RSZ Block Diagram• Correct MSDC feature description in Section 5.11.2 Features• Correct MSDC programming sequence in Section 5.11.6 Programming Guide

Version	Date	Author	Description
			<ul style="list-style-type: none"> Replace "Memory Stick and SD Card Controller" with "MMC and SD Controller" in Section 5.11 MMC and SD Controller Add APXGPT feature description in Section 5.18.2 Features and Section 5.19.2 Features Add Section 5.18.5 Programming Guide and Section 5.19.5 Programming Guide Change 8*8 key matrix into 7*7 key matrix in Section 5.4.3 Block Diagram Add Section 5.21.6 Programming Guide Add Section 5.17.5 Programming Guide Change "example of 8-bit conversion" into "example of 12-bit conversion" in Section 5.14.5 Theory of Operations Remove unrelated description "Figure 5-79 is the architecture of SYS_CIRQ. SYS_CIRQ_REG stores the mask/sensitivity/polarity attributes of each interrupt signals and SYS_CIRQ_CON is used to mask and detect edge-triggered interrupts" from Section 5.14.5 Theory of Operations
2.16	2019-09-20	Sophia	<ul style="list-style-type: none"> Fix the typo: Change "CA53" into "CA35" in Section 4.1.1 Introduction and 4.1.2.1 Big Little Cluster Unit Specifications Improve the figure appearance of Figure 4-1 MCUSYS Block Diagram and Figure 4-2 Power Domain Improve the table appearance of Table 4-2 Power State Table Remove the unused GIC ID 48~51, 56~59, 64~67, 72~75 and 413~428 in Table 4-1 Interrupt Request List for MCUSYS Add Section 6.35.4 Programming Guide Add Section 6.5.5 Programming Guide Add clock domain information to Figure 6-33 WR_CHANNEL Block Diagram Correct Simple SMI to EMI System bus Architecture in Figure 6-149 Simple System Architecture → There are 2 MM_IOMMU groups Update power domain information of Graphics Processing Unit(GPU) → Change the MTCMOS domain number from one to four in GPU in Table 3-24 VCORE MTCMOS Domains Update Figure 3-13 Block Diagram of MT2712 Power Domain to change the MTCMOS domain number from one to four inside GPU

Version	Date	Author	Description
2.16	2019-09-20	Sophia	<ul style="list-style-type: none"> • Remove blank rows and columns from Table 3-25 Power Mode Scenario • Change "MVDO" into "VDO" in Figure 6-153 DISPMT Block Diagram • Change "MVDO" into "VDO" in Figure 6-152 Internal Block Diagram of VDO • Add "Non-standalone mode" to Section 6.34.2 Features • Add 6.34.5 Programming Guide • Change "{fhctl2_pll_dds[20:0],10'b0}" into "{fhctl3_pll_dds[20:0],10'b0}" in Figure 3-11 Hopping Enable Control • Add Section 5.22.5 Clocks • Add Section 5.22.7 Programming Guide • Add Section 4.9 CM4SYS <ul style="list-style-type: none"> • Add "-" to the blank cells in tables • Update clock description in figures in Section 3.3.3.3 Clock PLL • Update clock monitor information in Figure 3-8 ABIST FMETER Structure • Update clock monitor information in Figure 3-9 CKGEN Structure • Remove redundant description from Table 3-19 Clock Gating Settings • Update frequency meter specification description in Section 3.3.5.3.1 Frequency Meter • Update hf_fa1sys_hp_ck and hf_fa2sys_hp_ck clock mux in Table 3-21 Clock Mux Table • Update AC specification in Table 3-1 XO Specifications • Add "WDT_DFD_CT" to Section 3.5.5.1 TOPRGU Initial • Change "dramc_sref_sta" to "ddr_sref_sta" in Section 3.5.5.6 DDR Reserved Mode Reset • Normalize the content in Section 4.7.1 Introduction • Normalize the content in Section 4.7.2 Features • Normalize the content in Section 4.8.1 Introduction • Normalize the content in Section 4.8.2 Features • Add VDEC operation clock description to Section 6.7.5 Block Diagram • Add the description of DPHY and CSI2 version to Section 6.4.2 Features • Add the description of camsv number to Section 6.4.3 Block Diagram • Modify the sequence number of camsv in Figure 6-30 CSI Block Diagram and Figure 6-31 CSI CLK Domain

Version	Date	Author	Description
2.16	2019-09-20	Sophia	<ul style="list-style-type: none">• Add the CSI clock domain description to Section 6.4.3 Block Diagram• Add clock domain to Figure 5-104 Flashif Block Diagram• Add Section 5.13.6 Programming Guide• Add Figure 5-45 MT2712 PCIe Block Diagram to describe PCIe architecture in subsys• Add Section 5.8.6 Programming Guide• Modify IP name from SSUSB_PCIE_PHYD to SSUSB_PCIE_PHY in Section 5.9 SSUSB_PCIE_PHY• List USB SuperSpeed and PCI Express key features separately in Section 5.9.2 Features <ul style="list-style-type: none">• Add Section 5.9.7 Clock Relationship• Fix the typo in Figure 5-128 System Level Thermal Controller Block Diagram• Fix the typo in Figure 5-128 System Level Thermal Controller Block Diagram• Change symbol "B" into "IO" and add comments for IES&SMT&RDSEL&TDSEL in Table 5-22 GPIO Control Table• Remove IO type4 and type5 from Figure 5-4 IO Types-->these IO types are not supported in GPIO• Change "Termination when not used" into "Connection For Unused Pins" in Table 5-23 MT2712 Pins Reset Status & Driving• Remove Table 6-21 DPHY Global Operation Timing Parameter Defined by MIPI Specification• Add Section 6.27.4 LVDS AC Timing• Add Section 6.25.4 DSI AC Timing• Add Section 6.1.4 Audio AC Timing• Add Section 5.22.4 ETHER_QOS AC Timing• Add Section 5.21.4 IRRX AC Timing• Add Section 5.16.5 PWM AC Timing• Add Section 5.15.4 I2C AC Timing• Add Section 5.14.4 AUXADC AC Timing• Add Section 5.13.4 Flashif AC Timing• Add Section 5.12.4 NFI AC Timing• Add Section 5.11.4 MSDC AC Timing• Add Section 5.10.5 SPI AC Timing• Add Section 5.9.5 USB 3.0 High Speed Controller AC Timing• Add Section 5.9.4 USB 2.0 High Speed Controller AC Timing• Add Section 5.9.6 PCIe PHY AC Timing• Add Section 5.5.4 UART AC Timing• Add Section 5.4.4 Keypad AC Timing

Version	Date	Author	Description
			<ul style="list-style-type: none"> • Add Section 5.2.4 GPIO AC Timing • Add Section 4.2.5 JTAG AC Timing • Replace "PRZ" with "RSZ" in Section 6.10.5 Programming Guide • Change sub-block name from "PRZ" to "RSZ" in Figure 6-46 MDP Functional Blocks of Multimedia Partition
2.16	2019-09-20	Sophia	<ul style="list-style-type: none"> • Change "Dynamic Voltage & Frequency Scaling (DVFS)" to "Dynamic Frequency Scaling (DFS)" and change "run at different frequency/voltage configurations" to "run at different frequency configurations" in Section 4.1.1 Introduction • Change "DVFS (Dynamic Voltage & Frequency Scaling)" to "DFS (Dynamic Frequency Scaling)" in Abbreviation • Update description of VA10 in Table 2-34 Temperature Sensor Specifications • Update description of VA10 and VA105 in Table 2-13 Signals Details • Add "Speed up to 2800 MHz" to Section 1.3.2 Memory • Update Table 2-37 Scalable Platform Solution • Modify the parameters and description of Table 2-11 • Remove eMMC pre-Mode description from Section 2.2.4.5 MSDC IO • Remove SD Card/GPIO/Sleep mode description from Table 2-9 General 1.8V IO DC Timing Parameters • Remove Operating Temperature description from Table 2-9 General 1.8V IO DC Timing Parameters and Table 2-10 General 3.3V IO DC Timing Parameters • Change "MT2712P only" to "MT2712P/S/H only" and change "None MT2712P" to "None MT2712P/S/H" in Table 2-6 Recommended Operating Conditions for Power Supply • Modify Figure 2-13 Top Mark of MT2712 • Change "VCORE/VGPU/VA10" into "VCCK"; change "VPROC" into "VCPU", and change "VMC/VMCH" into "VMC(Memory Card)" in Table 2-15 Signals Details • Update Figure 2-8 Power-On/Off Sequence with XTAL • Change "Vcc3IO_SPICT" to "VCC33IO_SPICTP" in Table 2-1 Pin Coordinate Using LPDDR4; Change "Vcc3IO_GPIO" to "VCC33IO_GPIO" and "Vcc3IO_GBE" to "VCC33IO_GBE" in Table 2-4 Detailed Pin List • Change "SDIO3.0. Up to 5 Hz update rate" to "SDIO3.0. Support SDR104 (Bus clock rate 208 MHz)" in Section 1.5.2 Companion Components
2.16	2019-09-20	Sophia	<ul style="list-style-type: none"> • Update Figure 3-10 FHCTL Block Diagram

Version	Date	Author	Description
			<ul style="list-style-type: none"> Change “DVFS/SSC/Dynamic SSC” to “DFS/SSC/Dynamic SSC” in Section 3.4.5.1 Initialize FHCTL for Each PLL and change “SSC/DVFS/Dynamic SSC” to “SSC/DFS/Dynamic SSC” in Section 3.4.5.2 Enable/Disable Hopping Change “DVFS technology” to “DFS technology” in Section 4.2.6 Application Processor Debug Subsystem
2.17	2019-11-01	Dan Liu	<ul style="list-style-type: none"> Add “internal code” to Figure 2-13 Top Mark of MT2712 Modify Figure 6-47 DISP Functional Blocks of Multimedia Partition
2.18	2020-06-10	Dan Liu	<ul style="list-style-type: none"> Modify Figure 2-5 Power-On/Off Sequence with XTAL Remove Figure 2-6 Power Measure Reference Point Add Section 2.5.8 CVBS Remove “Randomizer (TOSHIBA/SAMSUNG)” in Section 5.12 NAND Flash Interface Remove Section 1.4.2 Operating Conditions Modify Table 2-20 ARMCA72PLL Specifications Modify “Quad-core ARM® Cortex-CA35 MPCoreTM operating at 1.2 GHz” to “Quad-core ARM® Cortex-CA35 MPCoreTM operating up to 1.2 GHz” and “Dual-core ARM® Cortex-CA72 MPCoreTM operating at 1.4 GHz, 1M L2 with ECC” to “Dual-core ARM® Cortex-CA72 MPCoreTM operating up to 1.4 GHz, 1MB L2 with ECC” in Section 4.1.2.1 Big Little Cluster Unit Modify Table 2-38 Thermal Operating Specifications Add Figure 2-9 Power Measure Reference Point Modify the descriptions of IO Driving in Table 5-23 MT2712 Pins Reset Status & Driving Modify the symbol and notes in Table 5-45 SPI Master Electrical Specifications Remove “DDR4” in Chapter 1 System Overview Modify “Figure 2-1 Ball Map View from DDR4” to “Figure 2-1 Ball Map View from LPDDR4” Remove Table 2-2 Pin Coordinate Using DDR4 Remove DDR4 information in Table 2-4 Absolute Maximum Ratings and Table 2-5 Recommended Operating Conditions for Power Supply Remove Section 2.1.4.2 External Memory Interface for DDR4 Modify Table 6-2 I2S AC Timing Characristics Modify Table 6-3 TDM AC Timing Characristics Modify Figure 6-21 TDM Mode Timing Diagram Modify Table 6-4 PCM AC Timing Characristics Modify Figure 6-22 PCM Mode Timing Diagram Modify Table 6-5 I2S-IQ AC Timing Characristics

Version	Date	Author	Description
			<ul style="list-style-type: none"> • Modify Figure 6-23 I2S-IQ Slave Mode Timing Diagram • Modify Figure 6-135 LVDS Encoder Block Diagram • Add Section 6.27.4 LVDS Clock • Modify Figure 6-140 Single Link 6-bit VESA Standard • Modify Figure 6-141 Single Link 8-bit DISM Standard • Modify Figure 6-142 Single Link 8-bit VESA Standard • Modify Table 6-44 DC/AC Parameters • Remove Section 6.27.7 Clock Relationship • Modify Section 6.27.9 Programing Guide • Modify Figure 6-2 TDM Protocol • Modify Table 2-10 VCC18IO for MSDC0E/MSDC3 DC Timing Parameters • Add Table 2-12 MSDC Overshoot/Undershoot • Modify Table 5-49 eMMC5.0 Timing Parameter (high-speed mode) • Add note in Table 5-53 eMMC5.0 Output Parameter (high-speed 400 mode host input)
2.19	2021-03-16	Lapm Li Guoyang Hu Yajun Du Dongxue Zhang Yahua Cao	<ul style="list-style-type: none"> • Update Table 5-48: Add host internal sample clock rising edge timing requirement • Update Table 5-49: Add host internal sample clock rising edge timing requirement • Update Table 5-54: Add host internal sample clock rising edge timing requirement • Update Table 5-55: Add host internal sample clock rising edge timing requirement • Update Table 5-56: Add host internal sample clock rising edge timing requirement • Update Table 2-5 “Recommended Operating Conditions for Power Supply” • Add VCCK Suspend mode A voltage value to Table 2-5 Recommended Operating Conditions for Power Supply • Add 2712A segment information to Section 2.2 Electrical Characteristics and Section 2.7 Ordering Information • Remove 2712P spec from Chapter 1 System Overview, Section 1.1 General Description, Section 1.3 Key Features, Section 2.2 Electrical Characteristics and Section 2.7 Ordering Information • Modify the maximum working frequency of CA72 from 1.6 to 1.5 GHz in Section 3.6.2 Features
2.20	2021-10-09	WH Chou Chih-Wen Yang	<ul style="list-style-type: none"> • Modified DA_HS_EXIT in Figure 6-120 Registers for Data Lane Timing Parameters, Figure 6-121 Registers for Clock Lane Timing Parameters • Fixed wrong position of VSE in Figure 6-124 Non-burst Transmission: Sync-pulse Mode

Version	Date	Author	Description
			<ul style="list-style-type: none">• Replaced Figure 6-125 Sync-pulse Mode Line Period, Figure 6-128 Sync-event Mode Line Period for better explanation• Fixed line time formula error in Figure 6-126 Sync-pulse Mode Word-count Parameters, Figure 6-129 Sync-event Mode Word-count Parameters• Update codec info

Exhibit 1 Terms and Conditions

Your access to and use of this document and the information contained herein (collectively this "Document") is subject to your (including the corporation or other legal entity you represent, collectively "You") acceptance of the terms and conditions set forth below ("T&C"). By using, accessing or downloading this Document, You are accepting the T&C and agree to be bound by the T&C. If You don't agree to the T&C, You may not use this Document and shall immediately destroy any copy thereof.

This Document contains information that is confidential and proprietary to MediaTek Inc. and/or its affiliates (collectively "MediaTek") or its licensors and is provided solely for Your internal use with MediaTek's chipset(s) described in this Document and shall not be used for any other purposes (including but not limited to identifying or providing evidence to support any potential patent infringement claim against MediaTek or any of MediaTek's suppliers and/or direct or indirect customers). Unauthorized use or disclosure of the information contained herein is prohibited. You agree to indemnify MediaTek for any loss or damages suffered by MediaTek for Your unauthorized use or disclosure of this Document, in whole or in part.

MediaTek and its licensors retain titles and all ownership rights in and to this Document and no license (express or implied, by estoppels or otherwise) to any intellectual property rights is granted hereunder. This Document is subject to change without further notification. MEDIATEK DOES NOT ASSUME ANY RESPONSIBILITY ARISING OUT OF OR IN CONNECTION WITH ANY USE OF, OR RELIANCE ON, THIS DOCUMENT, AND SPECIFICALLY DISCLAIMS ANY AND ALL LIABILITY, INCLUDING, WITHOUT LIMITATION, CONSEQUENTIAL OR INCIDENTAL DAMAGES.

THIS DOCUMENT AND ANY OTHER MATERIALS OR TECHNICAL SUPPORT PROVIDED BY MEDIATEK IN CONNECTION WITH THIS DOCUMENT, IF ANY, ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. MEDIATEK SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, COMPLETENESS OR ACCURACY AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MEDIATEK SHALL NOT BE RESPONSIBLE FOR ANY MEDIATEK DELIVERABLES MADE TO MEET YOUR SPECIFICATIONS OR TO CONFORM TO A PARTICULAR STANDARD OR OPEN FORUM.

Without limiting the generality of the foregoing, MEDIATEK MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES MEDIATEK ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT, CIRCUIT OR SOFTWARE. You agree that You are solely responsible for the designing, validating and testing Your product incorporating MediaTek's product and ensure such product meets applicable standards and any safety, security or other requirements.

The above T&C and all acts in connection with the T&C or this Document shall be governed, construed and interpreted in accordance with the laws of Taiwan, without giving effect to the principles of conflicts of law.