everyday genius

MT8516A Application Processor Functional Specification

Version:	0.9
Release date:	2018-01-02

Specifications are subject to change without notice.

Document Revision History

Revision	Date	Author	Description
0.1	2017-03-23	Lifang Wang	Initial draft
0.2	2017-06-08	Lifang Wang	Update the parameters
0.3	2017-06-12	Lifang Wang	Update the specifications
0.4	2017-09-06	Lifang Wang	Update the document’s format and 2.7.1 Reference Clock, delete table for Reference clock operation mode
0.5	2017-09-22	Lifang Wang	Add Figure 3-14 and Figure3-15, Change Chapter 3.7.5.2 to Chapter 3.7.5.1.1
0.6	2017-11-07	Lifang Wang	Modify Table 2-1 ball name WBT_EXT_G & GPS_DPX_RFOUT to NC. Table 2 15 WBT_EXT_G & GPS_DPX_RFOUT is NC ball, remove the description. Fix type error MLD4→ NLD4, APK→ SPK. Table 5-1 fix type error. Modify O*SQICS to SQICS. Modify B0*SQISO to SQISO, B0*SQISI to SQISI B0*I2S0_LRCK to I2S0_LRCK Remove which is not match with SOC feature.
0.7	2017-11-26	Lifang Wang	Restore the pin rename for MIPI, HDMI, Camera to avoid confuse. Update Figure 2-0 and table 2-15. Table 5-1,2,3 update.
0.8	2017-11-30	Lifang Wang	Modify table 2-45, BT LT receiver Out-of-band blocking max parameter to min
0.9	2018-01-02	Lifang Wang	fix figure 1-1 block diagram error

Table of Contents

List of Tables and Figures

1 System Overview

MT8516A is a highly integrated connected audio platform incorporating application processing and connectivity subsystems designed to enable connected audio applications. The chip integrates a Quad-core ARM® Cortex-A35 MPCore™ operating up to 1.3 GHz. The MT8516A interfaces to NAND flash memory, LPDDR2, LPDDR3, DDR3, DDR3L and DDR4 for optimal performance and also supports booting from eMMC to minimize the overall BOM cost. In addition, an extensive set of interfaces such as TDM/PDM inputs are included for microphone voice input control / search applications on connected audio products.

The application processor, a Quad-core ARM® Cortex-A35 MPCore™, includes a NEON multimedia processing engine.

MT8516A includes two wireless connectivity functions: WLAN and Bluetooth. These built-in RF parts of those two block scans support 802.11 b/g/n. With two advanced radio technologies integrated into a single chip, MT8516A provides the industry's best and most convenient connectivity solution. MT8516A implements advanced and sophisticated Radio Coexistence algorithms and hardware mechanisms. It also supports single antenna sharing among 2.4 GHz antenna for Bluetooth, WLAN. The enhanced overall quality is achieved for simultaneous voice, data, and audio transmission. The small footprint with low-power consumption greatly reduces the PCB layout resource.

1.1 Platform Features

- **AP MCU subsystem**
  - Quad-core ARM® Cortex-A35 MPCore™ operating at 1.3 GHz
  - NEON multimedia processing engine with SIMDv2 / VFPv4 ISA support
  - 32KB L1 I-cache and 32KB L1 D-cache
  - 512KB unified L2 cache
  - DVFS technology with adaptive operating voltage from 1.05V to 1.31V
- **Wireless Connectivity MCU subsystem**
  - Andes N9 processor with 48KB I-cache, 40KB D-cache
- **External memory interface**
  - Supports LPDDR2, LPDDR3, DDR3/L, DDR4 up to 2GB
  - 32-bit data bus width
  - Memory clock up to 800 MHz
  - Supports self-refresh/partial self-refresh mode
  - Low-power operation
  - Programmable slew rate for memory controller's IO pads
  - Supports dual rank memory device
  - Advanced bandwidth arbitration control
- **Security**
  - ARM® TrustZone® Security
- **Storage**
  - NAND flash controller supporting NAND bootable, iNAND2® and MoviNAND®
  - 3 sets of memory card controller supporting SD/SDHC/MMC and SDIO2.0/3.0 protocols
- **Connectivity**
  - Two USB ports, port0 support USB 2.0 OTG mode but port1 only support USB 2.0 host mode. The two USB20 ports support connection Hub to transfer data communications with HS/FS/LS Device. USB2.0 high-speed dual mode supporting 8 Tx and 8 Rx endpoints.
  - 3 UARTs for external devices and debugging interfaces
  - SPI master for external devices
  - 3 I2C to control peripheral devices, e.g. CMOS image sensor, or LCM module
  - I2S master output and master/slave input for connection with optional external hi-end audio codec
  - GPIOs
  - 10M/100M Ethernet MAC with MII and RMII interface
  - IR receiver
- **Operating conditions**
  - Core voltage: 1.15V
  - Processor DVFS+SRAM voltage : 1.15V~1.31V (Typ. 1.15V ; sleep mode 0.85V)
  - I/O voltage: 1.8V/2.8V/3.3V
  - Memory: 1.2V/1.35V/1.5V
  - NAND: 1.8V/3.3V
  - LCM interface: 1.8V/3.3V
  - Clock source: 26-MHz, 32.768-kHz
- **Package**
  - Type: WB TFBGA
  - 12.6mm x 13.1mm
  - Height: 1.2 mm maximum
  - Ball count: 406 balls
  - Ball pitch: 0.5mm

1.2 Multimedia Features

- **Audio**
  - I2S Master Mode sampling rates supported: 8kHz to 192kHz
  - I2S In Slave mode sampling rates supported: 8kHz to 48kHz
  - Sample formats supported: 16-bit/24-bit, Mono/Stereo
  - Interfaces supported: DAI, I2S, TDM, SPDIF
  - Flexible and powerful audio post-processing technologies
  - Supports DIR(SPDIF-input) decode: supports 32, 44.1, 48, 88.2, 96kHz sample rate.
  - Supports SPDIF-output encode: supports 32, 44.1, 48, 88.2, 96kHz sample rate.
  - Supports Time Division Multiplexer I2S output (master mode only): supports 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 88.2, 96, 192kHz sample rate and channel number up to 2/4/8 in configuration by 1/2/4 data pins (corresponding to 2/4/8 channels),
  - Dedicated pin for TDM TX (not share clock pins with TDM RX).
  - Supports Time Division Multiplexer input: supports 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 88.2, 96, 192kHz sample rate and channel number up to 2/4/8 in 1 serial data pin,
  - Dedicated pin for TDM RX (not share clock pins with TDM TX).

- **Speech**
  - Noise reduction
  - Noise suppression
  - Dual-MIC noise cancellation
  - Echo cancellation
  - Echo suppression
  - Dual-MIC input
  - Digital MIC input

1.3 Wi-Fi/Bluetooth Features

- **Supports integrated Wi-Fi/Bluetooth**
  - Supports single antenna for Bluetooth and WLAN
  - Self calibration
  - Best-in-class current consumption performance
  - Intelligent BT/WLAN coexistence scheme that goes beyond PTA signaling (for example, transmit window and duration that take into account protocol exchange sequence, frequency, etc.)
- **Wi-Fi**
  - Single-band (2.4GHz) single stream 802.11 b/g/n MAC/BB/RF
  - 802.11 d/h/k compliant
  - Security: WFA WPA/WPA2 personal, WPS2.0, WAPI (Hardware)
  - QoS: WFA WMM, WMM PS
  - Supports 802.11n optional features: STBC, A-MPDU, Blk-Ack, RIFS, MCS feedback, 20/40MHz coexistence (PCO), unscheduled PSMP
  - Supports 802.11w protected managed frames
  - Supports Wi-Fi Direct (WFA P-2-P standard)
  - Supports HotSpot 2.0 Passpoint
  - Per packet TX power control
- **Bluetooth**
  - Bluetooth specification v2.1+EDR
  - Bluetooth v4.0 Low Energy (LE)
  - Rx sensitivity: GFSK -95dBm, DQPSK -94dBm, 8-DPSK -88dBm
  - Best-in-class BT/Wi-Fi coexistence performance
  - Up to 4 piconets simultaneously with background inquiry/page scan
  - Supports Scatternet
  - Packet Loss Concealment (PLC) function for better voice quality
  - Low-power scan function to reduce power consumption in scan modes

1.4 General Description

The MediaTek MT8516A has integrated 802.11 b/g/n and Bluetooth 4.0 + HS radios and passive devices (IPD) to provide a connected audio solution. The application processor incorporates a high efficient 64-bit Quad Cortex-A35 processor operating at 1.3 GHz. Rich memory interfaces (PCDDR3, DDR4, LPDDR3, eMMC, Raw NAND) provide high flexibility to support variant memory configurations. The elaborate MMD (MediaTek Module Design) provides verified schematics and PCB layout for memory and power source to reduce development time. Combo chip MT6630, 802.11ac/BT also gives the alternative to fulfill high end Wi-Fi/BT requirement. The MT8516A processor delivers high-performance computing, low-power consumption, and good multimedia experience.

World-leading technology

Based on MediaTek's world-leading SoC architecture with advanced 28nm RF process, the MT8516A integrates digital and RF into a single chip that is suitable for compact PCB design. The PMIC MT6392 is designed to supply all the power to MT8516A itself. The two-chip solution brings lower rBOM and design effort to cost-effectively develop applications with fast time to market.

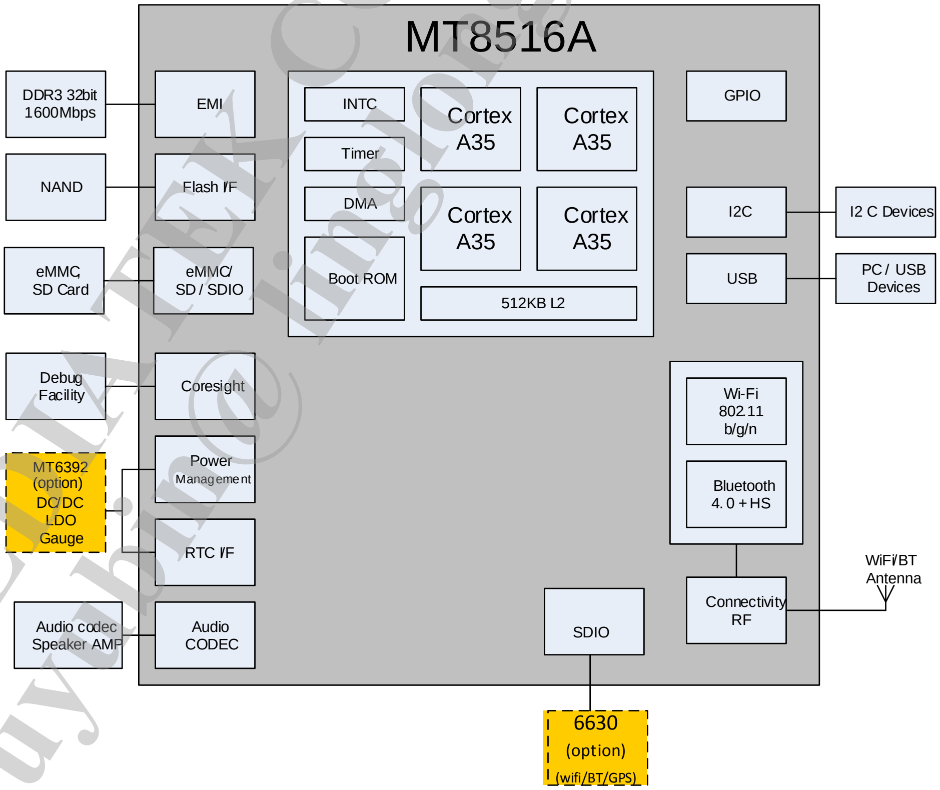
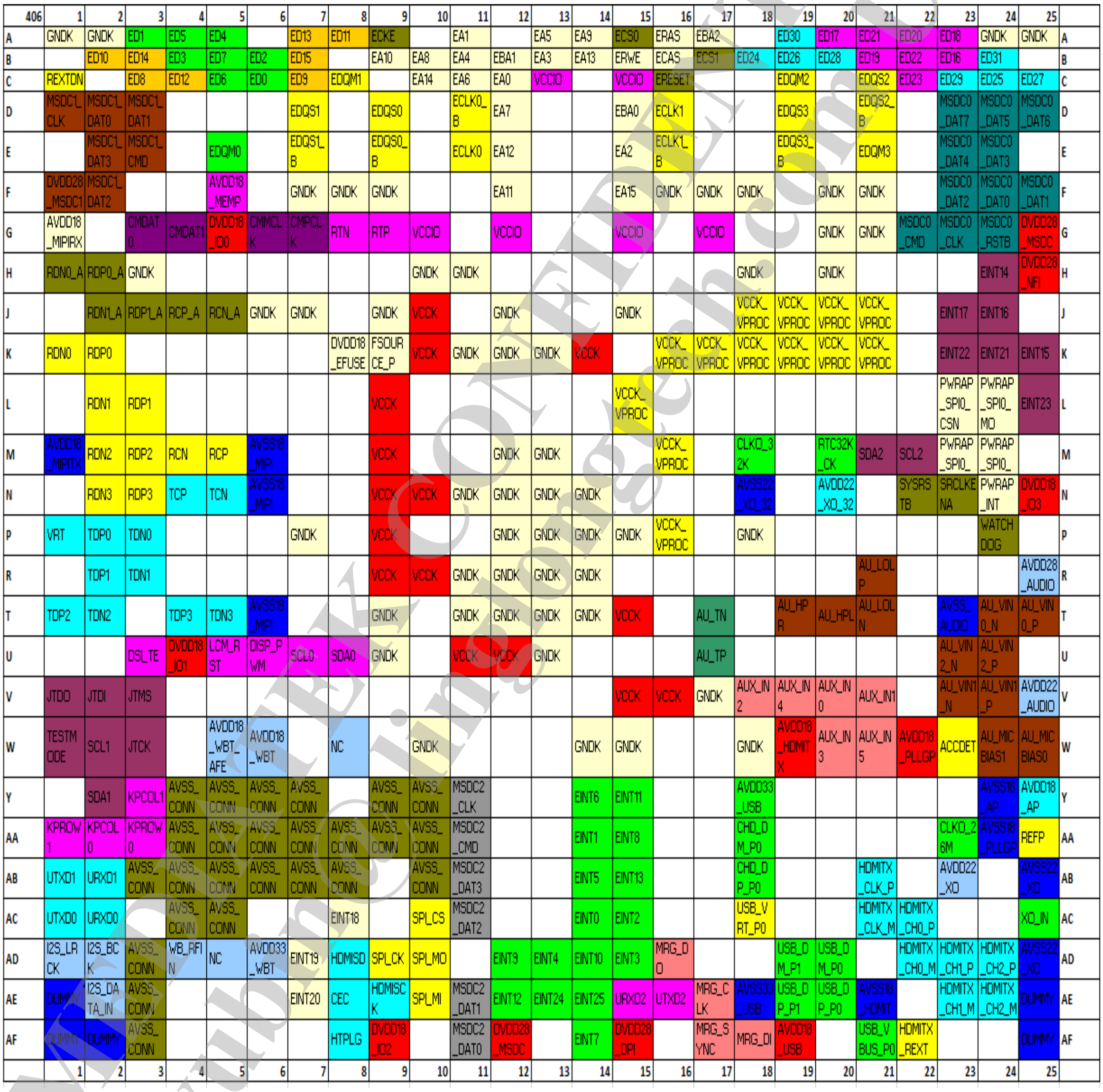


Figure 1-1: MT8516A Block Diagram

2 Product Description

2.1 Pin Description

2.1.1 Ball Map View



406	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	
A	GNDK	GNDK	ED1	ED5	ED4		ED13	ED11	ECKE		EA1		EA5	EA9	ECS0	ERAS	EBA2		ED30	ED17	ED21	ED20	ED18	GNDK	GNDK	A
B		ED10	ED14	ED3	ED7	ED2	ED15		EA10	EA8	EA4	EBA1	EA3	EA13	ERWE	ECAS	ECS1	ED24	ED26	ED28	ED19	ED22	ED16	ED31		B
C	REXTDN		ED8	ED12	ED6	ED0	ED9	EDQM1		EA14	EA6	EA0	VCCIO		VCCIO	ERESET			EDQM2		EDQS2	ED23	ED29	ED25	ED27	C
D	MSDC1_CLK	MSDC1_DAT0	MSDC1_DAT1				EDQS1		EDQS0		ECLK0_B	EA7			EBA0	ECLK1			EDQS3		EDQS2_B		MSDC0_DAT7	MSDC0_DAT5	MSDC0_DAT6	D
E		MSDC1_DAT3	MSDC1_CMD		EDQM0		EDQS1_B		EDQS0_B		ECLK0	EA12			EA2	ECLK1_B			EDQS3_B		EDQM3		MSDC0_DAT4	MSDC0_DAT3		E
F	DVDD28_MSDC1	MSDC1_DAT2			AVDD18_MEMP		GNDK	GNDK	GNDK			EA11			EA15	GNDK	GNDK	GNDK		GNDK	GNDK		MSDC0_DAT2	MSDC0_DAT0	MSDC0_DAT1	F
G	AVDD18_MIPIRX		CMDAT0	CMDAT1	DVDD18_IO0	CMMCLK	CMPCLK	RTN	RTP	VCCIO		VCCIO			VCCIO		VCCIO			GNDK	GNDK	MSDC0_CMD	MSDC0_CLK	MSDC0_RSTB	DVDD28_MSDC	G
H	RDN0_A	RDP0_A	GNDK							GNDK	GNDK							GNDK		GNDK				EINT14	DVDD28_NFI	H
J		RDN1_A	RDP1_A	RCP_A	RCN_A	GNDK	GNDK		GNDK	VCCK		GNDK			GNDK			VCCK_VPROC	VCCK_VPROC	VCCK_VPROC	VCCK_VPROC		EINT17	EINT16		J
K	RDN0	RDP0						DVDD18_EFUSE	FSOURCE_P	VCCK	GNDK	GNDK	GNDK	VCCK		VCCK_VPROC	VCCK_VPROC	VCCK_VPROC	VCCK_VPROC	VCCK_VPROC	VCCK_VPROC		EINT22	EINT21	EINT15	K
L		RDN1	RDP1						VCCK						VCCK_VPROC								PWRAP_SPI0_CSN	PWRAP_SPI0_MO	EINT23	L
M	AVDD18_MIPITX	RDN2	RDP2	RCN	RCP	AVSS18_MIPI			VCCK			GNDK	GNDK			VCCK_VPROC		CLKO_32K		RTC32K_CK	SDA2	SCL2	PWRAP_SPI0_	PWRAP_SPI0_		M
N		RDN3	RDP3	TCP	TCN	AVSS18_MIPI			VCCK	VCCK	GNDK	GNDK	GNDK	GNDK				AVSS22_XO_32		AVDD22_XO_32		SYSRSTB	SRCLKENA	PWRAP_INT	DVDD18_IO3	N
P	VRT	TDP0	TDN0				GNDK		VCCK			GNDK	GNDK	GNDK	GNDK	VCCK_VPROC		GNDK						WATCHDOG		P
R		TDP1	TDN1						VCCK	VCCK	GNDK	GNDK	GNDK	GNDK							AU_LOLP				AVDD28_AUDIO	R
T	TDP2	TDN2		TDP3	TDN3	AVSS18_MIPI			GNDK		GNDK	GNDK	GNDK	GNDK	VCCK		AU_TN		AU_HPR	AU_HPL	AU_LOLN		AVSS_AUDIO	AU_VIN0_N	AU_VIN0_P	T
U			DSI_TE	DVDD18_IO1	LCM_RST	DISP_PWM	SCL0	SDA0	GNDK		VCCK	VCCK	GNDK				AU_TP						AU_VIN2_N	AU_VIN2_P		U
V	JTDO	JTDI	JTMS												VCCK	VCCK	GNDK	AUX_IN2	AUX_IN4	AUX_IN0	AUX_IN1		AU_VIN1_N	AU_VIN1_P	AVDD22_AUDIO	V
W	TESTMODE	SCL1	JTCK		AVDD18_WBT_AFE	AVDD18_WBT		NC		GNDK				GNDK	GNDK			GNDK	AVDD18_HDMITX	AUX_IN3	AUX_IN5	AVDD18_PLLGP	ACCDET	AU_MICBIAS1	AU_MICBIAS0	W
Y		SDA1	KPCOL1	AVSS_CONN	AVSS_CONN	AVSS_CONN	AVSS_CONN		AVSS_CONN	AVSS_CONN	MSDC2_CLK			EINT6	EINT11			AVDD33_USB						AVSS18_AP	AVDD18_AP	Y
AA	KPROW1	KPCOL0	KPROW0	AVSS_CONN	AVSS_CONN	AVSS_CONN	AVSS_CONN	AVSS_CONN	AVSS_CONN	AVSS_CONN	MSDC2_CMD			EINT1	EINT8			CHD_DM_P0					CLKO_26M	AVSS18_PLLGP	REFP	AA
AB	UTXD1	URXD1	AVSS_CONN	AVSS_CONN	AVSS_CONN	AVSS_CONN	AVSS_CONN	AVSS_CONN		AVSS_CONN	MSDC2_DAT3			EINT5	EINT13			CHD_DP_P0			HDMITX_CLK_P		AVDD22_XO		AVSS22_XO	AB
AC	UTXD0	URXD0		AVSS_CONN	AVSS_CONN			EINT18		SPI_CS	MSDC2_DAT2			EINT0	EINT2			USB_VRT_P0			HDMITX_CLK_M	HDMITX_CH0_P			XO_IN	AC
AD	I2S_LRCK	I2S_BCK	AVSS_CONN	WB_RFIN	NC	AVDD33_WBT	EINT19	HDMISD	SPI_CK	SPI_MO		EINT9	EINT4	EINT10	EINT3	MRG_DO			USB_DM_P1	USB_DM_P0		HDMITX_CH0_M	HDMITX_CH1_P	HDMITX_CH2_P	AVSS22_XO	AD
AE	DUMMY	I2S_DATA_IN	AVSS_CONN				EINT20	CEC	HDMISCK	SPI_MI	MSDC2_DAT1	EINT12	EINT24	EINT25	URXD2	UTXD2	MRG_CLK	AVSS33_USB	USB_DP_P1	USB_DP_P0	AVSS18_HDMITX		HDMITX_CH1_M	HDMITX_CH2_M	DUMMY	AE
AF	DUMMY	DUMMY	AVSS_CONN					HTPLG	DVDD18_IO2		MSDC2_DAT0	DVDD28_MSDC		EINT7	DVDD28_DPI		MRG_SYNC	MRG_DI	AVDD18_USB		USB_VBUS_P0	HDMITX_REXT			DUMMY	AF
	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	

*Figure 2-1: DDR3 (2*16bits) ball map view of MT8516A*

2.1.2 Pin Coordinate

*Table 2-1: DDR3 (2*16bits) Pin Coordinates*

Ball Loc.	Ball name	Ball Loc.	Ball Name	Ball Loc.	Ball name
A1	GNDK	J6	GNDK	W1	TESTMODE
A2	GNDK	J7	GNDK	W2	SCL1
A3	ED1	J9	GNDK	W3	JTCK
A4	ED5	J10	VCCK	W5	AVDD18_WBT_AFE
A5	ED4	J12	GNDK	W6	AVDD18_WBT
A7	ED13	J15	GNDK	W8	NC
A8	ED11	J18	VCCK_VPROC	W10	GNDK
A9	ECKE	J19	VCCK_VPROC	W14	GNDK
A11	EA1	J20	VCCK_VPROC	W15	GNDK
A13	EA5	J21	VCCK_VPROC	W18	GNDK
A14	EA9	J23	EINT17	W19	AVDD18_HDMITX
A15	ECS0	J24	EINT16	W20	AUX_IN3
A16	ERAS	K1	RDN0	W21	AUX_IN5
A17	EBA2	K2	RDP0	W22	AVDD18_PLLGP
A19	ED30	K8	DVDD18_EFUSE	W23	ACCDET
A20	ED17	K9	FSOURCE_P	W24	AU_MICBIAS1
A21	ED21	K10	VCCK	W25	AU_MICBIAS0
A22	ED20	K11	GNDK	Y2	SDA1
A23	ED18	K12	GNDK	Y3	KPCOL1
A24	GNDK	K13	GNDK	Y4	AVSS_CONN
A25	GNDK	K14	VCCK	Y5	AVSS_CONN
B2	ED10	K16	VCCK_VPROC	Y6	AVSS_CONN
B3	ED14	K17	VCCK_VPROC	Y7	AVSS_CONN
B4	ED3	K18	VCCK_VPROC	Y9	AVSS_CONN
B5	ED7	K19	VCCK_VPROC	Y10	AVSS_CONN
B6	ED2	K20	VCCK_VPROC	Y11	MSDC2_CLK
B7	ED15	K21	VCCK_VPROC	Y14	EINT6
B9	EA10	K23	EINT22	Y15	EINT11
B10	EA8	K24	EINT21	Y18	AVDD33_USB
B11	EA4	K25	EINT15	Y24	AVSS18_AP
B12	EBA1	L2	RDN1	Y25	AVDD18_AP
B13	EA3	L3	RDP1	AA1	KPROW1
B14	EA13	L9	VCCK	AA2	KPCOL0
B15	ERWE	L15	VCCK_VPROC	AA3	KPROW0
B16	ECAS	L23	PWRAP_SPI0_CSN	AA4	AVSS_CONN
B17	ECS1	L24	PWRAP_SPI0_MO	AA5	AVSS_CONN
B18	ED24	L25	EINT23	AA6	AVSS_CONN
B19	ED26	M1	AVDD18_MIPITX	AA7	AVSS_CONN
B20	ED28	M2	RDN2	AA8	AVSS_CONN
B21	ED19	M3	RDP2	AA9	AVSS_CONN

Ball Loc.	Ball name	Ball Loc.	Ball Name	Ball Loc.	Ball name
B22	ED22	M4	RCN	AA10	AVSS_CONN
B23	ED16	M5	RCP	AA11	MSDC2_CMD
B24	ED31	M6	AVSS18_MIPI	AA14	EINT1
C1	REXTDN	M9	VCCK	AA15	EINT8
C3	ED8	M12	GNDK	AA18	CHD_DM_P0
C4	ED12	M13	GNDK	AA23	CLKO_26M
C5	ED6	M16	VCCK_VPROC	AA24	AVSS18_PLLGP
C6	ED0	M18	CLKO_32K	AA25	REFP
C7	ED9	M20	RTC32K_CK	AB1	UTXD1
C8	EDQM1	M21	SDA2	AB2	URXD1
C10	EA14	M22	SCL2	AB3	AVSS_CONN
C11	EA6	M23	PWRAP_SPI0_CK	AB4	AVSS_CONN
C12	EA0	M24	PWRAP_SPI0_MI	AB5	AVSS_CONN
C13	VCCIO	N2	RDN3	AB6	AVSS_CONN
C15	VCCIO	N3	RDP3	AB7	AVSS_CONN
C16	ERESET	N4	TCP	AB8	AVSS_CONN
C19	EDQM2	N5	TCN	AB10	AVSS_CONN
C21	EDQS2	N6	AVSS18_MIPI	AB11	MSDC2_DAT3
C22	ED23	N9	VCCK	AB14	EINT5
C23	ED29	N10	VCCK	AB15	EINT13
C24	ED25	N11	GNDK	AB18	CHD_DP_P0
C25	ED27	N12	GNDK	AB21	HDMITX_CLK_P
D1	MSDC1_CLK	N13	GNDK	AB23	AVDD22_XO
D2	MSDC1_DAT0	N14	GNDK	AB25	AVSS22_XO
D3	MSDC1_DAT1	N18	AVSS22_XO_32K	AC1	UTXD0
D7	EDQS1	N20	AVDD22_XO_32K	AC2	URXD0
D9	EDQS0	N22	SYSRSTB	AC4	AVSS_CONN
D11	ECLK0_B	N23	SRCLKENA	AC5	AVSS_CONN
D12	EA7	N24	PWRAP_INT	AC8	EINT18
D15	EBA0	N25	DVDD18_IO3	AC10	SPI_CS
D16	ECLK1	P1	VRT	AC11	MSDC2_DAT2
D19	EDQS3	P2	TDP0	AC14	EINT0
D21	EDQS2_B	P3	TDN0	AC15	EINT2
D23	MSDC0_DAT7	P7	GNDK	AC18	USB_VRT_P0
D24	MSDC0_DAT5	P9	VCCK	AC21	HDMITX_CLK_M
D25	MSDC0_DAT6	P12	GNDK	AC22	HDMITX_CH0_P
E2	MSDC1_DAT3	P13	GNDK	AC25	XO_IN
E3	MSDC1_CMD	P14	GNDK	AD1	I2S_LRCK
E5	EDQM0	P15	GNDK	AD2	I2S_BCK
E7	EDQS1_B	P16	VCCK_VPROC	AD3	AVSS_CONN
E9	EDQS0_B	P18	GNDK	AD4	WB_RFIN
E11	ECLK0	P24	WATCHDOG	AD5	NC
E12	EA12	R2	TDP1	AD6	AVDD33_WBT

Ball Loc.	Ball name	Ball Loc.	Ball Name	Ball Loc.	Ball name
E15	EA2	R3	TDN1	AD7	EINT19
E16	ECLK1_B	R9	VCCK	AD8	HDMISD
E19	EDQS3_B	R10	VCCK	AD9	SPI_CK
E21	EDQM3	R11	GNDK	AD10	SPI_MO
E23	MSDC0_DAT4	R12	GNDK	AD12	EINT9
E24	MSDC0_DAT3	R13	GNDK	AD13	EINT4
F1	DVDD28_MSDC1	R14	GNDK	AD14	EINT10
F2	MSDC1_DAT2	R21	AU_LOLP	AD15	EINT3
F5	AVDD18_MEMPLL	R25	AVDD28_AUDIO	AD16	MRG_DO
F7	GNDK	T1	TDP2	AD19	USB_DM_P1
F8	GNDK	T2	TDN2	AD20	USB_DM_P0
F9	GNDK	T4	TDP3	AD22	HDMITX_CH0_M
F12	EA11	T5	TDN3	AD23	HDMITX_CH1_P
F15	EA15	T6	AVSS18_MIPI	AD24	HDMITX_CH2_P
F16	GNDK	T9	GNDK	AD25	AVSS22_XO
F17	GNDK	T11	GNDK	AE1	DUMMY
F18	GNDK	T12	GNDK	AE2	I2S_DATA_IN
F20	GNDK	T13	GNDK	AE3	AVSS_CONN
F21	GNDK	T14	GNDK	AE7	EINT20
F23	MSDC0_DAT2	T15	VCCK	AE8	CEC
F24	MSDC0_DAT0	T17	AU_TN	AE9	HDMISCK
F25	MSDC0_DAT1	T19	AU_HPR	AE10	SPI_MI
G1	AVDD18_MIPIRX	T20	AU_HPL	AE11	MSDC2_DAT1
G3	CMDAT0	T21	AU_LOLN	AE12	EINT12
G4	CMDAT1	T23	AVSS_AUDIO	AE13	EINT24
G5	DVDD18_IO0	T24	AU_VIN0_N	AE14	EINT25
G6	CMMCLK	T25	AU_VIN0_P	AE15	URXD2
G7	CMPCLK	U3	DSI_TE	AE16	UTXD2
G8	RTN	U4	DVDD18_IO1	AE17	MRG_CLK
G9	RTP	U5	LCM_RST	AE18	AVSS33_USB
G10	VCCIO	U6	DISP_PWM	AE19	USB_DP_P1
G12	VCCIO	U7	SCL0	AE20	USB_DP_P0
G15	VCCIO	U8	SDA0	AE21	AVSS18_HDMITX
G17	VCCIO	U9	GNDK	AE23	HDMITX_CH1_M
G20	GNDK	U11	VCCK	AE24	HDMITX_CH2_M
G21	GNDK	U12	VCCK	AE25	DUMMY
G22	MSDC0_CMD	U13	GNDK	AF1	DUMMY
G23	MSDC0_CLK	U17	AU_TP	AF2	DUMMY
G24	MSDC0_RSTB	U23	AU_VIN2_N	AF3	AVSS_CONN
G25	DVDD28_MSDC0	U24	AU_VIN2_P	AF8	HTPLG
H1	RDN0_A	V1	JTDO	AF9	DVDD18_IO2
H2	RDP0_A	V2	JTDI	AF11	MSDC2_DAT0
H3	GNDK	V3	JTMS	AF12	DVDD28_MSDC2
H10	GNDK	V15	VCCK	AF14	EINT7

Ball Loc.	Ball name	Ball Loc.	Ball Name	Ball Loc.	Ball name
H11	GNDK	V16	VCCK	AF15	DVDD28_DPI
H18	GNDK	V17	GNDK	AF17	MRG_SYNC
H20	GNDK	V18	AUX_IN2	AF18	MRG_DI
H24	EINT14	V19	AUX_IN4	AF19	AVDD18_USB
H25	DVDD28_NFI	V20	AUX_IN0	AF21	USB_VBUS_P0
J2	RDN1_A	V21	AUX_IN1	AF22	HDMITX_REXT
J3	RDP1_A	V23	AU_VIN1_N	AF25	DUMMY
J4	RCP_A	V24	AU_VIN1_P		
J5	RCN_A	V25	AVDD22_AUDIO		

Table 2-2: DDR Pinmux Table

PKG/PCB Ball Location	Pin-Mux 1 - PCDDR3 16bitx2	Pin-Mux 2 - PCDDR4 16bitx2	Pin-Mux 3 - LP3_DSC	Pin-Mux 4 - LP3_POP	Pin-Mux 5 - DDR3_X8
C23	ED29	ED30	ED15	ED24	ED28
C25	ED27	ED26	ED11	ED31	ED26
B24	ED31	ED24	ED14	ED29	ED30
C24	ED25	ED28	ED10	ED28	ED24
B20	ED28	ED27	ED13	ED11	ED23
A19	ED30	ED29	ED12	ED9	ED17
B19	ED26	ED25	ED8	ED13	ED21
B18	ED24	ED31	ED9	ED10	ED19
E21	EDQM3	EDQM3	EDQM1	EDQM3	EDQM3
D19	EDQS3	EDQS3	EDQS1	EDQS3	EDQS3
E19	EDQS3_B	EDQS3_B	EDQS1_B	EDQS3_B	EDQS3_B
A22	ED20	ED22	ED26	ED27	ED27
B23	ED16	ED16	ED31	ED26	ED31
A23	ED18	ED20	ED27	ED25	ED25
B22	ED22	ED18	ED30	ED30	ED29
B21	ED19	ED19	ED28	ED14	ED22
C22	ED23	ED21	ED29	ED12	ED18
A21	ED21	ED23	ED25	ED8	ED16
A20	ED17	ED17	ED24	ED15	ED20
C19	EDQM2	EDQM2	EDQM3	EDQM1	EDQM2
C21	EDQS2	EDQS2	EDQS3	EDQS1	EDQS2
D21	EDQS2_B	EDQS2_B	EDQS3_B	EDQS1_B	EDQS2_B

PKG/PCB Ball Location	Pin-Mux 1 - PCDDR3 16bitx2	Pin-Mux 2 - PCDDR4 16bitx2	Pin-Mux 3 - LP3_DSC	Pin-Mux 4 - LP3_POP	Pin-Mux 5 - DDR3_X8
A08	ED11	ED12	ED7	ED5	ED12
B07	ED15	ED14	ED3	ED6	ED8
A07	ED13	ED8	ED2	ED7	ED10
C07	ED9	ED10	ED6	ED4	ED14
C03	ED8	ED13	ED5	ED19	ED3
C04	ED12	ED15	ED0	ED23	ED7
B02	ED10	ED9	ED4	ED17	ED5
B03	ED14	ED11	ED1	ED16	ED1
C08	EDQM1	EDQM1	EDQM0	EDQM0	EDQM1
D07	EDQS1	EDQS1	EDQS0	EDQS0	EDQS1
E07	EDQS1_B	EDQS1_B	EDQS0_B	EDQS0_B	EDQS1_B
C05	ED6	ED6	ED22	ED3	ED13
A05	ED4	ED4	ED18	ED1	ED11
C06	ED0	ED0	ED23	ED0	ED15
B06	ED2	ED2	ED19	ED2	ED9
B05	ED7	ED7	ED21	ED20	ED4
A04	ED5	ED3	ED17	ED21	ED6
B04	ED3	ED5	ED20	ED18	ED2
A03	ED1	ED1	ED16	ED22	ED0
E05	EDQM0	EDQM0	EDQM2	EDQM2	EDQM0
D09	EDQS0	EDQS0	EDQS2	EDQS2	EDQS0
E09	EDQS0_B	EDQS0_B	EDQS2_B	EDQS2_B	EDQS0_B
B11	EA4	EA3	EA3	EA4	EA4
D15	EBA0	EA12	VDDIO	VDDIO	EBA0
E15	EA2	EBG1	VDDIO	VDDIO	EA2
A16	ERAS	ERAS	EA6	EA7	ERAS
B16	ECAS	ECAS	EA8	EA9	ECAS
B12	EBA1	EBA1	VDDIO	VDDIO	EBA1
A17	EBA2	EBG0	VDDIO	VDDIO	EBA2
B15	ERWE	ERWE	VDDIO	VDDIO	ERWE
F15	EA15	EACT#	EA9	EA8	EA15
B14	EA13	EA2	VDDIO	VDDIO	EA13
B09	EA10	EA9	VDDIO	VDDIO	EA10
A14	EA9	EA8	EA7	EA6	EA9
A09	ECKE	ECKE	ECKE0	ECKE0	ECKE0

PKG/PCB Ball Location	Pin-Mux 1 - PCDDR3 16bitx2	Pin-Mux 2 - PCDDR4 16bitx2	Pin-Mux 3 - LP3_DSC	Pin-Mux 4 - LP3_POP	Pin-Mux 5 - DDR3_X8
B13	EA3	EBA0	VDDIO	VDDIO	EA3
C12	EA0	EA4	EA1	EA1	EA0
A13	EA5	EA6	EA5	EA5	EA5
E12	EA12	EA10	VDDIO	VDDIO	EA12
D12	EA7	EA0	EA0	EA3	EA7
F12	EA11	EA11	EA2	EA2	EA11
A11	EA1	EA5	VDDIO	VDDIO	EA1
C11	EA6	EA1	ECKE1	ECKE1	EA6
B10	EA8	EA7	EA4	EA0	EA8
C10	EA14	EA13	VDDIO	VDDIO	EA14
E11	ECLK0	ECLK0	ECLK0	ECLK0	ECLK0
D11	ECLK0_B	ECLK0_B	ECLK0_B	ECLK0_B	ECLK0_B
D16	ECLK1	ECLK1	N/A	N/A	ECLK1
E16	ECLK1_B	ECLK1_B	N/A	N/A	ECLK1_B
A15	ECS0	ECS0	ECS0	ECS0	ECS0
B17	ECS1	EODT	ECS1	ECS1	ECS1
C16	ERESET	ERESET	N/A	N/A	ERESET
C01	REXTDN	REXTDN	REXTDN	REXTDN	REXTDN

2.1.3 Detailed Pin Description

Table 2-3: 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 2-4: DI/DO/DIO type

Type	Description
GPIO	General purpose 1.8V IO
KP2KIO	Keypad 2K resistance IO

Type	Description
KP200KIO	Keypad 200K resistance IO
GPIOOD	General purpose 3.3V IO
I2C33IO	I2C IO
I2C5VTIO	SPI IO
MSDCIO	MSDC IO
AGPIO	Analog general purpose 1.8V IO

Table 2-5: DI/DO/DIO: GPIO type specification

Electrical Characteristics and Operating Conditions of 1.8V Applications						
Parameters	Descriptions	Min	Typ	Max	UNIT	Note
VDDIO	Supply voltage of IO power	1.62	1.8	1.98	V	
Inputs						
VIH	Input logic low voltage	0.65*VDDIO		VDDIO+0.3	V	
VIL	Input logic high voltage	-0.3		0.35*VDDIO	V	
Rpu	Input pull-up resistance	40	75	190	Kohm	
Rpd	Input pull-down resistance	40	75	190	Kohm	
Outputs						
VOH(DC)	DC Output logic low voltage	0.75*VDDIO			V	
VOL(DC)	DC Output logic high voltage			0.25*VDDIO	V	

Table 2-6: DIO: KP2KIO type specification

Electrical Characteristics and Operating Conditions of 1.8V Applications						
Parameters	Descriptions	Min	Typ	Max	UNIT	***Note***
VDDIO	Supply voltage of IO power	1.62	1.8	1.98	V	
Inputs						
VIH	Input logic low voltage	0.65*VDDIO		VDDIO+0.3	V	
VIL	Input logic high voltage	-0.3		0.35*VDDIO	V	
Rpu	Input pull-up resistance			2	Kohm	

Electrical Characteristics and Operating Conditions of 1.8V Applications						
Rpd	Input pull-down resistance			2	Kohm	
Outputs						
VOH(DC)	DC Output logic low voltage	0.75*VDDIO			V	
VOL(DC)	DC Output logic high voltage			0.25*VDDIO	V	

Table 2-7: 2Kohm type Pull up/down Setting

E	Pu/Pd	R1	R0	R value
0	0	0	0	High-Z
0	0	0	1	PU-75k
0	0	1	0	PU-2k
0	0	1	1	PU-75k/2k
0	1	0	0	High-Z
0	1	0	1	PD-75k
0	1	1	0	PD-2k
0	1	1	1	PD-75k/2k
1	X	X	X	High-Z

Table 2-8: DIO: KP200KIO type specification

Electrical Characteristics and Operating Conditions of 1.8V Applications						
Parameters	*Descriptions*	*Min*	*Typ*	*Max*	*UNIT*	*Note*
VDDIO	Supply voltage of IO power	1.62	1.8	1.98	V	
Inputs						
VIH	*Input logic low voltage*	*0.65*VDDIO*		*VDDIO+0.3*	*V*	
VIL	*Input logic high voltage*	*-0.3*		*0.35*VDDIO*	*V*	
Rpu	*Input pull-up resistance*	*200*			*Kohm*	
Rpd	*Input pull-down resistance*	*200*			*Kohm*	
Outputs						

Electrical Characteristics and Operating Conditions of 1.8V Applications						
VOH(DC)	*DC Output logic low voltage*	*0.75*VDDIO*			*V*	
VOL(DC)	*DC Output logic high voltage*			*0.25*VDDIO*	*V*	

Table 2-9: 200Kohm type Pull up/down Setting

E	Pu/Pd	R1	R0	R value
0	0	0	0	High-Z
0	0	0	1	PU-75k
0	0	1	0	PU-200k
0	0	1	1	PU-75k/200k
0	1	0	0	High-Z
0	1	0	1	PD-75k
0	1	1	0	PD-200k
0	1	1	1	PD-75k/200k
1	X	X	X	High-Z

Table 2-10: DIO: GPIOOD type specification

Operating Conditions of 3.3V Applications						
Parameters	Descriptions	Min	Typ	Max	UNIT	Note
VCC3IO	Supply voltage of SD IO power	2.97	3.3	3.63	V	
Outputs						
VOH(DC)	DC Output logic low voltage	VCC3IO-0.4V		VCC3IO+0.3	V	VCC3IO=min, IOH= - 2mA
VOL(DC)	DC Output logic high voltage	-0.3		0.4	V	VCC3IO=min, IOL= -2mA
Inputs						
VIH	Input logic low voltage	2.0		VCC3IO+0.3	V	
VIL	Input logic high voltage	-0.3		0.8	V	
Rpu1	Input pull-up resistance	40	75	190	Kohm	control pin PU=1
Rpd1	Input pull-down resistance	40	75	190	Kohm	control pin PD=1

Operating Conditions of 1.8V Applications						
Parameters	Descriptions	Min	Typ	Max	UNIT	Note
VCC3IOIO	Supply voltage of SIM IO power	1.7	1.8	1.9	V	
Outputs						
IO Voh	IO output Ioh=1mA	VCC3IO-0.2		VCC3IO+0.3	V	
IO Vol	IO output Iol=-1mA	-0.3		0.2	V	
Inputs						
VIH	Input logic low voltage	1.27		VCC3IO+0.3	V	
VIL	Input logic high voltage	-0.3		0.58	V	
Rpu1	Input pull-up resistance	10	50	100	Kohm	control pin PU=1
Rpd1	Input pull-down resistance	10	50	100	Kohm	control pin PD=1

Table 2-11: DIO: I2C33IO type specification

Electrical Characteristics and Operating Conditions of 1.8V Applications						
Parameters	**Descriptions**	**Min**	**Typ**	**Max**	**UNIT**	**Note**
VDDIO	Supply voltage of IO power	1.62	1.8	1.98	V	
Inputs						
VIH	Input logic low voltage	0.65*VDDIO		VDDIO+0.3	V	
VIL	Input logic high voltage	-0.3		0.35*VDDIO	V	
Rpd	Input pull-down resistance	40	75	350	Kohm	
Outputs						
VOL(DC)	DC Output logic low voltage (VIN>=2V)			0.4	V	
VOL(DC)	DC Output logic low voltage (VIN<2V)			0.2*VDDIO	V	
External Pull-up Resistance						
Rpull-up	External Pull-up resistance		1.0		Kohm	

Table 2-12: DIO: I2C5VTIO type specification

Electrical Characteristics and Operating Conditions of 1.8V Applications						
Parameters	**Descriptions**	**Min**	**Typ**	**Max**	**UNIT**	**Note**
VDDIO	Supply voltage of IO power	1.62	1.8	1.98	V	
Inputs						
VIH	Input logic low voltage	0.65*VDDIO		VDDIO+0.3	V	
VIL	Input logic high voltage	-0.3		0.35*VDDIO	V	
Rpd	Input pull-down resistance	40	75	550	Kohm	
Outputs						
VOL(DC)	DC Output logic low voltage			0.2*VDDIO	V	
IOL(DC)	DC Output logic low current	3			mA	
External Pull-up Resistance						
Rpull-up	External Pull-up resistance		1.0		Kohm	

Table 2-13: DIO: MSDCIO type specifications

Operating Conditions of 3.3V Applications						
Parameters	Descriptions	Min	Typ	Max	UNIT	Note
VCC3IO	Supply voltage of SD IO power	2.97	3.3	3.63	V	
Outputs						
IO Voh	IO output Ioh=2mA	0.75*VCC3IO		VCC3IO+0.3	V	
IO Vol	IO output Iol=-2mA	-0.3		0.125*VCC3IO	V	
Inputs						
VIH	Input logic low voltage	0.625*VCC3IO		VCC3IO+0.3	V	
VIL	Input logic high voltage	-0.3		0.25*VCC3IO	V	
Rpu1	Input pull-up resistance	10	50	100	Kohm	control pin R0=0, R1=1

Operating Conditions of 3.3V Applications						
Rpd1	Input pull-down resistance	10	50	100	Kohm	control pin R0=0, R1=1
Rpu0	Input pull-up resistance	5	7.5	10	Kohm	control pin R0=1, R1=0
Rpd0	Input pull-down resistance	5	7.5	10	Kohm	control pin R0=1, R1=0

Operating Conditions of 1.8V Applications						
Parameters	Descriptions	Min	Typ	Max	UNIT	Note
VCC3IOIO	Supply voltage of SIM IO power	1.7	1.8	1.9	V	
Outputs						
IO Voh	IO output Ioh=1mA	VCC3IO-0.45		VCC3IO+0.3	V	
IO Vol	IO output Iol=-1mA	-0.3		0.45	V	
Inputs						
VIH	Input logic low voltage	0.65*VCC3IO		VCC3IO+0.3	V	
VIL	Input logic high voltage	-0.3		0.35*VCC3IO	V	
Rpu1	Input pull-up resistance	10	50	100	Kohm	4, control pin R0=0, R1=1
Rpd1	Input pull-down resistance	10	50	100	Kohm	4, control pin R0=0, R1=1
Rpu0	Input pull-up resistance	5	7.5	10	Kohm	4, control pin R0=1, R1=0
Rpd0	Input pull-down resistance	5	7.5	10	Kohm	4, control pin R0=1, R1=0

Table 2-14: DIO: AGPIO type specification

Electrical Characteristics and Operating Conditions of 1.8V Applications						
Parameters	Descriptions	Min	Typ	Max	UNIT	Note
VDDIO	Supply voltage of IO power	1.62	1.8	1.98	V	
Inputs						
VIH	Input logic low voltage	0.65*VDDIO		VDDIO+0.3	V	

Electrical Characteristics and Operating Conditions of 1.8V Applications						
VIL	Input logic high voltage	-0.3		0.35*VDDIO	V	
Rpu	Input pull-up resistance	40	75	190	Kohm	
Rpd	Input pull-down resistance	40	75	190	Kohm	
Outputs						
VOH(DC)	DC Output logic low voltage	0.75*VDDIO			V	
VOL(DC)	DC Output logic high voltage			0.25*VDDIO	V	

Table 2-15: Detailed pin description

Pin name	Type	DI/DO/DIO type	Description	Power domain
SYSTEM				
SYSRSTB	DI	GPIO	System reset input	DVDD18_IO3
WATCHDOG	DIO	GPIO	Watchdog reset output	DVDD18_IO3
TESTMODE	DI	GPIO	Test mode	DVDD18_IO1
RTC32K_CK	DIO	GPIO	32K clock input	DVDD18_IO3
SRCLKENA	DIO	GPIO	26MHz co-clock enable output	DVDD18_IO3
PMIC				
PWRAP_SPI0_MO	DIO	GPIO	PMIC SPI control interface	DVDD18_IO3
PWRAP_SPI0_MI	DIO	GPIO	PMIC SPI control interface	DVDD18_IO3
PWRAP_SPI0_CSN	DIO	GPIO	PMIC SPI control interface	DVDD18_IO3
PWRAP_SPI0_CK	DIO	GPIO	PMIC SPI control interface	DVDD18_IO3
PWRAP_INT	DIO	GPIO	PMIC SPI control interface	DVDD18_IO3
JTAG				
JTCK	DIO	GPIO	JTCK	DVDD18_IO1
JTDO	DIO	GPIO	JTDO	DVDD18_IO1
JTDI	DIO	GPIO	JTDI	DVDD18_IO1
JTMS	DIO	GPIO	JTMS	DVDD18_IO1
LCD				
DISP_PWM	DIO	GPIO	Display PWM output	DVDD18_IO1
DSI_TE	DIO	GPIO	DSI tearing effect control	DVDD18_IO1
LCM_RST	DIO	GPIO	LCM reset	DVDD18_IO1
KeyPad				
KPCOL0	DIO	KP200KIO	Key Pad column 0	DVDD18_IO1
KPCOL1	DIO	KP200KIO	Key Pad column 1	DVDD18_IO1
KPROW0	DIO	KP2KIO	Key Pad raw 0	DVDD18_IO1
KPROW1	DIO	KP2KIO	Key Pad raw 1	DVDD18_IO1

Pin name	Type	DI/DO/DIO type	Description	Power domain
I2S				
I2S_DATA_IN	DIO	GPIO	I2S data input pin	DVDD18_IO1
I2S_BCK	DIO	GPIO	I2S clock	DVDD18_IO1
I2S_LRCK	DIO	GPIO	I2S word select	DVDD18_IO1
I2S merge interface				
MRG_DO	DIO	GPIOOD	MTK audio interface	DVDD28_DPI
MRG_CLK	DIO	GPIOOD	MTK audio interface	DVDD28_DPI
MRG_DI	DIO	GPIOOD	MTK audio interface	DVDD28_DPI
MRG_SYNC	DIO	GPIOOD	MTK audio interface	DVDD28_DPI
EINT				
EINT0	DIO	GPIOOD	External interrupt 0	DVDD28_DPI
EINT1	DIO	GPIOOD	External interrupt 1	DVDD28_DPI
EINT2	DIO	GPIOOD	External interrupt 2	DVDD28_DPI
EINT3	DIO	GPIOOD	External interrupt 3	DVDD28_DPI
EINT4	DIO	GPIOOD	External interrupt 4	DVDD28_DPI
EINT5	DIO	GPIOOD	External interrupt 5	DVDD28_DPI
EINT6	DIO	GPIOOD	External interrupt 6	DVDD28_DPI
EINT7	DIO	GPIOOD	External interrupt 7	DVDD28_DPI
EINT8	DIO	GPIOOD	External interrupt 8	DVDD28_DPI
EINT9	DIO	GPIOOD	External interrupt 9	DVDD28_DPI
EINT10	DIO	GPIOOD	External interrupt 10	DVDD28_DPI
EINT11	DIO	GPIOOD	External interrupt 11	DVDD28_DPI
EINT12	DIO	GPIOOD	External interrupt 12	DVDD28_DPI
EINT13	DIO	GPIOOD	External interrupt 13	DVDD28_DPI
EINT18	DIO	GPIOOD	External interrupt 18	DVDD18_IO2
EINT19	DIO	GPIOOD	External interrupt 19	DVDD18_IO2
EINT20	DIO	GPIOOD	External interrupt 20	DVDD18_IO2
EINT24	DIO	GPIOOD	External interrupt 24	DVDD28_DPI
EINT25	DIO	GPIOOD	External interrupt 25	DVDD28_DPI
UART				
URXD0	DIO	AGPIO	UART0 RX	DVDD18_IO1
UTXD0	DIO	AGPIO	UART0 TX	DVDD18_IO1
URXD1	DIO	AGPIO	UART1 RX	DVDD18_IO1
UTXD1	DIO	AGPIO	UART1 TX	DVDD18_IO1
URXD2	DIO	AGPIO	UART2 RX	DVDD28_DPI
UTXD2	DIO	AGPIO	UART2 TX	DVDD28_DPI
SPI				
SPI_CS	DIO	GPIO	SPI chip select	DVDD18_IO2
SPI_MI	DIO	GPIO	SPI data in	DVDD18_IO2
SPI_MO	DIO	GPIO	SPI data out	DVDD18_IO2
SPI_CK	DIO	GPIO	SPI clock	DVDD18_IO2

Pin name	Type	DI/DO/DIO type	Description	Power domain
NAND Flash Interface				
EINT14	DIO	MSDCIO	NCLE	DVDD28_NFI
EINT15	DIO	MSDCIO	NCEB1	DVDD28_NFI
EINT16	DIO	MSDCIO	NCEB0	DVDD28_NFI
EINT17	DIO	MSDCIO	NREB	DVDD28_NFI
EINT21	DIO	MSDCIO	NRNB	DVDD28_NFI
EINT22	DIO	MSDCIO	NRE_C	DVDD28_NFI
EINT23	DIO	MSDCIO	NDQS_C	DVDD28_NFI
MSDC0				
MSDC0_DAT7	DIO	MSDCIO	MSDC0 data7 pin / NLD7	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_DAT6	DIO	MSDCIO	MSDC0 data6 pin / NLD6	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_DAT5	DIO	MSDCIO	MSDC0 data5 pin / NLD4	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_RSTB	DIO	MSDCIO	MSDC0 reset output /NLD0	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_DAT4	DIO	MSDCIO	MSDC0 data4 pin / NLD3	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_DAT2	DIO	MSDCIO	MSDC0 data2 pin / NLD5	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_DAT3	DIO	MSDCIO	MSDC0 data3 pin / NLD1	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_CMD	DIO	MSDCIO	MSDC0 command pin / NALE	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_CLK	DIO	MSDCIO	MSDC0 clock output / NWEB	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_DAT1	DIO	MSDCIO	MSDC0 data1 pin / NLD8	DVDD28_MSDC0 / DVDD18_IO3
MSDC0_DAT0	DIO	MSDCIO	MSDC0 data0 pin / NLD2	DVDD28_MSDC0 / DVDD18_IO3
MSDC1				
MSDC1_CLK	DIO	MSDCIO	MSDC1 clock output	DVDD28_MSDC1
MSDC1_CMD	DIO	MSDCIO	MSDC1 command pin	DVDD28_MSDC1
MSDC1_DAT0	DIO	MSDCIO	MSDC1 data0 pin	DVDD28_MSDC1
MSDC1_DAT1	DIO	MSDCIO	MSDC1 data1 pin	DVDD28_MSDC1
MSDC1_DAT2	DIO	MSDCIO	MSDC1 data2 pin	DVDD28_MSDC1
MSDC1_DAT3	DIO	MSDCIO	MSDC1 data3 pin	DVDD28_MSDC1
MSDC2				
MSDC2_CLK	DIO	MSDCIO	MSDC2 clock output	DVDD28_MSDC2
MSDC2_CMD	DIO	MSDCIO	MSDC2 command pin	DVDD28_MSDC2

Pin name	Type	DI/DO/DIO type	Description	Power domain
MSDC2_DAT0	DIO	MSDCIO	MSDC2 data0 pin	DVDD28_MSDC2
MSDC2_DAT1	DIO	MSDCIO	MSDC2 data1 pin	DVDD28_MSDC2
MSDC2_DAT2	DIO	MSDCIO	MSDC2 data2 pin	DVDD28_MSDC2
MSDC2_DAT3	DIO	MSDCIO	MSDC2 data3 pin	DVDD28_MSDC2
EFUSE				
FSOURCE_P	P		E-FUSE blowing power control	DVDD18_EFUSE
EMI				
ECLK0	AIO		DRAM clock 0 output	VCCIO
ECLK0_B	AIO		DRAM clock 0 output #	VCCIO
ECLK1	AIO		DRAM clock 1 output	VCCIO
ECLK1_B	AIO		DRAM clock 1 output #	VCCIO
ECKE	AIO		DRAM command output CKE	VCCIO
ECS0	AIO		DRAM chip select 0 #	VCCIO
ECS1	AIO		DRAM chip select 1 #	VCCIO
ECAS	AIO		DRAM cmd column strobe output	VCCIO
ERAS	AIO		DRAM cmd row strobe ouput	VCCIO
ERESET	AIO		DRAM reset pin	VCCIO
ERWE	AIO		DRAM cmd write enable	VCCIO
EA0	AIO		DRAM address output 0	VCCIO
EA1	AIO		DRAM address output 1	VCCIO
EA2	AIO		DRAM address output 2	VCCIO
EA3	AIO		DRAM address output 3	VCCIO
EA4	AIO		DRAM address output 4	VCCIO
EA5	AIO		DRAM address output 5	VCCIO
EA6	AIO		DRAM address output 6	VCCIO
EA7	AIO		DRAM address output 7	VCCIO
EA8	AIO		DRAM address output 8	VCCIO
EA9	AIO		DRAM address output 9	VCCIO
EA10	AIO		DRAM address output 10	VCCIO
EA11	AIO		DRAM address output 11	VCCIO
EA12	AIO		DRAM address output 12	VCCIO
EA13	AIO		DRAM address output 13	VCCIO
EA14	AIO		DRAM address output 14	VCCIO
EA15	AIO		DRAM address output 15	VCCIO
EBA0	AIO		DRAM banks address	VCCIO
EBA1	AIO		DRAM banks address	VCCIO
EBA2	AIO		DRAM banks address	VCCIO
EDQM0	AIO		DRAM DQM 0	VCCIO
EDQM1	AIO		DRAM DQM 1	VCCIO
EDQM2	AIO		DRAM DQM 2	VCCIO
EDQM3	AIO		DRAM DQM 3	VCCIO

Pin name	Type	DI/DO/DIO type	Description	Power domain
EDQS0	AIO		DRAM DQS 0	VCCIO
EDQS0_B	AIO		DRAM DQS 0 #	VCCIO
EDQS1	AIO		DRAM DQS 1	VCCIO
EDQS1_B	AIO		DRAM DQS 1 #	VCCIO
EDQS2	AIO		DRAM DQS 2	VCCIO
EDQS2_B	AIO		DRAM DQS 2 #	VCCIO
EDQS3	AIO		DRAM DQS 3	VCCIO
EDQS3_B	AIO		DRAM DQS 3 #	VCCIO
ED0	AIO		DRAM data pin 0	VCCIO
ED1	AIO		DRAM data pin 1	VCCIO
ED2	AIO		DRAM data pin 2	VCCIO
ED3	AIO		DRAM data pin 3	VCCIO
ED4	AIO		DRAM data pin 4	VCCIO
ED5	AIO		DRAM data pin 5	VCCIO
ED6	AIO		DRAM data pin 6	VCCIO
ED7	AIO		DRAM data pin 7	VCCIO
ED8	AIO		DRAM data pin 8	VCCIO
ED9	AIO		DRAM data pin 9	VCCIO
ED10	AIO		DRAM data pin 10	VCCIO
ED11	AIO		DRAM data pin 11	VCCIO
ED12	AIO		DRAM data pin 12	VCCIO
ED13	AIO		DRAM data pin 13	VCCIO
ED14	AIO		DRAM data pin 14	VCCIO
ED15	AIO		DRAM data pin 15	VCCIO
ED16	AIO		DRAM data pin 16	VCCIO
ED17	AIO		DRAM data pin 17	VCCIO
ED18	AIO		DRAM data pin 18	VCCIO
ED19	AIO		DRAM data pin 19	VCCIO
ED20	AIO		DRAM data pin 20	VCCIO
ED21	AIO		DRAM data pin 21	VCCIO
ED22	AIO		DRAM data pin 22	VCCIO
ED23	AIO		DRAM data pin 23	VCCIO
ED24	AIO		DRAM data pin 24	VCCIO
ED25	AIO		DRAM data pin 25	VCCIO
ED26	AIO		DRAM data pin 26	VCCIO
ED27	AIO		DRAM data pin 27	VCCIO
ED28	AIO		DRAM data pin 28	VCCIO
ED29	AIO		DRAM data pin 29	VCCIO
ED30	AIO		DRAM data pin 30	VCCIO
ED31	AIO		DRAM data pin 31	VCCIO
REXTDN	AIO		DRAM REXTDN pin	VCCIO

Pin name	Type	DI/DO/DIO type	Description	Power domain
RTN	AIO		NC	VCCIO
RTP	AIO		DRAM voltage reference 2, connected to ½ VCCIO	VCCIO
CAM				
CMPCLK	DIO	GPIO	Pixel clock from sensor	DVDD18_IO0
CMMCLK	DIO	GPIO	Master clock to sensor	DVDD18_IO0
CMDAT0	DIO	GPIO	CAM sensor Data0	DVDD18_IO0
CMDAT1	DIO	GPIO	CAM sensor Data1	DVDD18_IO0
I2C0				
SCL0	DIO	I2C33IO	I2C0 clock	DVDD18_IO1
SDA0	DIO	I2C33IO	I2C0 data	DVDD18_IO1
I2C1				
SCL1	DIO	I2C33IO	I2C1 clock	DVDD18_IO1
SDA1	DIO	I2C33IO	I2C1 data	DVDD18_IO1
I2C2				
SCL2	DIO	I2C33IO	I2C2 clock	DVDD18_IO3
SDA2	DIO	I2C33IO	I2C2 data	DVDD18_IO3
XO				
XO_IN	AIO		26MHz clock input for AP	AVDD22_XO
CLKO_26M	AIO		26MHz clock output to PMIC	AVDD22_XO
CLKO_32K	AIO		32KHz clock output to PMIC	AVDD22_XO_32K
ABB				
REFP	AIO		Positive reference port for internal circuit	AVDD18_AP
AUX_IN0	AIO		AuxADC external input channel 0	AVDD18_AP
AUX_IN1	AIO		AuxADC external input channel 1	AVDD18_AP
AUX_IN2	AIO		AuxADC external input channel 2	AVDD18_AP
AUX_IN3	AIO		AuxADC external input channel 3	AVDD18_AP
AUX_IN4	AIO		AuxADC external input channel 4	AVDD18_AP
AUX_IN5	AIO		AuxADC external input channel 5	AVDD18_AP
WBT				
WB_RFIN	AIO		WF/BT RF IO port	AVDD33_WBT
MIPI				
TDN3	AIO		DSI0 lane3 N / LVDSTX lane3 N	DVDD18_MIPITX
TDP3	AIO		DSI0 lane3 P / LVDSTX lane3 P	DVDD18_MIPITX
TDN2	AIO		DSI0 lane2 N / LVDSTX CK lane N	DVDD18_MIPITX
TDP2	AIO		DSI0 lane2 P / LVDSTX CK lane P	DVDD18_MIPITX
TCN	AIO		DSI0 CK lane N / LVDSTX lane2 N	DVDD18_MIPITX
TCP	AIO		DSI0 CK lane P / LVDSTX lane2 P	DVDD18_MIPITX

Pin name	Type	DI/DO/DIO type	Description	Power domain
TDN1	AIO		DSI0 lane1 N / LVDSTX lane1 N	DVDD18_MIPITX
TDP1	AIO		DSI0 lane1 P / LVDSTX lane1 P	DVDD18_MIPITX
TDN0	AIO		DSI0 lane0 N / LVDSTX lane0 N	DVDD18_MIPITX
TDP0	AIO		DSI0 lane0 P / LVDSTX lane0 P	DVDD18_MIPITX
VRT	AO		External resistor for DSI bias Connect 1.5K ohm 1% resistor to ground	DVDD18_MIPITX
RDN3	AIO		CSI0 lane3 N / CAM sensor Data4	DVDD18_MIPIRX
RDP3	AIO		CSI0 lane3 P / CAM sensor Data5	DVDD18_MIPIRX
RDN2	AIO		CSI0 lane2 N / CAM sensor Data8	DVDD18_MIPIRX
RDP2	AIO		CSI0 lane2 P / CAM sensor Data9	DVDD18_MIPIRX
RCN	AIO		CSI0 CK lane N	DVDD18_MIPIRX
RCP	AIO		CSI0 CK lane P	DVDD18_MIPIRX
RDN1	AIO		CSI0 lane1 N	DVDD18_MIPIRX
RDP1	AIO		CSI0 lane1 P	DVDD18_MIPIRX
RDN0	AIO		CSI0 lane0 N	DVDD18_MIPIRX
RDP0	AIO		CSI0 lane0 P	DVDD18_MIPIRX
RDN1_A	AIO		CSI1 lane1 N / CAM sensor Data2	DVDD18_MIPIRX
RDP1_A	AIO		CSI1 lane1 P / CAM sensor Data3	DVDD18_MIPIRX
RCN_A	AIO		CSI1 CK lane N / CAM sensor Data6	DVDD18_MIPIRX
RCP_A	AIO		CSI1 CK lane P / CAM sensor Data7	DVDD18_MIPIRX
RDN0_A	AIO		CSI1 lane0 N / CAM sensor HSYNC	DVDD18_MIPIRX
RDP0_A	AIO		CSI1 lane0 P / CAM sensor VSYNC	DVDD18_MIPIRX
USB				
USB_DP_P0	AIO		USB port0 D+ differential data line	AVDD33_USB
USB_DM_P0	AIO		USB port0 D- differential data line	AVDD33_USB
CHD_DP_P0	AIO		BC1.1 Charger DP	AVDD33_USB
CHD_DM_P0	AIO		BC1.1 Charger DM	AVDD33_USB
USB_VRT_P0	AO		USB output for bias current; connect with 5.11K 1% Ohm to GND	AVDD18_USB
USB_VBUS_P0	AI		Power for connected device	AVDD18_USB
USB_DP_P1	AIO		USB port1 D+ differential data line	AVDD33_USB
USB_DM_P1	AIO		USB port1 D- differential data line	AVDD33_USB
HDMI Transmitter				
CEC	DIO	I2C5VTIO	HDMITX CEC	DVDD18_IO2
HTPLG	DIO	I2C5VTIO	HDMITX Hot Plug Detection Pin	DVDD18_IO2

Pin name	Type	DI/DO/DIO type	Description	Power domain
HDMISCK	DIO	I2C5VTIO	HDMITX I2C clock pin	DVDD18_IO2
HDMISD	DIO	I2C5VTIO	HDMITX I2C data pin	DVDD18_IO2
HDMITX_REXT	AIO		External resistor for HDMITX bias	AVDD18_HDMITX
HDMITX_CLK_M	AIO		HDMITX channel CK M	AVDD18_HDMITX
HDMITX_CLK_P	AIO		HDMITX channel CK P	AVDD18_HDMITX
HDMITX_CH0_M	AIO		HDMITX channel 0 M	AVDD18_HDMITX
HDMITX_CH0_P	AIO		HDMITX channel 0 P	AVDD18_HDMITX
HDMITX_CH1_M	AIO		HDMITX channel 1 M	AVDD18_HDMITX
HDMITX_CH1_P	AIO		HDMITX channel 1 P	AVDD18_HDMITX
HDMITX_CH2_M	AIO		HDMITX channel 2 M	AVDD18_HDMITX
HDMITX_CH2_P	AIO		HDMITX channel 2 P	AVDD18_HDMITX
Audio Codec				
ACCDET	AIO		Accessory detection input	AVDD28_AUDIO
AU_MICBIAS1	AIO		Microphone bias for earphone	AVDD28_AUDIO
AU_MICBIAS0	AIO		Microphone bias for main and 2nd microphone	AVDD28_AUDIO
AU_VIN0_P	AIO		Audio analog input 1 positive port	AVDD22_AUDIO
AU_VIN0_N	AIO		Audio analog input 1 negative port	AVDD22_AUDIO
AU_VIN1_P	AIO		Audio analog input 2 positive port	AVDD22_AUDIO
AU_VIN1_N	AIO		Audio analog input 2 negative port	AVDD22_AUDIO
AU_VIN2_P	AIO		Audio analog input 3 positive port	AVDD22_AUDIO
AU_VIN2_N	AIO		Audio analog input 3 negative port	AVDD22_AUDIO
AU_LOLN	AIO		Lineout N to drive SPK AMP	AVDD28_AUDIO
AU_LOLP	AIO		Lineout P to drive SPK AMP	AVDD28_AUDIO
AU_HPL	AIO		L-CH headphone output	AVDD28_AUDIO
AU_HPR	AIO		R-CH headphone output	AVDD28_AUDIO
AU_TN	AIO		Audio Codec Test Pin N	AVDD22_AUDIO
AU_TP	AIO		Audio Codec Test Pin P	AVDD22_AUDIO
Analog power				
AVDD18_PLLGP	P		Analog power input 1.8V for PLL and oscillator	-
AVDD18_AP	P		Analog power input 1.8V for AuxADC, TSENSE	-
AVDD18_MEMPLL	P		Analog power for MEMPLL	-
AVDD18_WBT	P		Analog power 1.8V for WBT RF	-
AVDD18_WBT_AFE	P		Analog power 1.8V for WBT AFE	-
AVDD33_WBT	P		Analog power 3.5V (default) for WBT TX PA and IQM	-
AVDD18_MIPITX	P		Analog power for MIPI DSI	-
AVDD18_MIPIRX	P		Analog power for MIPI CSI	-

Pin name	Type	DI/DO/DIO type	Description	Power domain
AVDD18_HDMITX	P		Analog power input 1.8V for HDMI/MHL transmitter	-
AVDD18_USB	P		Analog power 1.8V for USB	-
AVDD33_USB	P		Analog power 3.3V for USB	-
AVDD22_AUDIO	P		Analog power 2.2V for AUDIO CODEC	-
AVDD28_AUDIO	P		Analog power 2.8V for AUDIO CODEC	-
AVDD22_XO	P		Analog power 2.2V for Crystal Oscillator	-
AVDD22_XO_32K	P		Analog power 2.2V for 32K output clock buffer	-
Digital power				
DVDD18_IO0	P		Digital power input for IO	-
DVDD18_IO1	P		Digital power input for IO	-
DVDD18_IO2	P		Digital power input for IO	-
DVDD18_IO3	P		Digital power input for IO	-
DVDD18_EFUSE	P		Digital power input for efuse IO	-
DVDD28_MSDC0	P		Digital power input for 1.8V/3.3V MSDC0/NAND flash IO	-
DVDD28_DPI	P		Digital power input for 3.3V DPI IO	-
DVDD28_NFI	P		Digital power input for 3.3V NAND Flash IO	-
DVDD28_MSDC1	P		Digital power input for 1.8/3.3V MSDC IO	-
DVDD28_MSDC2	P		Digital power input for 1.8/3.3V MSDC IO	-
VCCIO	P		Digital power input for 1.24V/1.39V/2.5V EMI	-
VCCK	P		Digital power input for core	-
VCCK_VPROC	P		Digital power input for processor	-
Analog ground				
AVSS18_PLLGP	G		Analog ground for PLL	-
AVSS18_AP	G		Analog ground for AuxADC, TSENSE	-
AVSS22_XO	G		Analog ground for Crystal Oscillator	-
AVSS22_XO_32K	G		Analog ground for 32K output clock buffer	-
AVSS_CONN	G		Analog ground for connectivity RF	-

Pin name	Type	DI/DO/DIO type	Description	Power domain
AVSS18_HDMITX	G		Analog ground for HDMI/MHL tranmitter	-
AVSS18_MIPI	G		Analog ground for MIPI	-
AVSS_AUDIO	G		Analog ground for Audio Codec	-
AVSS33_USB	G		Analog ground for USB	-
Digital ground				
GNDK	G		Digital ground	-

This product can’t support Camera, HDMI, MIPI DSI and CSI, DSP_PWM, LCM_RST, DPI, the related pin you can only as the GPI or GPIO function.

2.1.4 Interface Application Notice

Table 2-16: Interface Application Notice

Interface Types	Total Sets	Interface with Non-MTK IC	Constraints
I2C	3	Y	Max speed 1 MHz
PWRAP SPI (PMIC SPI)	1	N	
SPI interface	1	Y	N/A
UART	3	Y	N/A
Key pad	1	Y	
USB2.0	2	Y	N/A
Nand Flash/MSDC0/eMMC	1	Y	
MSDC1/SPI NOR	1	Y	
MSDC2 (SDIO)	1	Y	Need external GPIO for sleep wake-up
Ethernet MII/RMII	1	Y	N/A
Audio I2S 8-CH Output	1	Y	Only master mode
Audio I2S 2-CH Input	1	Y	Slave mode max input rate support <= 48K
Audio PCM	1	Y	8K/16K/32K
Audio TDM (RX)	1	Y	Master mode only Support one data pin for multi-channel input (max channel=8)

2.2 Electrical Characteristics

2.2.1 Absolute Maximum Ratings

Table 2-17: Absolute maximum ratings for power supply

Symbol or Pin name	Description	Min.	Max.	Unit
AVDD18_PLLGP	Analog power input 1.8V for PLL	1.7	1.9	V
AVDD18_AP	Analog power input 1.8V for AUXADC, TSENSE	1.7	1.9	V
AVDD18_WBT	Analog power 1.8V for WBT RF	1.7	1.9	V
AVDD18_WBT_AFE	Analog power 1.8V for WBT AFE	1.7	1.9	V
AVDD33_WBT	Analog power 3.5V (default) for WBT TX PA and IQM	3.3	3.6	V

Symbol or Pin name	Description	Min.	Max.	Unit
AVDD18_ANA	Analog power	1.7	1.9	V
AVDD33_USB	Analog power 3.3V for USB	3.135	3.465	V
AVDD18_USB	Analog power 1.8V for USB	1.7	1.9	V
AVDD18_MEMPLL	Analog power for MEMPLL	1.7	1.9	V
AVDD22_AUDIO	Analog power 2.2V for AUDIO CODEC	2.1	2.3	V
AVDD28_AUDIO	Analog power 2.8V for AUDIO CODEC	2.7	2.9	V
AVDD22_XO	Analog power 2.2V for Crystal Oscillator	2.1	2.3	V
AVDD22_XO_32K	Analog power 2.2V for 32K output clock buffer	2.1	2.3	V
DVDD18_IO0	Digital power input for IO	1.7	1.9	V
DVDD18_IO1	Digital power input for IO	1.7	1.9	V
DVDD18_IO2	Digital power input for IO	1.7	1.9	V
DVDD18_IO3	Digital power input for IO	1.7	1.9	V
DVDD18_EFUSE	Digital power input for efuse IO	1.8	2.2	V
DVDD28_MII	Digital power input for MII IO	1.7	3.63	V
DVDD28_NFI	Digital power input for NAND flash IO	1.7	3.63	V
DVDD28_MSDC0	Digital power input for EMMC/NAND flash IO	1.7	3.63	V
DVDD28_MSDC1	Digital power input for MSDC1 IO	1.7	3.63	V
DVDD28_MSDC2	Digital power input for MSDC2 IO	1.7	3.63	V
VCCIO	Digital power input for EMI	1.14	1.575	V
VCCK	Digital power input for core	0.765	1.31	V
VCCK_VPROC	Digital power input for CPU	0.765	1.31	V

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

2.2.2 Recommended Operating Conditions

Table 2-18: Recommended operating conditions for power supply

Symbol or pin name	Description	Min.	Typ.	Max.	Unit
AVDD18_PLLGP	Analog power input 1.8V for PLL	1.7	1.8	1.9	V
AVDD18_AP	Analog power input 1.8V for AUXADC, TSENSE	1.7	1.8	1.9	V
AVDD18_WBT	Analog power 1.8V for WBT RF	1.7	1.8	1.9	V
AVDD18_WBT_AFE	Analog power 1.8V for WBT AFE	1.7	1.8	1.9	V
AVDD33_WBT	Analog power 3.5V (default) for WBT TX PA and IQM	3.3	3.5	3.6	V
AVDD18_ANA	Analog power	1.7	1.8	1.9	V

Symbol or pin name	Description	Min.	Typ.	Max.	Unit
AVDD33_USB	Analog power 3.3V for USB	3.135	3.3	3.465	V
AVDD18_USB	Analog power 1.8V for USB	1.7	1.8	1.9	V
AVDD18_MEMPLL	Analog power for MEMPLL	1.7	1.8	1.9	V
AVDD22_AUDIO	Analog power 2.2V for AUDIO CODEC	2.1	2.2	2.3	V
AVDD28_AUDIO	Analog power 2.8V for AUDIO CODEC	2.7	2.8	2.9	V
AVDD22_XO	Analog power 2.2V for Crystal Oscillator	2.1	2.2	2.3	V
AVDD22_XO_32K	Analog power 2.2V for 32K output clock buffer	2.1	2.2	2.3	V
DVDD18_IO0	Digital power input for IO	1.7	1.8	1.9	V
DVDD18_IO1	Digital power input for IO	1.7	1.8	1.9	V
DVDD18_IO2	Digital power input for IO	1.7	1.8	1.9	V
DVDD18_IO3	Digital power input for IO	1.7	1.8	1.9	V
DVDD18_EFUSE	Digital power input for efuse IO	1.8	2.0	2.2	V
DVDD28_MII	Digital power input for MII IO	1.7	1.8	1.9	V
		2.52	2.8	3.08	
		2.97	3.3	3.63	
DVDD28_NFI	Digital power input for NAND flash IO	1.7	1.8	1.9	V
		2.52	2.8	3.08	
		2.97	3.3	3.63	
DVDD28_MSDC0	Digital power input for EMMC/NAND flash IO	1.7	1.8	1.9	V
		2.52	2.8	3.08	
		2.97	3.3	3.63	
DVDD28_MSDC1	Digital power input for MSDC1 IO	1.7	1.8	1.9	V
		2.52	2.8	3.08	
		2.97	3.3	3.63	
DVDD28_MSDC2	Digital power input for MSDC2 IO	1.7	1.8	1.9	V
		2.52	2.8	3.08	
		2.97	3.3	3.63	
VCCIO	Digital power input for EMI (DDR4)	1.14	1.2	1.26	V
	Digital power input for EMI (DDR3L)	1.283	1.35	1.42	
	Digital power input for EMI (DDR3)	1.425	1.5	1.575	
	Digital power input for EMI (LPDDR2/3)	1.14	1.2	1.26	
VCCK	Digital power input for core	0.765	1.15	1.31	V
VCCK_VPROC	Digital power input for processor	0.765	1.15	1.31	V

2.2.3 Storage Conditions

- Shelf life in sealed bag: 12 months at < 40°C and < 90% relative humidity (RH).
- After bag opened, devices subjected to infrared reflow, vapor-phase reflow, or equivalent processing must be:
- Mounted within 168 hours at factory conditions of 30°C/60% RH, or
- Stored at 20% RH.
- Devices require baking before mounting, if:
- Humidity Indicator Card is >20% when read at 23°C+/- 5°C or
- 2a or 2b is not met.
- If baking is required, devices may be baked for:
- 192 hours at 40°C+5°C/- 0°C and < 5% RH for low temperature device containers, or
- 24 hours at 125°C+5°C/- 0°C for high temperature device containers.

2.2.4 AC Electrical Characteristics and Timing Diagram

2.2.4.1 External Memory Interface for DDR3

The external memory interface shown below is used to connect DDR3 device for MT8516A. It includes pins ED_CLK, ED_CLK_B, RESET_B, ECKE, ECS#, EWR#, ERAS#, ECAS#, EDS[3:0], EDS#[3:0], EA[15:0] and ED[31:0].

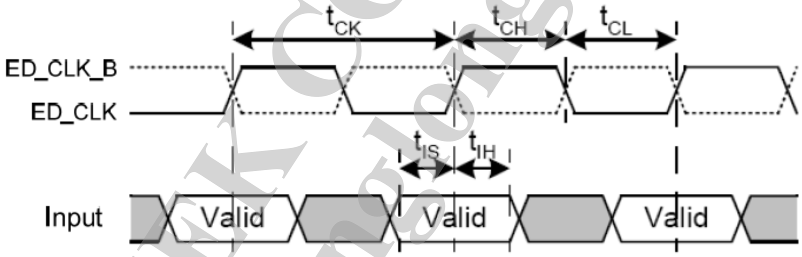


Figure 2-2: Basic timing parameter for DDR3 command

Table 2-19: DDR3 AC timing parameter table of external memory interfaces

Symbol	Description	Min.	Typ.	Max.	Unit
tDQSCK	DQS output access time from CK/CK'	-0.225		0.225	ns
tCK	Clock cycle time	1.25		1.875	ns
tCH	Clock high level width	0.47		0.53	tCK
tCL	Clock low level width	0.47		0.53	tCK
tDS	DQ & DM input setup time	0.01			ns
tDH	DQ & DM input hold time	0.045			ns
tIS	Address & control input setup time	0.045			ns
tIH	Address & control input hold time	0.12			ns

tLZ	DQ & DQS low-impedance time from CK/CK'	-0.45		0.225	ns
tHZ	DQ & DQS high-impedance time from CK/CK'			0.225	ns
tDQSQ	DQS-DQ skew			0.1	ns
tQH	DQ/DQS output hold time from DQS	0.38			tCK
tDQSH	DQS input high-level width	0.45		0.55	tCK
tDQSL	DQS input low-level width	0.45		0.55	tCK
tMRD	MODE register set command period	4			tCK
tRPRE	Read preamble	0.9			tCK
tRPST	Read postamble	0.3			tCK
tRAS	ACTIVE to PRECHARGE command period	28		9*REFI	tCK
tRC	ACTIVE to ACTIVE command period	36			tCK
tRFC	AUTO REFRESH to ACTIVE/AUTO REFRESH command period	300ns (4gb)			ns
tRCD	ACTIVE to READ or WRITE delay	8			tCK
tRP	PRECHARGE command period	8			tCK
tRRD	ACTIVE bank A to ACTIVE bank B delay	4			tCK
tWR	WRITE recovery time	6			tCK
tWTR	Internal write to READ command time	6			tCK
tXSDLL	Exit Self Refresh to command requiring a locked DLL	512			tCK
tXPDLL	Exit power down with DLL frozen to commands requiring a locked DLL	20			tCK
tXP	EXIT power down with DLL to next valid command delay	6			tCK
tCKE	CKE min. pulse width(high & low pulse width)	4			tCK
tREFI	Average periodic refresh interval	3.9			us

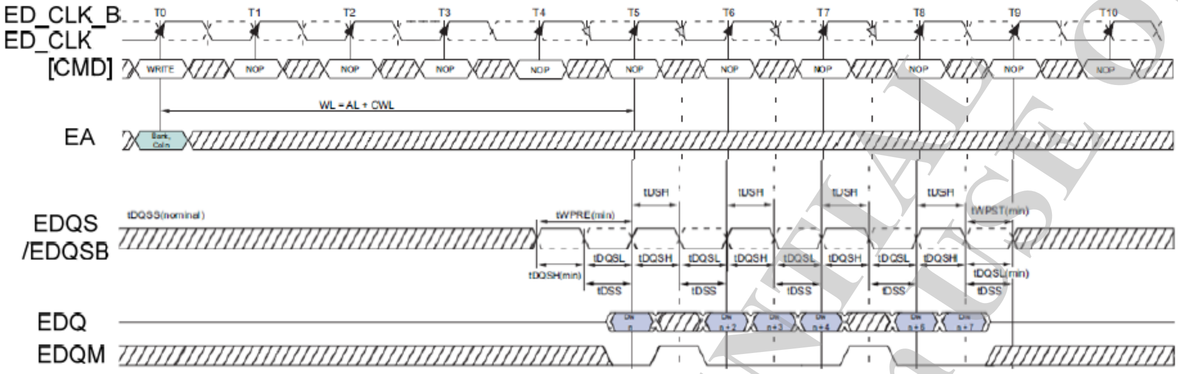


Figure 2-3: Basic Timing Parameter for DDR3 Write

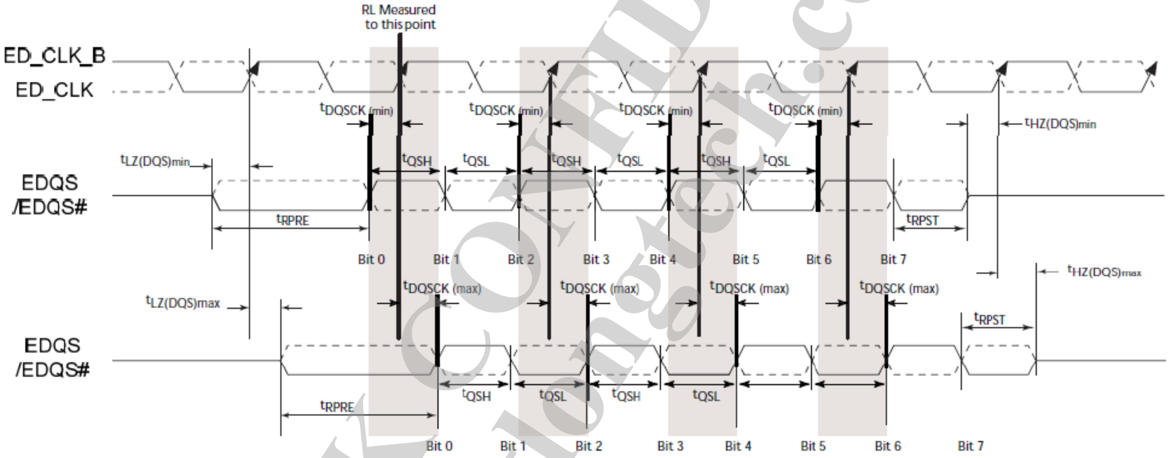
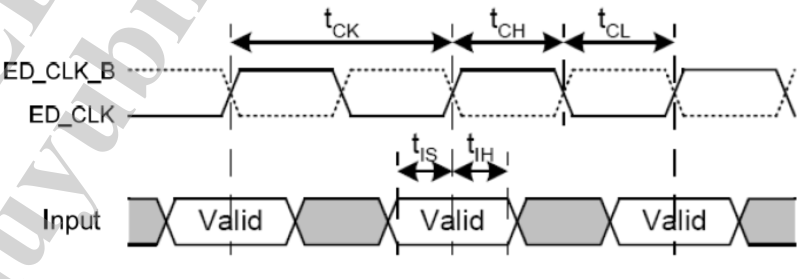


Figure 2-4: Basic Timing Parameter for DDR3 Read

2.2.4.2 External Memory Interface for DDR4

The external memory interface shown below is used to connect DDR4 device for MT8516A. It includes pins ED_CLK, ED_CLK_B, RESET_B, ECKE, EACT, ECS#, EWR#, ERAS#, ECAS#, EDQM[3:0], EDQS[3:0],EDQS#[3:0], EA[13:0] and ED[31:0].



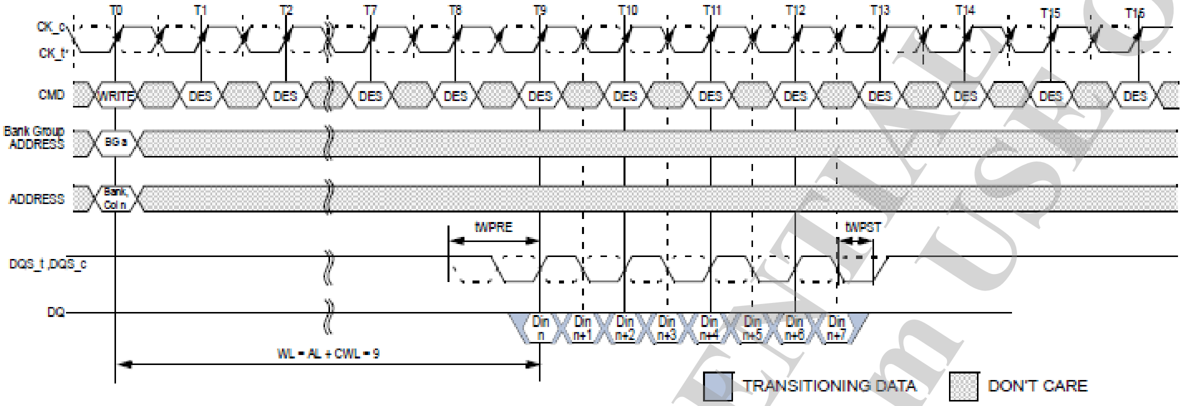
Input = EA0~EA13, ECKE, ECS#, EWR#, ERAS#, and ECAS#

Figure 2-5: Basic timing parameter for DDR4 commands

Table 2-20: DDR4 AC timing parameter table of external memory interfaces

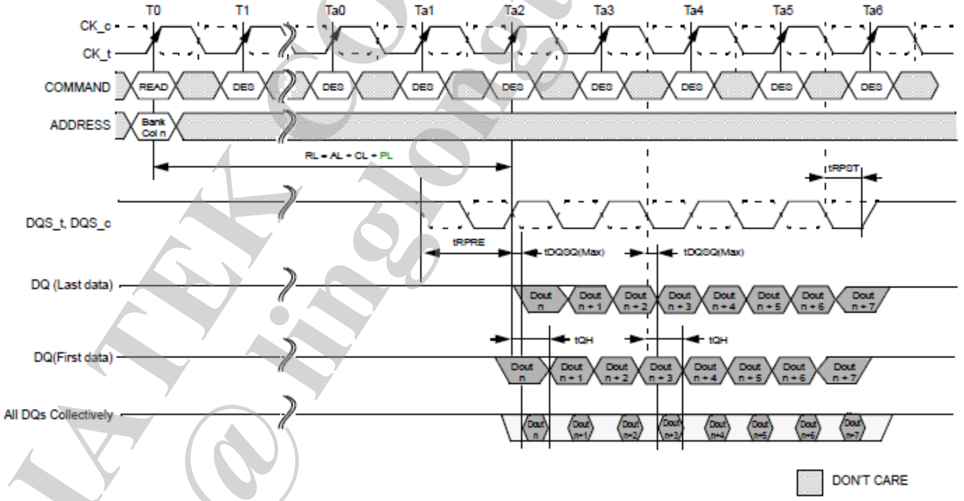
Symbol	Description	Min.	Typ.	Max.	Unit
tDQSCK	DQS output access time from CK/CK'	-0.225		0.225	ns
tCK	Clock cycle time	1.25		1.875	ns
tCH	Clock high level width	0.48		0.52	tCK
tCL	Clock low level width	0.48		0.52	tCK
tDS	DQ & DM input setup time	0.1875			ns
tDH	DQ & DM input hold time	0.1875			ns
tIS	Address & control input setup time	0.115			ns
tIH	Address & control input hold time	0.14			ns
tLZ	DQ & DQS low-impedance time from CK/CK'	-0.45		0.225	ns
tHZ	DQ & DQS high-impedance time from CK/CK'			0.225	ns
tDQSQ	DQS-DQ skew			0.16	ns
tQH	DQ/DQS output hold time from DQS	0.74			tCK
tDQSH	DQS input high-level width	0.38		0.62	tCK
tDQSL	DQS input low-level width	0.38		0.62	tCK
tMRD	MODE register set command period	8			tCK
tRPRE	Read preamble	0.9			tCK
tRPST	Read postamble	0.33			tCK
tRAS	ACTIVE to PRECHARGE command period	28		9*REFI	tCK
tRC	ACTIVE to ACTIVE command period	38			tCK
tRFC	AUTO REFRESH to ACTIVE/AUTO REFRESH command period	300ns (4gb)			ns
tRCD	ACTIVE to READ or WRITE delay	10			tCK
tRP	PRECHARGE command period	10			tCK
tRRD	ACTIVE bank A to ACTIVE bank B delay	4			tCK
tWR	WRITE recovery time	6			tCK
tWTR-S	Internal write to READ command time(different bank group)	2			tCK
tWTR-L	Internal write to READ command time(same bank group)	6			tCK
tXSDLL	Exit Self Refresh to command requiring a locked DLL	597			tCK
tXP	EXIT power down with DLL to next valid command delay	6			tCK
tCKE	CKE min. pulse width(high & low pulse width)	4			tCK
tCCD_S	CAS_n-to-CAS_n command delay to different bank group	4			tCK
tCCD_L	CAS_n-to-CAS_n command delay to same bank group	5			tCK

Symbol	Description	Min.	Typ.	Max.	Unit
tREFI	Average periodic refresh interval			7.8	us



NOTE :
1. BL = 8 ,WL = 9, AL = 0, CWL = 9, Preamble = 1tCK
2. DIN n = data-in to column n.
3. DES commands are shown for ease of illustration; other commands may be valid at these times.
4. BL8 setting activated by either MR0[A1:A0 = 0:0] or MR0[A1:A0 = 0:1] and A12 = 1 during WRITE command at T0.
5. CA Parity = Disable, CS to CA Latency = Disable, Write DBI = Disable.

Figure 2-6: Basic timing parameter for DDR4 write



NOTE 1 BL = 8, AL = 0, CL = 11, Preamble = 1tCK
NOTE 2 DOUT n = data-out from column n.
NOTE 3 DES commands are shown for ease of illustration; other commands may be valid at these times.
NOTE 4 BL8 setting activated by either MR0[A1:0 = 00] or MR0[A1:0 = 01] and A12 = 1 during READ command at T0.
NOTE 5 Output timings are referenced to VDDQ, and DLL on for locking.
NOTE 6 tDQSQ defines the skew between DQS_t,DQS_c to Data and does not define DQS_t, DQS_c to Clock.
NOTE 7 Early Data transitions may not always happen at the same DQ. Data transitions of a DQ can vary (either early or late) within a burst

Figure 2-7: Basic timing parameter for DDR4 read

2.2.4.3 External Memory Interface for LPDDR2

The external memory interface shown below is used to connect LPDDR2 device for MT8516A. It includes pins ED_CLK, ED_CLK_B, RESET_B, ECKE, ECS#, EDQS[3:0],EDQS#[3:0], EA[9:0] and ED[31:0].

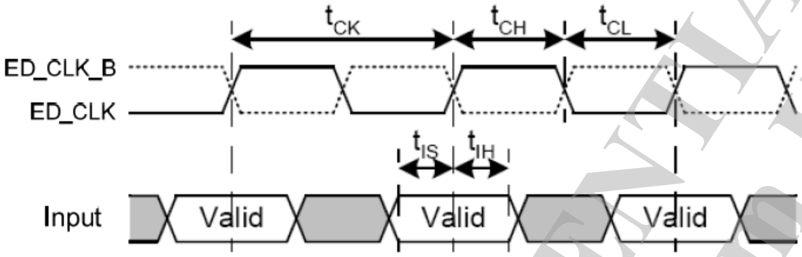


Figure 2-8: Basic timing parameter for LPDDR2 commands

Table 2-21: LPDDR2 AC timing parameter table of external memory interfaces

Symbol	Description	Min.	Typ.	Max.	Unit
tDQSCK	DQS output access time from CK/CK'	2.5		5.5	ns
tCK	Clock cycle time	1.87		10	ns
tCH	Clock high level width	0.45		0.55	tCK
tCL	Clock low level width	0.45		0.55	tCK
tDS	DQ & DM input setup time	0.21			ns
tDH	DQ & DM input hold time	0.21			ns
tIS	Address & control input setup time	0.22			ns
tIH	Address & control input hold time	0.22			ns
tLZ	DQ & DQS low-impedance time from CK/CK'	2.178			ns
tHZ	DQ & DQS high-impedance time from CK/CK'			5.4	ns
tDQSQ	DQS-DQ skew			0.2	ns
tQH	DQ/DQS output hold time from DQS	0.48			tCK
tDQSH	DQS input high-level width	0.4			tCK
tDQSL	DQS input low-level width	0.4			tCK
tMRD	MODE register set command period	5			tCK
tRPRE	Read preamble	0.9			tCK
tRPST	Read postamble	0.38			tCK
tRAS	ACTIVE to PRECHARGE command period	23			tCK
tRC	ACTIVE to ACTIVE command period	34			tCK
tRFC	AUTO REFRESH to ACTIVE/AUTO REFRESH command period	70			tCK
tRCD	ACTIVE to READ or WRITE delay	10			tCK
tRP	PRECHARGE command period	10			tCK

Symbol	Description	Min.	Typ.	Max.	Unit
tRRD	ACTIVE bank A to ACTIVE bank B delay	6			tCK
tWR	WRITE recovery time	8			tCK
tWTR	Internal write to READ command time	4			tCK
tXP	EXIT power down to next valid command delay	4			tCK
tCKE	CKE min. pulse width(high & low pulse width)	3			tCK
tREFI	Average periodic refresh interval	3.9			us

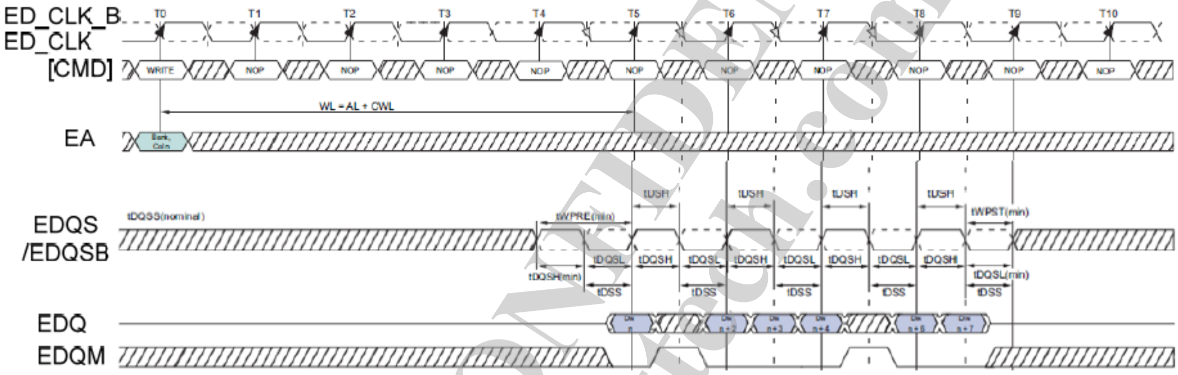


Figure 2-9: Basic timing parameter for LPDDR2 write

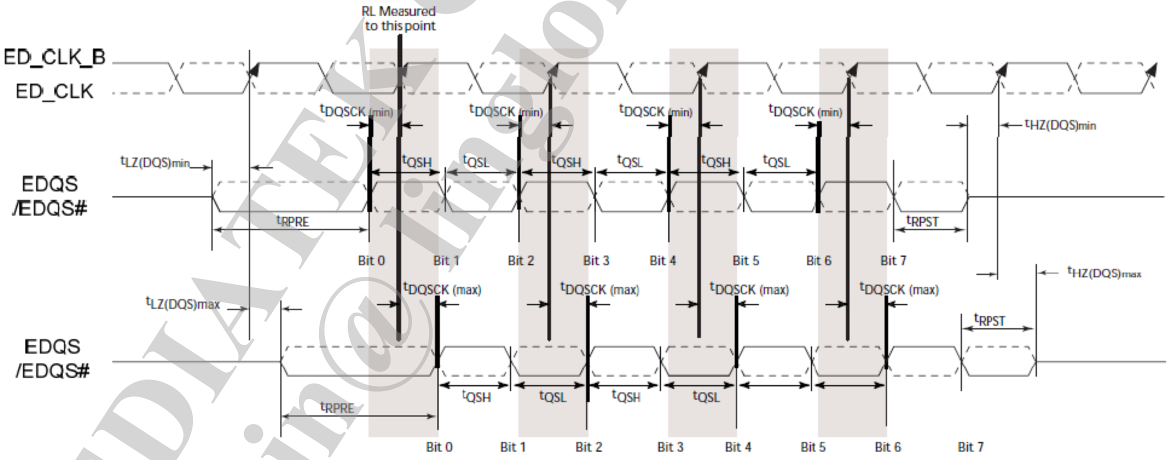


Figure 2-10: Basic timing parameter for LPDDR2 read

2.2.4.4 External Memory Interface for LPDDR3

The external memory interface shown below is used to connect LPDDR3 device for MT8516A. It includes pins ED_CLK, ED_CLK_B, RESET_B, ECKE, ECS#, EDQS[3:0],EDQS#[3:0], EA[9:0] and ED[31:0].

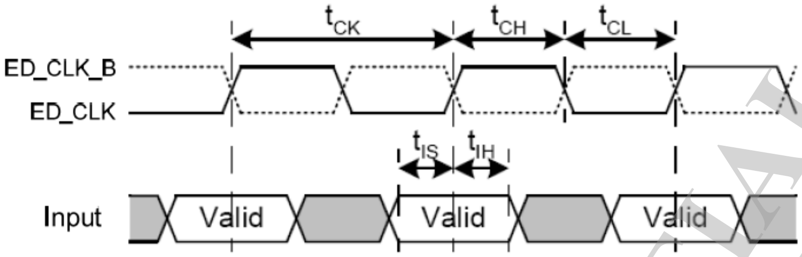


Figure 2-11: Basic timing parameter for LPDDR3 commands

Table 2-22: LPDDR3 AC timing parameter table of external memory interfaces

Symbol	Description	Min.	Typ.	Max.	Unit
tDQSCK	DQS output access time from CK/CK'	2.5		5.5	ns
tCK	Clock cycle time	1.25		10	ns
tCH	Clock high level width	0.45		0.55	tCK
tCL	Clock low level width	0.45		0.55	tCK
tDS	DQ & DM input setup time	0.15			ns
tDH	DQ & DM input hold time	0.15			ns
tIS	Address & control input setup time	0.25			ns
tIH	Address & control input hold time	0.25			ns
tLZ	DQ & DQS low-impedance time from CK/CK'	2.2			ns
tHZ	DQ & DQS high-impedance time from CK/CK'			5.4	ns
tDQSQ	DQS-DQ skew			0.135	ns
tQH	DQ/DQS output hold time from DQS	0.4			tCK
tDQSH	DQS input high-level width	0.4			tCK
tDQSL	DQS input low-level width	0.4			tCK
tMRW	MODE register set command period	10			tCK
tMRR	MODE register set command period	4			tCK
tRPRE	Read preamble	0.9			tCK
tRPST	Read postamble	0.3			tCK
tRAS	ACTIVE to PRECHARGE command period	18			tCK
tRC	ACTIVE to ACTIVE command period	27			tCK
tRFC	AUTO REFRESH to ACTIVE/AUTO REFRESH command period	84			tCK
tRCD	ACTIVE to READ or WRITE delay	8			tCK
tRP	PRECHARGE command period	10			tCK
tRRD	ACTIVE bank A to ACTIVE bank B delay	4			tCK

tWR	WRITE recovery time	6			tCK
tWTR	Internal write to READ command time	4			tCK
tXP	EXIT power down to next valid command delay	3			tCK
tCKE	CKE min. pulse width(high & low pulse width)	6			tCK
tREFI	Average periodic refresh interval	3.9			us

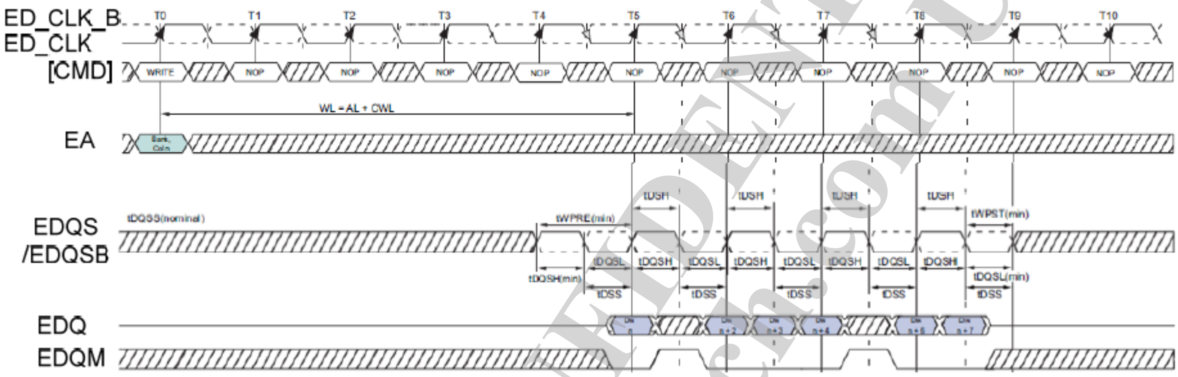


Figure 2-12: Basic timing parameter for LPDDR3 write

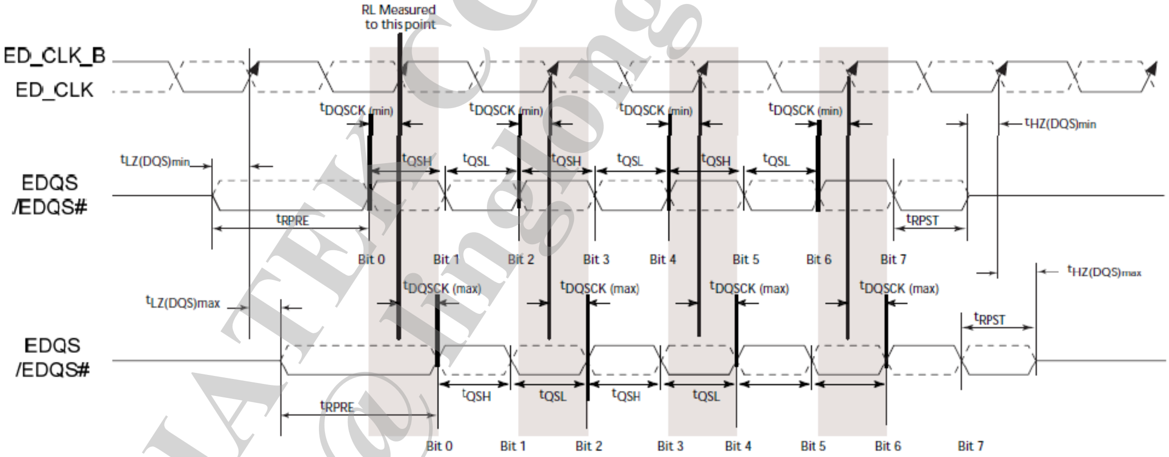


Figure 2-13: Basic timing parameter for LPDDR3 read

Table 2-23: DDR parameter requirements at component pin

Parameter	Comment	Min	Typ	Max	Unit
Per-bit deskew availability			N/A		Y/N
On-Chip dynamic skew span between DQ/DQS	WRITE mode			40	ps
On-Chip dynamic skew span between CMD/CLK				40	ps

Parameter	Comment	Min	Typ	Max	Unit
On-Chip static skew within DQ byte	If per-bit deskew is not available			40	ps
On-Chip static skew within CA bus	If training is not available			40	ps
Max allowed DQ/DQS byte skew span	READ mode, if per-bit deskew is not available			100	ps
Max allowed DQ/DQS single-bit skew span	READ mode, if per-bit deskew is available			165	ps
Required Teye (Aperture-based)	DQ READ, skew between DQ/DQS			120	ps
Required VIH DC/VIL DC			DDR3L: 90 DDR4:75 LP3:100		mV
Required VIH AC/VIL AC			DDR3L: 135 DDR4:100 LP3:150		mV
Max allowed overshoot/undershoot value	See Figure 2-14		DDR3L: 0.4 DDR4:0.3 LP3:0.35		V
Max allowed overshoot/undershoot area	See Figure 2-14		DDR3L: 0.33 DDR4:0.25 LP3:0.1		V*ns

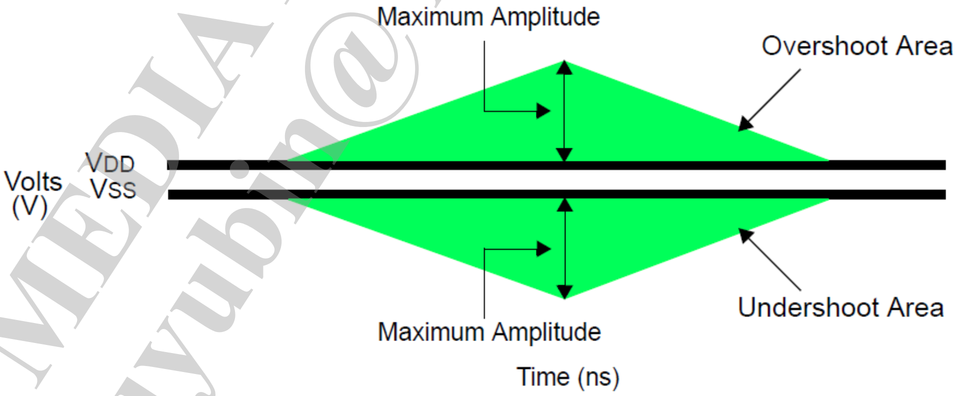


Figure 2-14: Control Overshoot and Undershoot Definition Block

Table 2-24: DDR3 skew tolerances

Length Matching Channel (Board +Package)	Tolerance
Impedance	
DQ-DQS	700 mil
DQ-DQ	700 mil
DQS-CLK	1200 mil
CMD-CLK	200 mil
CMD- CMD (within the same DRAM)	600 mil
CMD- CMD (across DRAMs)	200 mil
CTRL-CLK	200 mil
CTRL-CTRL (within same DRAM)	600 mil
CTRL-CTRL (across DRAMs)	200 mil
CLK-CLK	200 mil

2.2.4.5 I2C Parameter Specification

Table 2-25: I2C parameter specification

Parameter	Description	Min	Typ	Max	Unit
VOL	Output voltage Low	0		0.2VDD[1]	V
VOH	Output voltage High			1.01VDD	V
Trise	Rise time of SDA and SCL signals	0		120	ns
Tfall	Fall time of SDA and SCL signals	6.5[2]		120	ns
Thigh	Pulse duration, SCL high			0.26	µs
Tlow	Pulse duration, SCL low			0.5	µs
TSU	Setup time, SDA to SCL	0.05			µs
TST,STA	Setup time, SCL to start condition	0.26			µs
THD,STA	Hold time, start condition to SCL	0.26			µs
TST,STO	Setup time, SCL to stop condition	0.26			µs
T(BUF)	Bus free time between stop and start condition	0.5			µs
Cb[3]	Capacitive load for each bus line			550	pF

VDD is 1.8V.

For E1: I2C IO can meet spec minimum value: 20 ×(VDD / 5.5 V)=6.5 request， For E2 I2C IO will modify to improve for resistance and C loading range adjustment.

The maximum bus capacitance allowable may vary from this value depending on the actual operating voltage and frequency of the application.

2.2.4.6 eMMC Parameter Specification

Table 2-26: eMMC parameter specification

Parameter			Min	Typ	Max	Unit
Clock	Period	HS200	5		-	ns
		DDR50	10		-	ns
		SDR50	20		-	ns
		SDR25	40		-	ns
	Timing	Trise			1.0	ns
		Tfall			1.0	ns
		Duty Cycle	30/45		70/55	%
Timing	HS200	Setup Time	1.4			ns
		Hold Time	0.8			ns
	SDR25,50	Setup Time	3			ns
		Hold Time	3			ns
Voltage	Supply	I/O Supply	2.5			V
	Input	High	0.9*VCC3IO		VCC3IO+0.3	V

Parameter			Min	Typ	Max	Unit
		Low	-0.3		0.1*VCC3IO	V
	Output	High	1.4		VCC3IO+0.3	V
		Low	-0.3		0.35*VCC3IO	V

2.2.4.7 SD Parameter Specification

Table 2-27: SD parameter specification

Parameter			Min	Typ	Max	Unit
Clock	Period	SDR104	4.8			ns
		SDR50	10			ns
		SDR25	20			ns
		SDR12	38			ns
	Timing	Trise	-		0.96	ns
		Tfall	-		0.96	ns
		Duty Cycle	30		70	%
Timing	SDR104	Setup Time	1.4		-	ns
		Hold Time	0.8		-	ns
	SDR12,25,50	Setup Time	3.0		-	ns
		Hold Time	0.8		-	ns
Voltage	Supply	I/O Supply				V
	Input	High	1.27		VCC3IO+03	V
		Low	-0.3		0.58	V
	Output	High	1.4		VCC3IO+03	V
		Low	-0.3		0.45	V

2.3 System Configuration

2.3.1 Constant Tie Pins

Table 2-28: Constant tied pins of MT8516A

Pin name	Description
TESTMODE	Test mode (tie to GND)
FSOURCE_P	EFUSE blowing (tie to GND)

2.4 Power-on Sequence

The power-on/off sequence with XTAL is shown in the following figure:

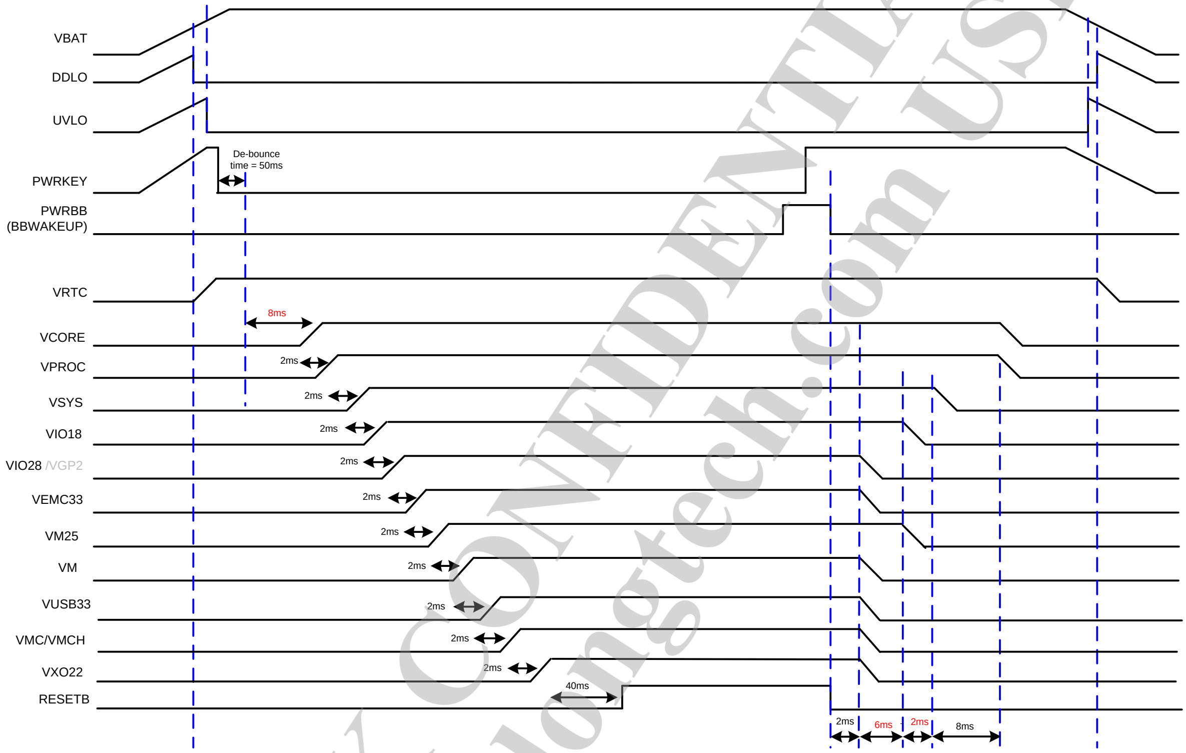


Figure 2-15: Power on/off sequence with XTAL

Figure below shows the power-on/off sequence without XTAL. VXO22 is always turned on when VBAT is above the DDLO threshold.

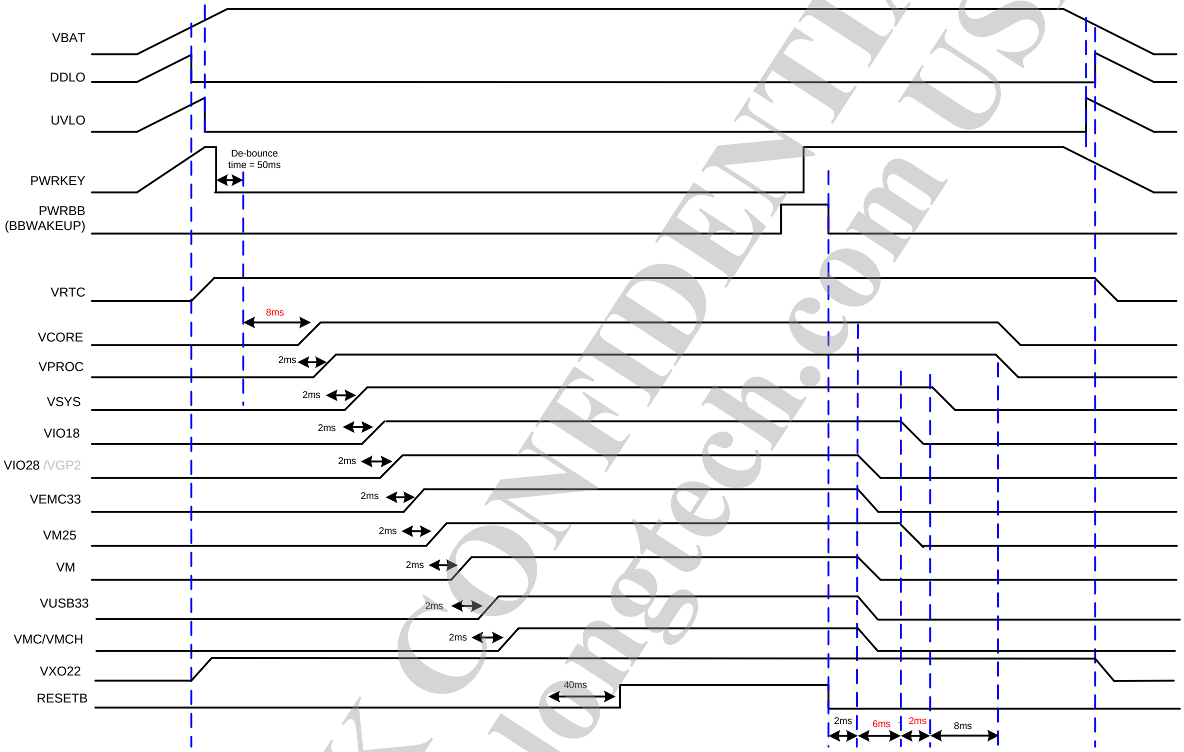


Figure 2-16: Power on/off sequence without XTAL

2.5 Analog Baseband

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. During the writing or reading 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. An analog block includes the following analog functions for the complete application processor:

- Auxiliary ADC: Provides an ADC for the battery and other auxiliary analog functions monitoring.
- Clock generation: PLLs providing clock signals to MCU, USB, MSDC units.

2.5.2 Features

The analog blocks include the following analog functions for complete application processor:

- AUXADC
- Phase locked loop
- Temperature sensor
- AUDIO CODEC

2.5.3 Block Diagram

2.5.3.1 AUXADC

2.5.3.1.1 Block Descriptions

The auxiliary ADC 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 measuring and some for external voltage measuring. 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.

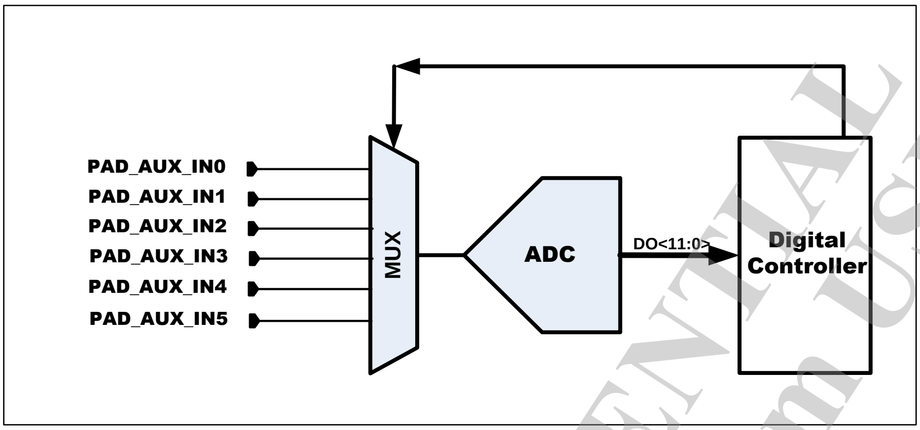


Figure 2-17: AUXADC Block Diagram

Table 2-29: 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	Internal use(AUDIO)
Channel 6	Internal use(AUDIO)
Channel 7	Internal use(AUDIO)
Channel 8	Internal use(AUDIO)
Channel 9	External use (AUX_IN2)
Channel 10	Internal use(Thermal Sensor)
Channel 11	Internal use(Thermal Sensor)
Channel 12	External use (AUX_IN3)
Channel 13	External use (AUX_IN4)
Channel 14	External use (AUX_IN5)
Channel 15	Internal use(ACC_DET)

2.5.3.1.2 Functional Specifications

See the table below for the functional specifications of auxiliary ADC.

Table 2-30: AUXADC specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
N	Resolution		12		Bit
FC	Clock rate		4		MHz
FS	Sampling rate @ N-Bit		4/(N+4)		MSPS
	Input swing	0		1.5	V
CIN	Input capacitance Unselected channel Selected channel		50 4		fF pF
RIN	Input resistance Unselected channel	400			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 (1kHz full swing input & 1.0833MHz clock rate)	62	68		dB
DVDD	Digital power supply	0.945	1.15	1.31	V
AVDD	Analog power supply	1.7	1.8	1.9	V
	Current consumption Power-up Power-down		535 15		uA uA

2.5.3.2 Phase Locked Loop

2.5.3.2.1 Block Descriptions

There are total 8 PLLs in PLL macro, providing several clocks for CPU, BUS, MSDC and image-sensor.

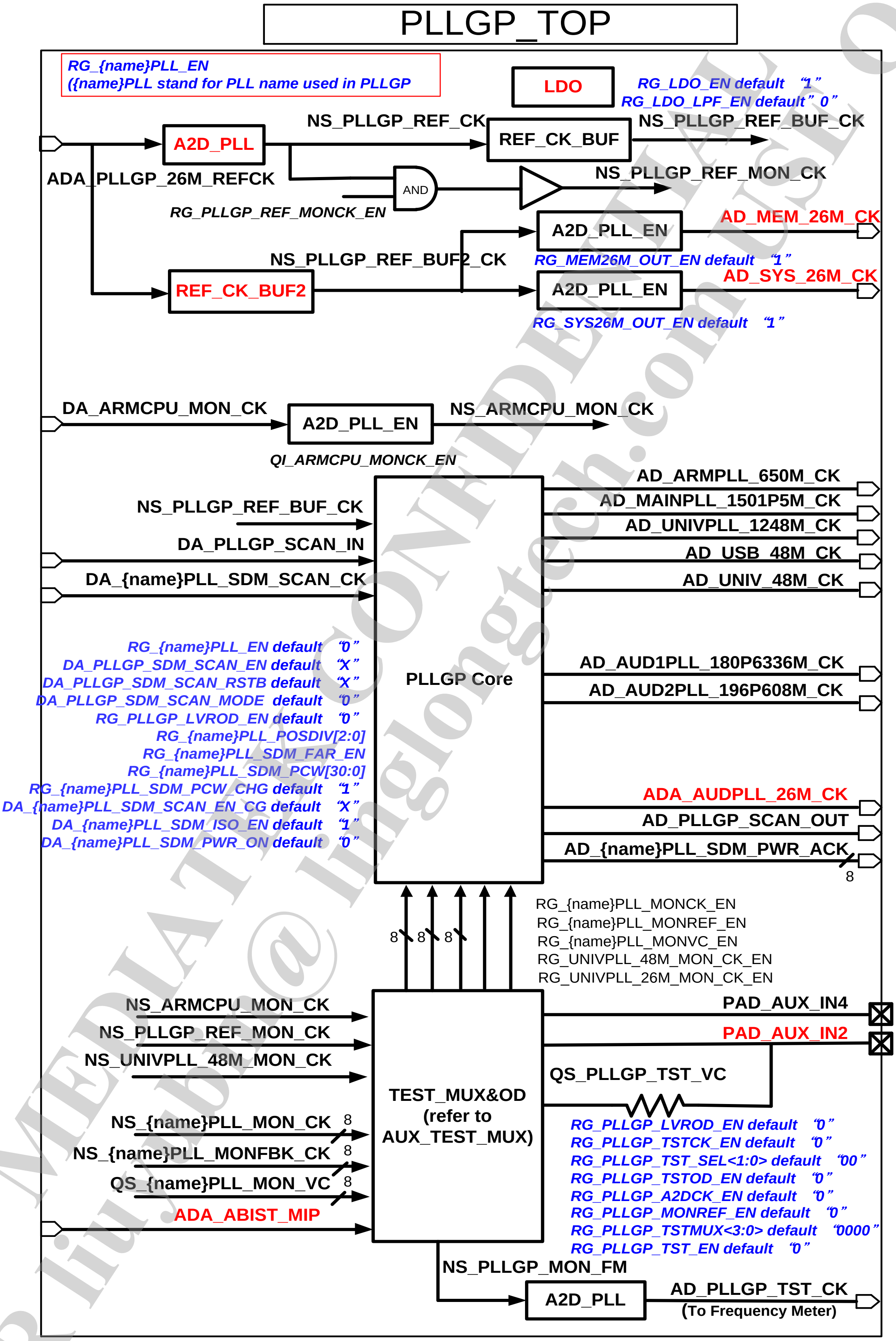

PLLGP_TOP
RG_{name}PLL_EN
({name}PLL stand for PLL name used in PLLGP
LDO
RG_LDO_EN default "1"
RG_LDO_LPF_EN default" 0"
A2D_PLL
NS_PLLGP_REF_CK
REF_CK_BUF
NS_PLLGP_REF_BUF_CK
ADA_PLLGP_26M_REFCK
AND
NS_PLLGP_REF_MON_CK
RG_PLLGP_REF_MONCK_EN
A2D_PLL_EN
AD_MEM_26M_CK
NS_PLLGP_REF_BUF2_CK
RG_MEM26M_OUT_EN default "1"
REF_CK_BUF2
A2D_PLL_EN
AD_SYS_26M_CK
RG_SYS26M_OUT_EN default "1"
DA_ARMCPU_MON_CK
A2D_PLL_EN
NS_ARMCPU_MON_CK
QI_ARMCPU_MONCK_EN
NS_PLLGP_REF_BUF_CK
DA_PLLGP_SCAN_IN
DA_{name}PLL_SDM_SCAN_CK
AD_ARMPLL_650M_CK
AD_MAINPLL_1501P5M_CK
AD_UNIVPLL_1248M_CK
AD_USB_48M_CK
AD_UNIV_48M_CK
RG_{name}PLL_EN default "0"
DA_PLLGP_SDM_SCAN_EN default "X"
DA_PLLGP_SDM_SCAN_RSTB default "X"
DA_PLLGP_SDM_SCAN_MODE default "0"
RG_PLLGP_LVROD_EN default "0"
RG_{name}PLL_POSDIV[2:0]
RG_{name}PLL_SDM_FAR_EN
RG_{name}PLL_SDM_PCW[30:0]
RG_{name}PLL_SDM_PCW_CHG default "1"
DA_{name}PLL_SDM_SCAN_EN_CG default "X"
DA_{name}PLL_SDM_ISO_EN default "1"
DA_{name}PLL_SDM_PWR_ON default "0"
PLLGP Core
AD_AUD1PLL_180P6336M_CK
AD_AUD2PLL_196P608M_CK
ADA_AUDPLL_26M_CK
AD_PLLGP_SCAN_OUT
AD_{name}PLL_SDM_PWR_ACK
8
8
8
8
RG_{name}PLL_MONCK_EN
RG_{name}PLL_MONREF_EN
RG_{name}PLL_MONVC_EN
RG_UNIVPLL_48M_MON_CK_EN
RG_UNIVPLL_26M_MON_CK_EN
NS_ARMCPU_MON_CK
NS_PLLGP_REF_MON_CK
NS_UNIVPLL_48M_MON_CK
NS_{name}PLL_MON_CK
8
NS_{name}PLL_MONFBK_CK
8
QS_{name}PLL_MON_VC
8
ADA_ABIST_MIP
TEST_MUX&OD
(refer to
AUX_TEST_MUX)
PAD_AUX_IN4
PAD_AUX_IN2
QS_PLLGP_TST_VC
RG_PLLGP_LVROD_EN default "0"
RG_PLLGP_TSTCK_EN default "0"
RG_PLLGP_TST_SEL<1:0> default "00"
RG_PLLGP_TSTOD_EN default "0"
RG_PLLGP_A2DCK_EN default "0"
RG_PLLGP_MONREF_EN default "0"
RG_PLLGP_TSTMUX<3:0> default "0000"
RG_PLLGP_TST_EN default "0"
NS_PLLGP_MON_FM
A2D_PLL
AD_PLLGP_TST_CK
(To Frequency Meter)

Figure 2-18: PLL Block Diagram

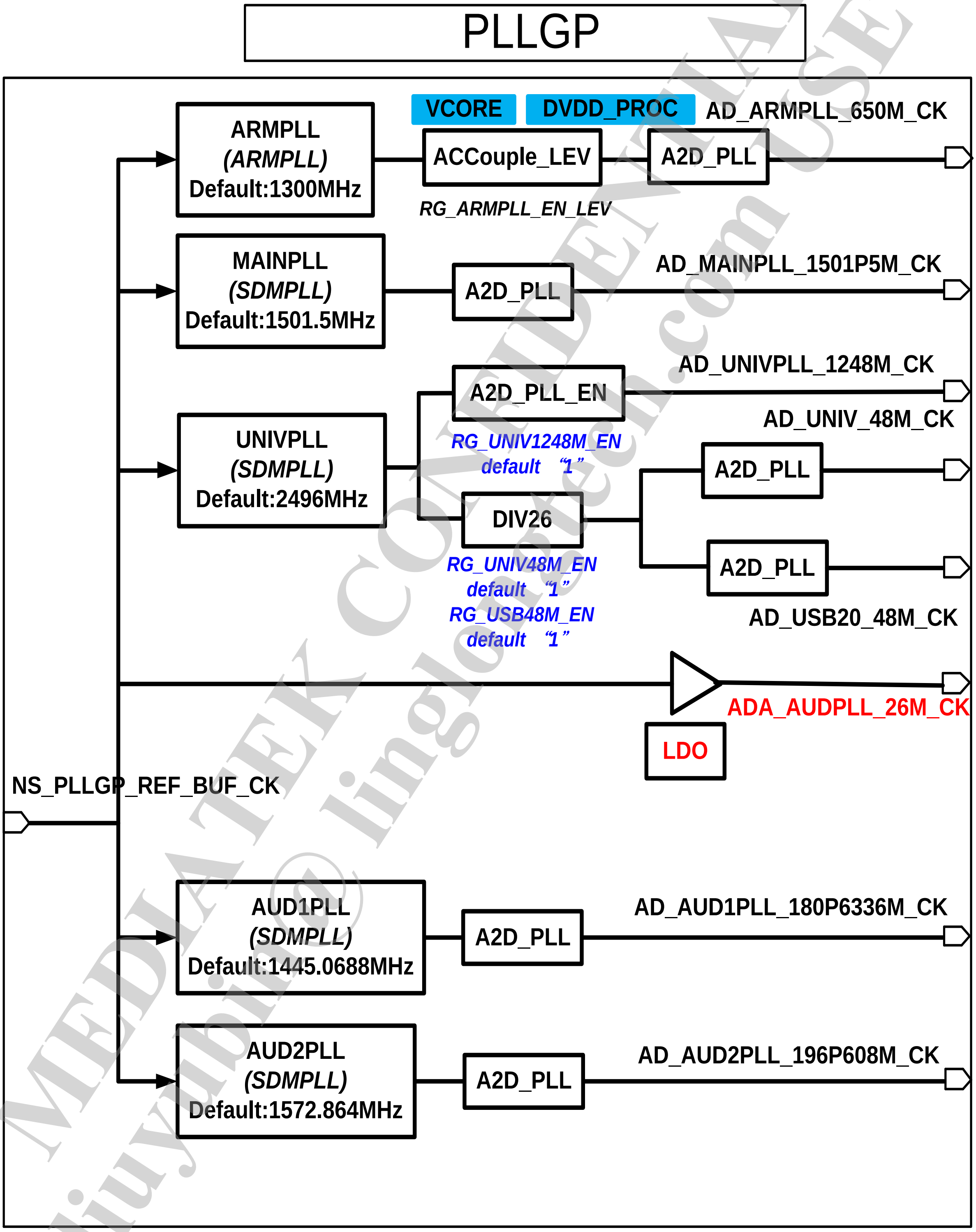


Figure 2-19: PLL Core Block Diagram

2.5.3.2.2 Functional Specifications

See the table below for the functional specifications of PLL.

Table 2-31: 26M Reference specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
Fin	Input clock frequency		26		MHz

Fout	Output clock frequency		26(System)
			26(MEM)
			26(AUDPLL)
					MHz

	Settling time		20		us
	Output clock duty cycle	45	50	55	%
	Output clock jitter (period jitter)		60		ps
DVDD	Digital power supply	0.945	1.15	1.31	V
AVDD	Analog power supply	1.7	1.8	1.9	V
	Current consumption		1.2		mA
	Power-down current consumption			12	uA

Table 2-32: ARMPLL specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
Fin	Input clock frequency		26		MHz
Fout	Output clock frequency		650		MHz
	Settling time		20		us
	Output clock duty cycle	45	50	55	%
	Output clock jitter (period jitter)		60		ps
DVDD	Digital power supply	0.945	1.15	1.31	V
AVDD	Analog power supply	1.7	1.8	1.9	V
	Current consumption		1.2		mA
	Power-down current consumption			12	uA

Table 2-33: MAINPLL specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
Fin	Input clock frequency		26		MHz
Fout	Output clock frequency		1501.5		MHz
	Settling time		20		us
	Output clock duty cycle	45	50	55	%

Symbol	Parameter	Min.	Typ.	Max.	Unit
	Output clock jitter (period jitter)		30		ps
DVDD	Digital power supply	0.945	1.15	1.31	V
AVDD	Analog power supply	1.7	1.8	1.9	V
	Current consumption		1.2		mA
	Power-down current consumption			12	uA

Table 2-34: UNIVPLL specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
Fin	Input clock frequency		26		MHz

Fout	Output clock frequency	N/A	1248
			48(USB)
			48(UNIV)
				N/A	MHz

	Settling time		20		us
	Output clock duty cycle	47	50	53	%
	Output clock jitter (period jitter)		< 30ps P-P for 1248M < 60ps P-P for 48M		ps
DVDD	Digital power supply	0.945	1.15	1.31	V
AVDD	Analog power supply	1.7	1.8	1.9	V
	Current consumption		0.8		mA
	Power-down current consumption			12	uA

Table 2-35: AUD1PLL specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
Fin	Input clock frequency		26		MHz
Fout	Output clock frequency		180.6336		MHz
	Settling time		20		us
	Output clock duty cycle	47	50	53	%
	Output clock jitter (period jitter)		100		ps
DVDD	Digital power supply	0.945	1.15	1.31	V
AVDD	Analog power supply	1.7	1.8	1.9	V
	Current consumption		1.2		mA
	Power-down current consumption			12	uA

Table 2-36: AUD2PLL specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
Fin	Input clock frequency		26		MHz
Fout	Output clock frequency		196.608		MHz
	Settling time		20		us
	Output clock duty cycle	47	50	53	%
	Output clock jitter (period jitter)		100		ps
DVDD	Digital power supply	0.945	1.15	1.31	V
AVDD	Analog power supply	1.7	1.8	1.9	V
	Current consumption		1.2		mA
	Power-down current consumption			12	uA

2.5.3.3 Temperature Sensor

2.5.3.3.1 Block Descriptions

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

2.5.3.3.2 Functional Specifications

See the table below for the functional specifications of temperature sensor.

Table 2-37: Temperature sensor specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
	Resolution		0.15		°C
	Temperature range	0		105	°C
	Accuracy	-7		7	°C
	Active current		300		uA
	Quiescent current		3		uA

2.5.3.4 AUDIO CODEC

2.5.3.4.1 Block Descriptions

The audio uplink path is composed of PGA and audio ADC. There are three input pairs of the uplink path to support dual-MIC, earphone-MIC and digital MIC. The audio downlink is composed of stereo audio DACs, stereo headphone drivers and lineout driver. The necessary MIC bias voltages and multi-key accessory detection are also provided by this completed audio codec. The Audio Downlink includes the following blocks: DAC and headphone driver, there are 2 Channels to support stereo headphone; and a voice amplifier lineout to drive off-chip speaker amplifier.

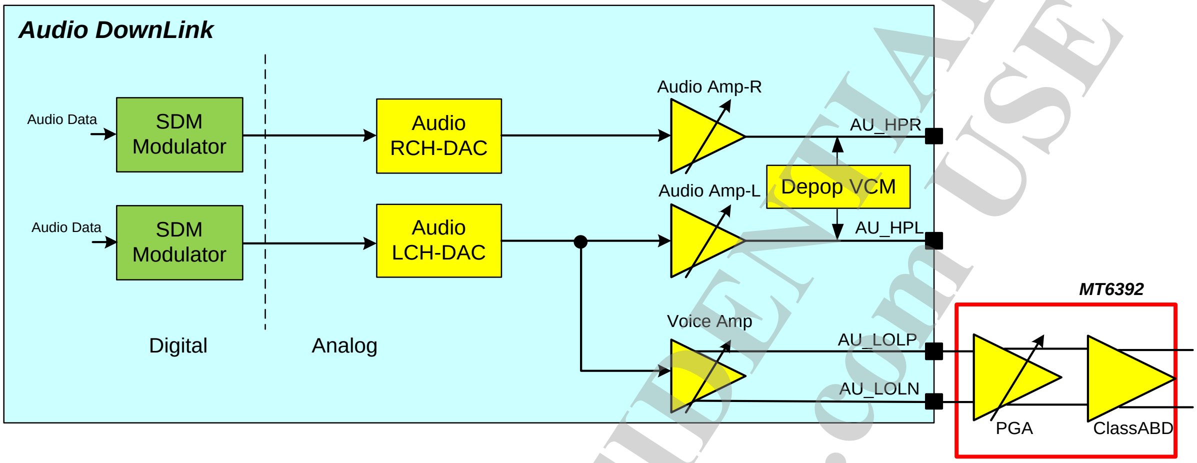


Figure 2-20: Audio Downlink Block Diagram

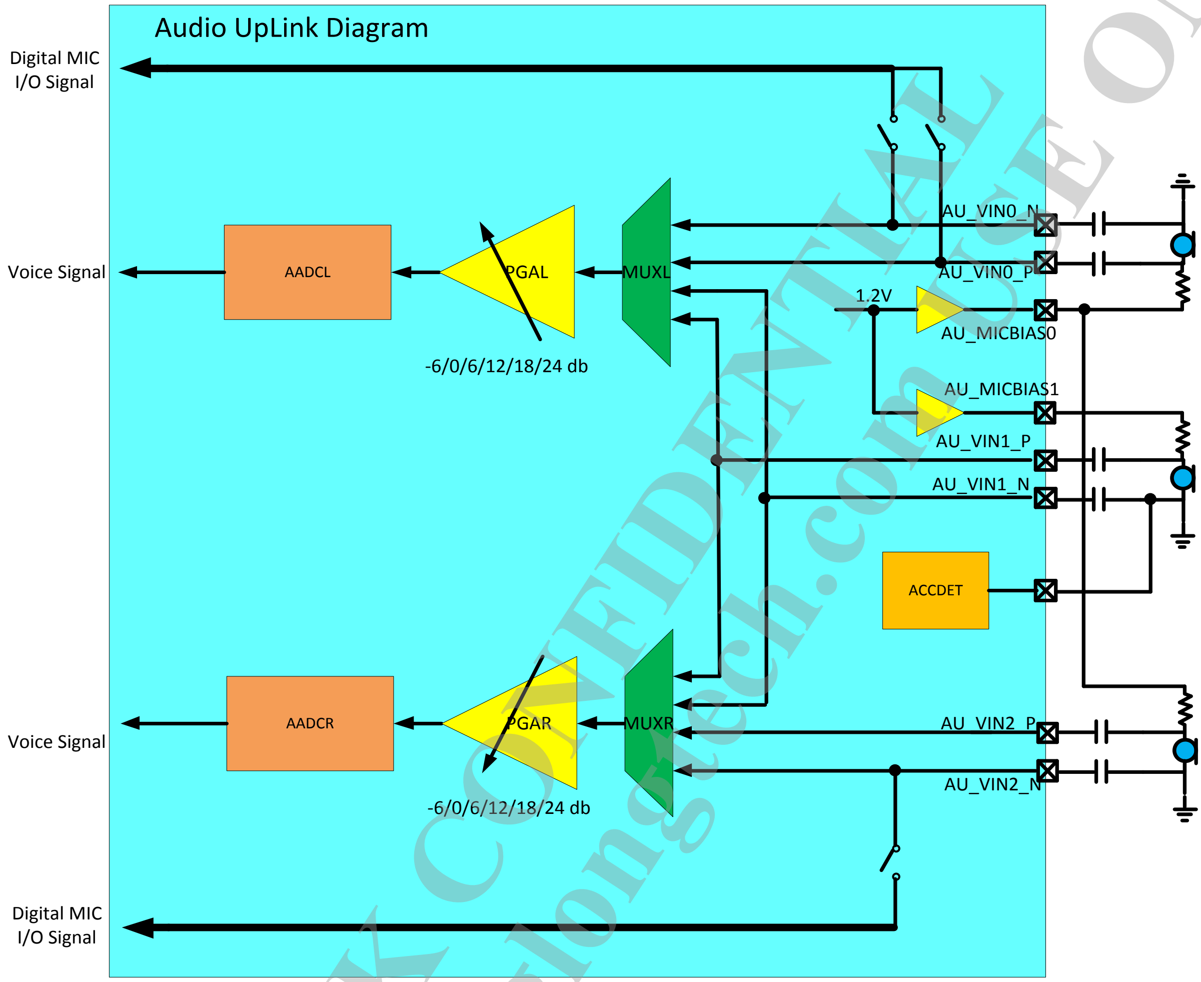


Figure 2-21: Audio Uplink Block Diagram

2.5.3.4.2 Functional Specifications

The analog blocks include the following analog functions for complete application processor:

- Audio Downlink
- stereo headphone drivers
- Lineout driver
- Audio Uplink
- Dual MIC/earphone MIC/digital MIC
- multi-key accessory detection
- MIC bias voltages

Table 2-38: Audio Downlink and Uplink specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
	2.8V Analog Power(V28)	2.7	2.8	2.9	V
	2.2V Analog Power(V22)	2.1	2.2	2.3	V
	Digital Power Supply(V10)	0.945	1.15	1.31	V
AUDIO DownLink, AUDIOLINK SPEC					
	Clock Frequency (FCK)		6.5		MHz
	Sample Rate (Fs)	32	44.1	48	KHz
	Current Consumption (IDC)		11		mA
	Peak Signal to Noise Ratio (PSNR) HP AMP Gain=0dB; @All zeros fed to DAC Input		90		dB
	Dynamic Range (DR) HP AMP Gain=0dB; @-60dBFS Input		90		dB
	Output Swing for 0dBFS Input Level			0.85	Vrms
	Total Harmonic Distortion (THD) Plus Noise 11 mW@ 0dBFS, 64Ω		-83	-70	dB
	Output Resistor Load(Single-ended)	64	132		Ohm
	Output Capacitor Load			250	pF
	L-R Channel Crosstalk (XT)		92		dB
AUDIO DownLink, VoiceLINK SPEC					
	Peak Signal to Noise Ratio (PSNR) Lineout Amp Gain=4dB; @All zeros fed to DAC Input		91		dB
	Dynamic Range (DR) Lineout Amp Gain=4dB; @-60dBFS Input		91		dB
	Output Swing for 0dBFS Input Level			1.273	Vrms
	THD+N Total Harmonic Distortion Plus Noise @ 0dBFS,12KΩ		-83		dB
	Output Resistor Load(Differential)	12			Ohm
	Peak Signal to Noise Ratio (PSNR) Lineout Amp Gain=4dB; @All zeros fed to DAC Input		91		dB

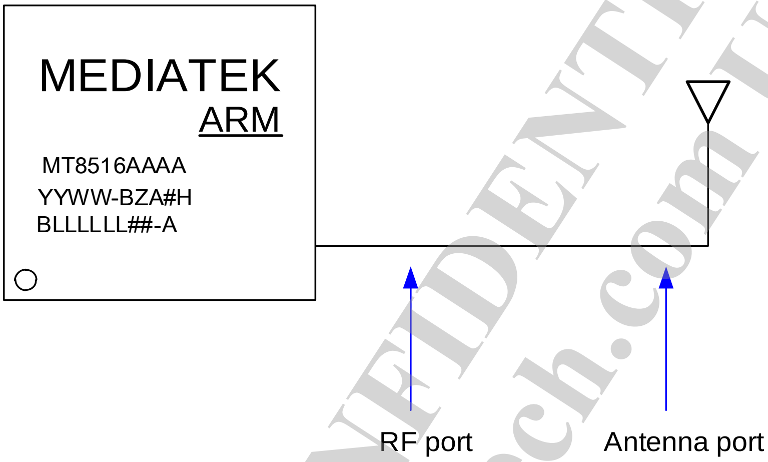
Analog Uplink MIC Path					
	Current Consumption (1 channel)		2		mA
	Total Harmonic Distrotion+Noise(THD+N) Input Level : -60dBm0(PGA gain=0dB) Input Level : 0dBm0(PGA gain=0dB)		-25 -82		dB dB
	Input Impedance(Differential)	13	20	27	KOhm
	L-R Channel Crosstalk(XT)		95		dB
Digital MIC Path					
	DMIC Clock Frequency		1.625/3.25		MHz
	DMIC Clock Duty Cycle	40		60	%
	DMIC Clock Rise time(Max CL=80p)		10		ns
	DMIC Clock Fall time(Max CL=80p)		10		ns
	Sample Rate(FS)	8	16	32/48	KHz
Audio Uplink MICBIAS					
	Microphone0 Biasing Voltage	1.9		2.2	V
	Microphone1 Biasing Voltage	1.9		2.5	V
	Current draw from microphone bias		2		mA

For DMIC 48K Hz: Must use 3.25M DMIC clock PDM to PCM conversion swing limitation at 48K mode, When digital swing < -12dBFS, PDM to PCM conversion output would be boosted 12dB When PCM's digital swing >= -12dBFS, it will cause saturation in recorded PCM data

Frequency response passband ripple (only for digital filter) 0~20K : 2dB， 0~8K : 0.4dB

2.6 Connectivity RF Characteristic

The WLAN radio characteristics are described in this section where the RF port and antenna port of MT8516A can be directly connected by a 50Ohm trace.



2.6.1 Wi-Fi RF Radio Characteristic

The WLAN radio characteristics are described in this section. Unless otherwise specified, all specifications are measured at the chip output RF port.

2.6.1.1 Wi-Fi Receiver Specification

Note: The specification value is valid at room temperature (25oC).

Table 2-39: 2.4GHz receiver specificatio

Parameter	Description	Min.	Typ.	Max.	Unit
Frequency range		2,412	-	2,484	MHz
RX sensitivity [a]	1 Mbps DSSS		-95		dBm
	2 Mbps DSSS		-93		dBm
	5.5 Mbps DSSS		-91		dBm
	11 Mbps DSSS		-88		dBm
RX Sensitivity [a]	6 Mbps OFDM		-91.5		dBm
	9 Mbps OFDM		-90		dBm
	12 Mbps OFDM		-89		dBm
	18 Mbps OFDM		-86.5		dBm
	24 Mbps OFDM		-83.5		dBm
	36 Mbps OFDM		-80		dBm

Parameter	Description	Min.	Typ.	Max.	Unit
	48 Mbps OFDM		-76		dBm
	54 Mbps OFDM		-74.5		dBm
RX sensitivity [b]	MCS 0		-91.5		dBm
BW = 20MHz	MCS 1		-88		dBm
Green field	MCS 2		-86		dBm
800nS guard interval	MCS 3		-83		dBm
Non-STBC	MCS 4		80		dBm
	MCS 5		-75.5		dBm
	MCS 6		-74		dBm
	MCS 7		-72.5		dBm
RX sensitivity	MCS 0		-88.5		dBm
BW = 40MHz	MCS 1		-85		dBm
Green field	MCS 2		-83		dBm
800nS guard interval	MCS 3		-80		dBm
Non-STBC	MCS 4		-77		dBm
	MCS 5		-72.5		dBm
	MCS 6		-71		dBm
	MCS 7		-69		dBm
Maximum receive level	11 Mbps DSSS			-5	dBm
	6 Mbps OFDM			-10	dBm
	54 Mbps OFDM			-10	dBm
	MCS0			-10	dBm
	MCS7			-10	dBm
Adjacent channel rejection (30MHz offset)	1 Mbps DSSS			40	dB
Adjacent channel rejection (25MHz offset)	11 Mbps DSSS			40	dB
Adjacent channel rejection (25MHz offset)	6 Mbps OFDM			34	dB
	54 Mbps OFDM			22	dB
Adjacent channel rejection (25MHz offset), BW = 20MHz	MCS 0			25	dB
	MCS 7			5	dB
Adjacent channel rejection (40MHz offset), BW = 40MHz	MCS 0			26	dB
	MCS 7			1	dB
Blocking level for 1dB RX sensitivity degradation	776 ~ 794 MHz CDMA2000				dBm
	824 ~ 849 MHz GSM				dBm
	880 ~ 915 MHz GSM				dBm

Parameter	Description	Min.	Typ.	Max.	Unit
	1,710 ~ 1,785 MHz GSM				dBm
	1,850 ~ 1,910 MHz GSM				dBm
	1,850 ~ 1,910 MHz WCDMA				dBm
	1,920 ~ 1,980 MHz WCDMA				dBm

a: Degraded by 1.5dB at 850C

b: Sensitivity degradation in different MCS modes: mixed-mode normal GI: 1dB, mixed-mode short GI: 1dB, and STBC:1dB

2.6.1.2 Wi-Fi Transmitter Specification

Note:

The specification value is valid at room temperature (250C).

All specifications are measured at the RF port unless otherwise specified.

Typical output power degradation around 3dB at FCC band edge channels

Table 2-40: 2.4GHz transmitter specification

Parameter	Description	Min.	Typ.	Max.	Unit
Frequency range		2,412	-	2,484	MHz
Output power	802.11b, 1~11 Mbps DSSS		19		dBm
VCN35=3.5V	802.11g, 6 ~36Mbps OFDM		16.5		dBm
	802.11g, 48 ~54Mbps OFDM		16		dBm
	802.11n, HT20 MCS0~4		16.5		dBm
	802.11n, HT20 MCS5~7		15.5		dBm
	802.11n, HT40 MCS0~4		15.5		dBm
	802.11n, HT40 MCS5~7		14		dBm
EVM	802.11b, 1~11 Mbps DSSS @Pout=19dBm		25		%
	802.11g, 6 ~36Mbps OFDM@Pout=16.5dBm			-19	dB
	802.11g, 48 ~54Mbps OFDM@Pout=16dBm		-28		dB
	802.11n, HT20 MCS0~4@Pout=16.5dBm			-19	dB
	802.11n, HT20 MCS5~7@Pout=15.5dBm		-28		dB
	802.11n, HT40 MCS0~4@Pout=15.5dBm			-19	dB

Parameter	Description	Min.	Typ.	Max.	Unit
	802.11n, HT40 MCS5~7@Pout=14dBm		-30		dB
TX power accuracy	-20~65 oC,5~22dBm			±1.5	dB
Loadpull variation at VSWR = 2:1	Output power variation			±1.5	dB
	EVM degradation		4		dB
Transmitted power (Data rate = 6M, Pout = 17dBm)	76 ~ 108 MHz		-142		dBm/Hz
	776 ~ 794 MHz		-142		dBm/Hz
	869 ~ 960 MHz		-142		dBm/Hz
	925 ~ 960 MHz		-142		dBm/Hz
	1,570 ~ 1,580 MHz		-140		dBm/Hz
	1,805 ~ 1,880 MHz		-131		dBm/Hz
	1,930 ~ 1,990 MHz		-126		dBm/Hz
	2,110 ~ 2,170MHz		-125		dBm/Hz
Harmonic output power (Data rate = 1M, Pout = 19dBm) i	2nd harmonic			-43	dBm/MHz
	3rd harmonic			-43	dBm/MHz

2.6.2 Bluetooth RF Radio Characteristics

2.6.2.1 Basic Data Rate Receiver Specification

Table 2-41: Basic data rate receiver specification

Parameter	Description	Min.	Typ.	Max.	Unit
Frequency range		2,402		2,480	MHz
Receiver sensitivity	BER < 0.1%		-92		dBm
Max. usable signal	BER < 0.1%	-20	-5		dBm
C/I co-channel	Co-channel selectivity (BER < 0.1%)	-	6	11	dB
C/I 1MHz	Adjacent channel selectivity (BER < 0.1%)	-	-7	0	dB
C/I 2MHz	2nd adjacent channel selectivity (BER < 0.1%)	-	-39	-30	dB
C/I ≧3MHz	3rd adjacent channel selectivity (BER < 0.1%)	-	-43	-40	dB
C/I image channel	Image channel selectivity (BER < 0.1%)	-	-20	-9	dB
C/I image 1MHz	1MHz adjacent to image channel selectivity (BER < 0.1%)	-	-35	-20	dB
Out-of-band blocking*	30MHz to 2,000MHz	-10			dBm
	2,001MHz to 2,339MHz	-27			dBm
	2,501MHz to 3,000MHz	-27			dBm
	3,001MHz to 12.75GHz	-10			dBm
Intermodulation	Max. interference level to maintain 0.1% BER	-39			dBm

2.6.2.2 Basic Data Rate Transmitter Specification

Table 2-42: Basic data rate transmitter specification

Parameter	Description	Min.	Typ.	Max.	Unit
Frequency range		2,402	-	2,480	MHz
Output power	At max power output level		6		dBm
Power control step		2	4	8	dB
ICFT	Initial carrier frequency drift	-75	±18	75	kHz
Carrier frequency drift	One slot packet (DH1)	-25	±15	25	kHz
	Three slot packet (DH3)	-40	±15	40	kHz
	Five slot packet (DH5)	-40	±15	40	kHz
	Max. drift rate	-20	10	20	kHz/50us
Modulation characteristic	$\Delta f1_{avg}$	140	157	175	kHz

Parameter	Description	Min.	Typ.	Max.	Unit
	$\triangle f2_{max}$ (for at least 99% of all $\Delta f2_{max}$)	115	145	-	kHz
	$\triangle f2_{avg}/\Delta f1_{avg}$	0.8	0.98	-	
20-dB bandwidth		-	922	1,000	kHz
In-band spurious emission	±2MHz offset		-38	-20	dBm
	±3MHz offset		-43	-40	dBm
	>±3MHz offset		-43		dBm
Out-of-band spurious emission**	30MHz to 1GHz			-36	dBm
	1GHz to 12.75GHz			-30	dBm
	1.8GHz to 1.9GHz			-47	dBm
	5.15 to 5.3GHz			-47	dBm

2.6.2.3 Enhanced Data Rate Receiver Specification

Table 2-43: Enhanced data rate receiver specification

Parameter	Description	Min.	Typ.	Max.	Unit
Frequency range		2,402	-	2,480	MHz
Receiver sensitivity	π/4 DQPSK (BER < 0.01%)	-	-91	-70	dBm
	8PSK (BER < 0.01%)	-	-85.5	-70	dBm
Max. usable signal	π/4 DQPSK (BER < 0.1%)	-20	-5	-	dBm
	8PSK (BER < 0.1%)	-20	-5	-	dBm
C/I co-channel	π/4 DQPSK (BER < 0.1%)	-	9	13	dB
	8PSK (BER < 0.1%)	-	16	21	dB
C/I 1MHz	π/4 DQPSK (BER < 0.1%)	-	-12	0	dB
	8PSK (BER < 0.1%)	-	-6	5	dB
C/I 2MHz	π/4 DQPSK (BER < 0.1%)	-	-36	-30	dB
	8PSK (BER < 0.1%)	-	-33	-25	dB
C/I ≧3MHz	π/4 DQPSK (BER < 0.1%)	-	-43	-40	dB
	8PSK (BER < 0.1%)	-	-40	-33	dB
C/I image channel	π/4 DQPSK (BER < 0.1%)	-	-20	-7	dB
	8PSK (BER < 0.1%)	-	-15	0	dB
C/I image 1MHz	π/4 DQPSK (BER < 0.1%)	-	-40	-20	dB
	8PSK (BER < 0.1%)	-	-30	-13	dB

2.6.2.4 Enhanced Data Rate Transmitter Specification

Table 2-44: Enhanced data rate transmitter specification

Parameter	Description		Min.	Typ.	Max.	Unit
Frequency range			2,402		2,480	MHz
Output power	π/4 DQPSK			3		dBm
	8PSK			3		dBm
Relative transmit power	π/4 DQPSK		-4	-1.7	1	dB
	8PSK		-4	-1.7	1	dB
Frequency stability	ω₀	π/4 DQPSK	-10	±4	10	kHz
		8PSK	-10	±4	10	kHz
	ωᵢ	π/4 DQPSK	-75	±20	75	kHz
		8PSK	-75	±20	75	kHz
	|ω₀+ωᵢ|	π/4 DQPSK	-75	±20	75	kHz
		8PSK	-75	±20	75	kHz
Modulation accuracy	RMS DEVM	π/4 DQPSK	-	8	20	%
		8PSK	-	8	13	%
	99% DEVM	π/4 DQPSK	-	12	30	%
		8PSK	-	12	20	%
	Peak DEVM	π/4 DQPSK	-	17	35	%
		8PSK	-	17	25	%
In-band spurious emission	±1MHz offset	π/4 DQPSK		-29	-26	dB
		8PSK		-29	-26	dB
	±2MHz offset	π/4 DQPSK		-23	-20	dBm
		8PSK		-23	-20	dBm
	±3MHz offset	π/4 DQPSK		-42	-40	dBm
		8PSK		-42	-40	dBm

2.6.2.5 LE Receiver Specification

Table 2-45: Bluetooth LE receiver specification

Parameter	Description	Min.	Typ.	Max.	Unit
Frequency range		2,402		2,480	MHz
Receiver sensitivity (*)	PER < 30.8%		-95	-70	dBm
Max. usable signal	PER < 30.8%	-20	-5		dBm
C/I co-channel	Co-channel selectivity (PER < 30.8%)		6	21	dB
C/I 1MHz	Adjacent channel selectivity (PER < 30.8%)		-7	15	dB
C/I 2MHz	2nd adjacent channel selectivity		-30	-17	dB
	(PER < 30.8%)				
	3rd adjacent channel selectivity		-33	-27	dB

C/I ≧3MHz	(PER < 30.8%)				
C/I Image channel	Image channel selectivity (PER < 30.8%)		-20	-9	dB
C/I Image 1MHz	1MHz adjacent to image channel selectivity		-30	-15	dB
	(PER < 30.8%)				
Out-of-band blocking	30MHz to 2,000MHz	-30			dBm
	2,001MHz to 2,339MHz	-35			dBm
	2,501MHz to 3,000MHz	-35			dBm
	3,001MHz to 12.75GHz	-30			dBm

2.6.2.6 LE Transmitter Specification

Table 2-46: Bluetooth LE transmitter specification

Parameter	Description	Min.	Typ.	Max.	Unit
Frequency range		2,402	-	2,480	MHz
Output power(*)	At max. power output level	-20	6	10	dBm
Carrier frequency offset and drift	Frequency offset	-150	±10	150	kHz
	Frequency drift	-50	±10	50	kHz
	Max. drift rate	-20	±10	20	kHz/50us
Modulation characteristic	$\Delta f1_{avg}$	225	251	275	kHz
	$\Delta f2_{max}$ (For at least 99% of all $\Delta f2_{max}$)	185	215		kHz
	$\Delta f2_{avg}/\Delta f1_{avg}$	0.8	0.88		
In-band spurious emission	±2M offset		-35	-20	dBm
	>±3MHz offset		-40	-30	dBm

*The measurement does not include exceptions in these bands. Exceptions can pass Bluetooth SIG spec.

**The measurement is at chip output.

2.7 Crystal Oscillator

2.7.1 Reference Clock

A 26MHz crystal oscillator with one external 26MHz cock buffer and one 32kHz clock output is integrated in SOC.
The mode of operation will be detected automatically, which means if an external clock is detected, it will enter external 26MHz clock mode, otherwise it will enter 32kHz clock mode.

2.7.2 Reference Output Clock Buffers Specification (for PMIC MT6392)

Table 2-47: Reference output clock buffer specification

XMODE_TP2	26M CLOCK output buffer
Max. driving capability	30pF // 3K
Swing Vpp (Max./Min.)	1.2V/0.7V
Waveform	Square
PN requirement 5Hz	-73dBc/Hz (worst)
PN requirement 10Hz	-80 dBc/Hz (worst)
PN requirement 100Hz	-105 dBc/Hz (worst)
PN requirement 1kHz	-127 dBc/Hz (worst)
PN requirement 10kHz	-140 dBc/Hz (worst)
PN requirement 100kHz	-143 dBc/Hz (worst)

2.7.3 XTAL component characteristic specification for crystal oscillation mode

Table 2-48: XTAL component spec

XTAL characteristics	Specification
Frequency Tolerance@25deg	+-10ppm
Frequency Stability over temperature	+-10ppm
ESR	<30ohm
CL	10.5pF~12.0pF
TS	10-15ppm/pF
DL	>100uW

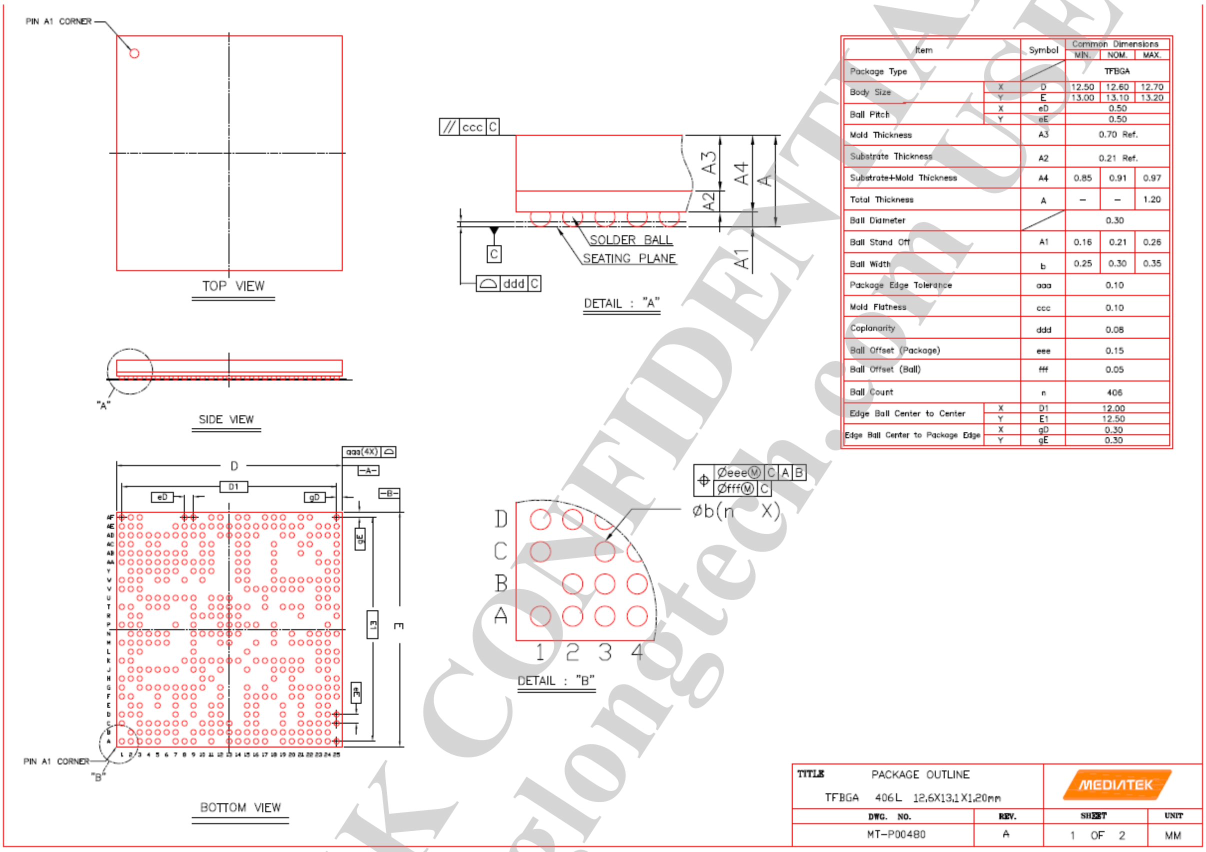
2.7.4 External reference clock oscillator specification

Table 2-49: External reference clock source specification

External reference clock(TCXO) characteristics	Specification
Max. driving capability	30pF // 3K
Swing Vpp (Max./Min.)	1.2V/0.7V
Waveform	Square
PN requirement 5Hz	<-83 dBc/Hz
PN requirement 10Hz	<-90 dBc/Hz
PN requirement 100Hz	<-115 dBc/Hz
PN requirement 1kHz	<-137 dBc/Hz
PN requirement 10kHz	<-150 dBc/Hz
PN requirement 100kHz	<-153 dBc/Hz

2.8 Package Information

2.8.1 Package Outlines

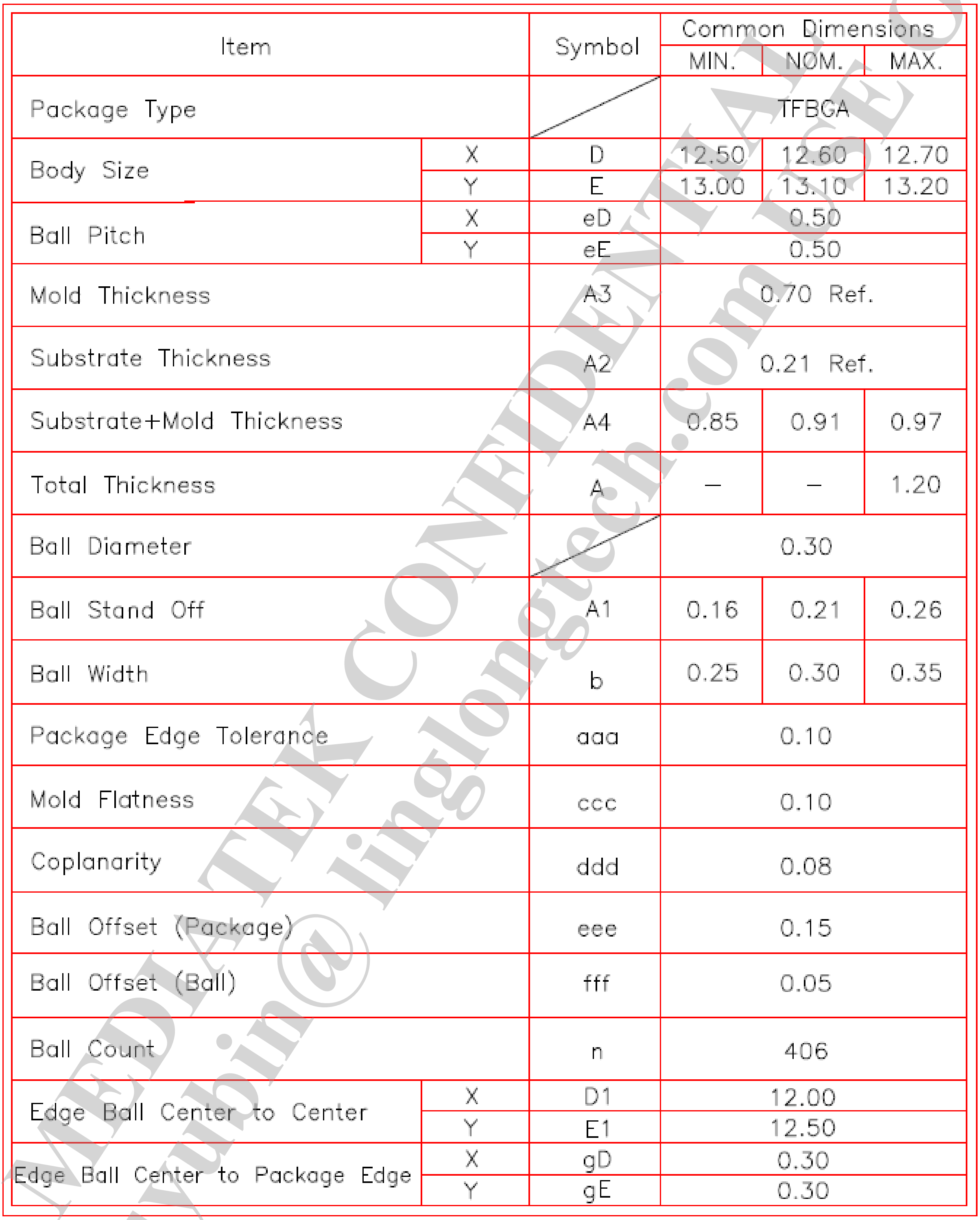


Item		Symbol	Common Dimensions		
			MIN.	NOM.	MAX.
Package Type			TFBGA		
Body Size	X	D	12.50	12.60	12.70
	Y	E	13.00	13.10	13.20
Ball Pitch	X	eD	0.50		
	Y	eE	0.50		
Mold Thickness		A3	0.70 Ref.		
Substrate Thickness		A2	0.21 Ref.		
Substrate+Mold Thickness		A4	0.85	0.91	0.97
Total Thickness		A	–	–	1.20
Ball Diameter			0.30		
Ball Stand Off		A1	0.16	0.21	0.26
Ball Width		b	0.25	0.30	0.35
Package Edge Tolerance		aaa	0.10		
Mold Flatness		ccc	0.10		
Coplanarity		ddd	0.08		
Ball Offset (Package)		eee	0.15		
Ball Offset (Ball)		fff	0.05		
Ball Count		n	406		
Edge Ball Center to Center	X	D1	12.00		
	Y	E1	12.50		
Edge Ball Center to Package Edge	X	gD	0.30		
	Y	gE	0.30		

TITLE	PACKAGE OUTLINE TFBGA 406L 12.6X13.1X1.20mm		MEDIATEK	
DWG. NO.		REV.	SHEET	UNIT
MT-P00480		A	1 OF 2	MM

*Figure 2-22: Outlines and dimensions of TFBGA 12.6mm*13.1mm, 406-ball, 0.5mm pitch package*

Table 2-50: Package Details



Item		Symbol	Common Dimensions		
			MIN.	NOM.	MAX.
Package Type				TFBGA	
Body Size	X	D	12.50	12.60	12.70
	Y	E	13.00	13.10	13.20
Ball Pitch	X	eD		0.50	
	Y	eE		0.50	
Mold Thickness		A3		0.70 Ref.	
Substrate Thickness		A2		0.21 Ref.	
Substrate+Mold Thickness		A4	0.85	0.91	0.97
Total Thickness		A	—	—	1.20
Ball Diameter				0.30	
Ball Stand Off		A1	0.16	0.21	0.26
Ball Width		b	0.25	0.30	0.35
Package Edge Tolerance		aaa		0.10	
Mold Flatness		ccc		0.10	
Coplanarity		ddd		0.08	
Ball Offset (Package)		eee		0.15	
Ball Offset (Ball)		fff		0.05	
Ball Count		n		406	
Edge Ball Center to Center	X	D1		12.00	
	Y	E1		12.50	
Edge Ball Center to Package Edge	X	gD		0.30	
	Y	gE		0.30	

2.8.2 Thermal Operating Specifications

Table 2-51: Thermal operating specifications

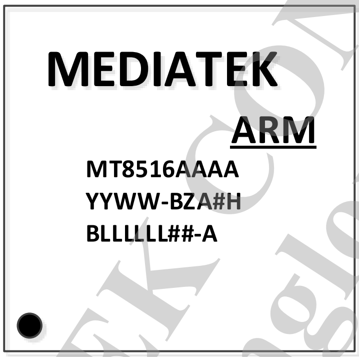
Symbol	Description	Value	Unit	Notes
	Maximum operating junction temperature	105	°C	
	Package thermal resistances in nature convection	31.03	°C/Watt	

2.8.3 Lead-free Packaging

MT8516A is provided in a lead-free package and meets RoHS requirements.

2.9 Ordering Information

2.9.1 Top Marking Definition



MT8516AAAA : Part No
YYWW : Data Code
#: Subcontractor Code
LLLLLL: Lot ID

Figure 2-23: Top mark of MT8516A

2.9.2 Ordering Part Number

MT8516AAAA/B

3 Clock and Power Control

3.1 Chrystal Oscillator (XO)

3.1.1 Introduction

A 26MHz crystal oscillator with one external 26MHz cock buffer and one 32kHz clock output is integrated in SOC. The mode of operation (internal crystal oscillator clock or external CLK) is selected via the input XMODE_TP2 with the following function.

Table 3-1. Reference Clock Operation Mode

XMODE_TP2	Mode
Low	External CLK
High	Crystal oscillator CLK

3.1.2 XO Block Diagram

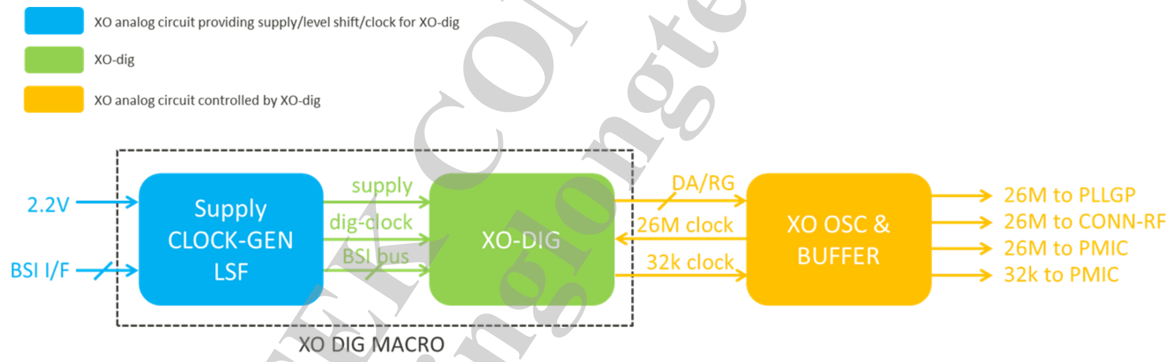


Figure 3-1. Chrystal Oscillator Block Diagram

3.1.3 Features

Reference Output Clock Buffers Specification (for PMIC MT6392)

Table 3-2. Reference Output Clock Buffer Specification

XMODE_TP2	26M CLOCK output buffer
Max. driving capability	30pF // 3K
Swing Vpp (Max./Min.)	1.2V/0.7V
Waveform	Square
PN requirement 5Hz	-73 (worst)
PN requirement 10Hz	-80 (worst)

XMODE_TP2	26M CLOCK output buffer
PN requirement 100Hz	-105 (worst)
PN requirement 1kHz	-127 (worst)
PN requirement 10kHz	-140 (worst)
PN requirement 100kHz	-143 (worst)

XTAL component characteristic specification for crystal oscillation mode

Table 3-3. XTAL Component Specification

XTAL characteristics	Specification
Frequency Tolerance@25deg	+-10ppm
Frequency Stability over temperature	+-10ppm
ESR	<30ohm
CL	10.5p~12.0p
TS	>10ppm/pF
DL	>100uW

External reference clock oscillator specification

Table 3-4. External Reference Clock Source Specification

External reference clock(TCXO) characteristics	Specification
Max. driving capability	30pF // 3K
Swing Vpp (Max./Min.)	1.2V/0.7V
Waveform	Square
PN requirement 5Hz	<-83
PN requirement 10Hz	<-90
PN requirement 100Hz	<-115
PN requirement 1kHz	<-137
PN requirement 10kHz	<-150
PN requirement 100kHz	-153

3.1.4 Register Definition

Refer to chapter 1.1 in “MT8516A Application Processor Registers.”

3.2 AP Mixed Mode Control System

3.2.1 Phase Locked Loop

3.2.1.1 Clock Introduction

The 26 MHz from XO or external OT 26 MHz clock. This signal is then converted to the square-wave signal through CLKSQ and used as the reference clock of Phase-Locked Loops (PLL). There are total 8 PLLs in the analog PLLGP macro: ARMPLL, MAINPLL, UNIVPLL, AUD1PLL and AUD2PLL. These PLLs provide clock sources for CPU, BUS, MSDC.

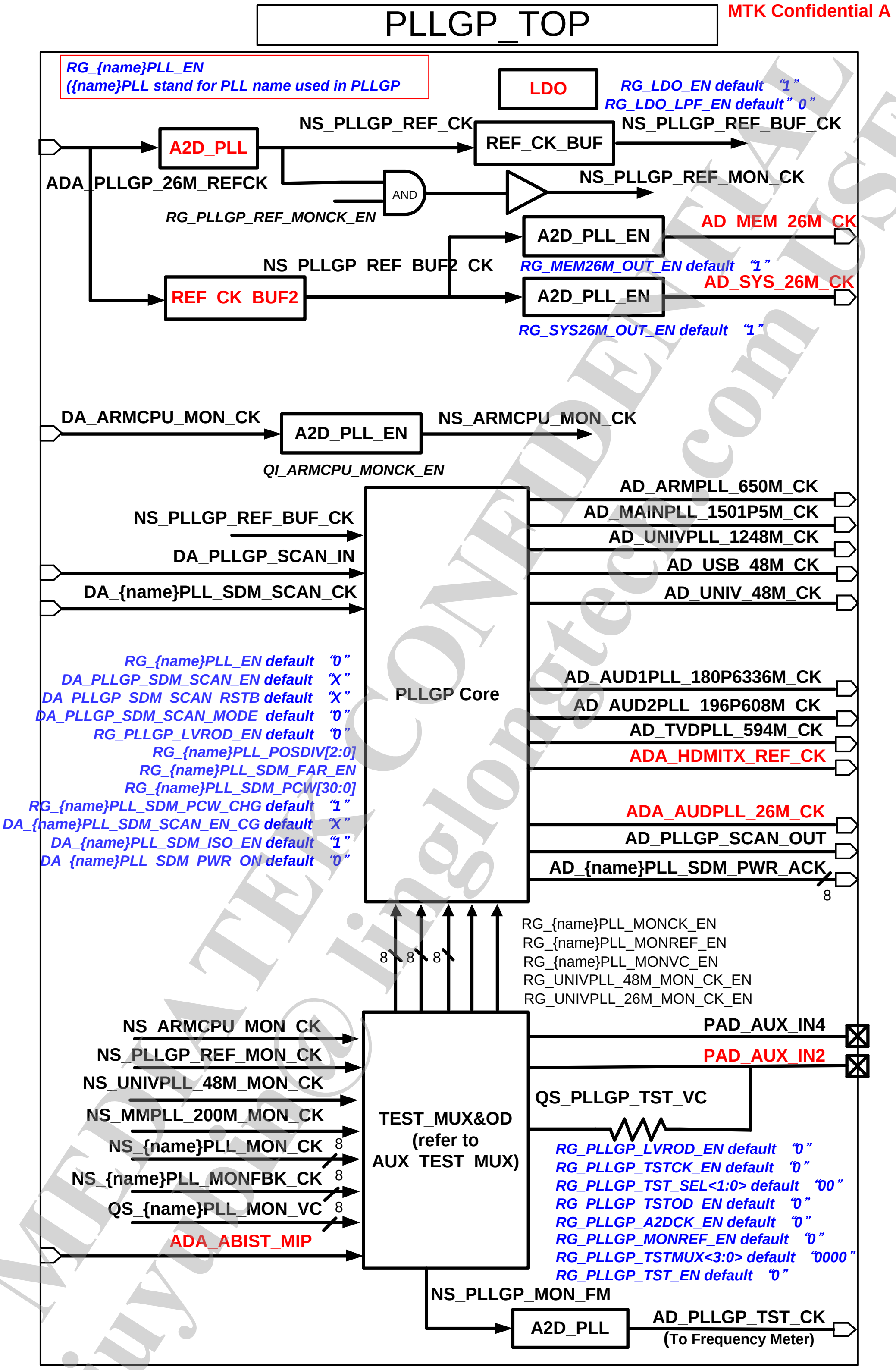


Figure 3-2.Clock Sources Block Diagram

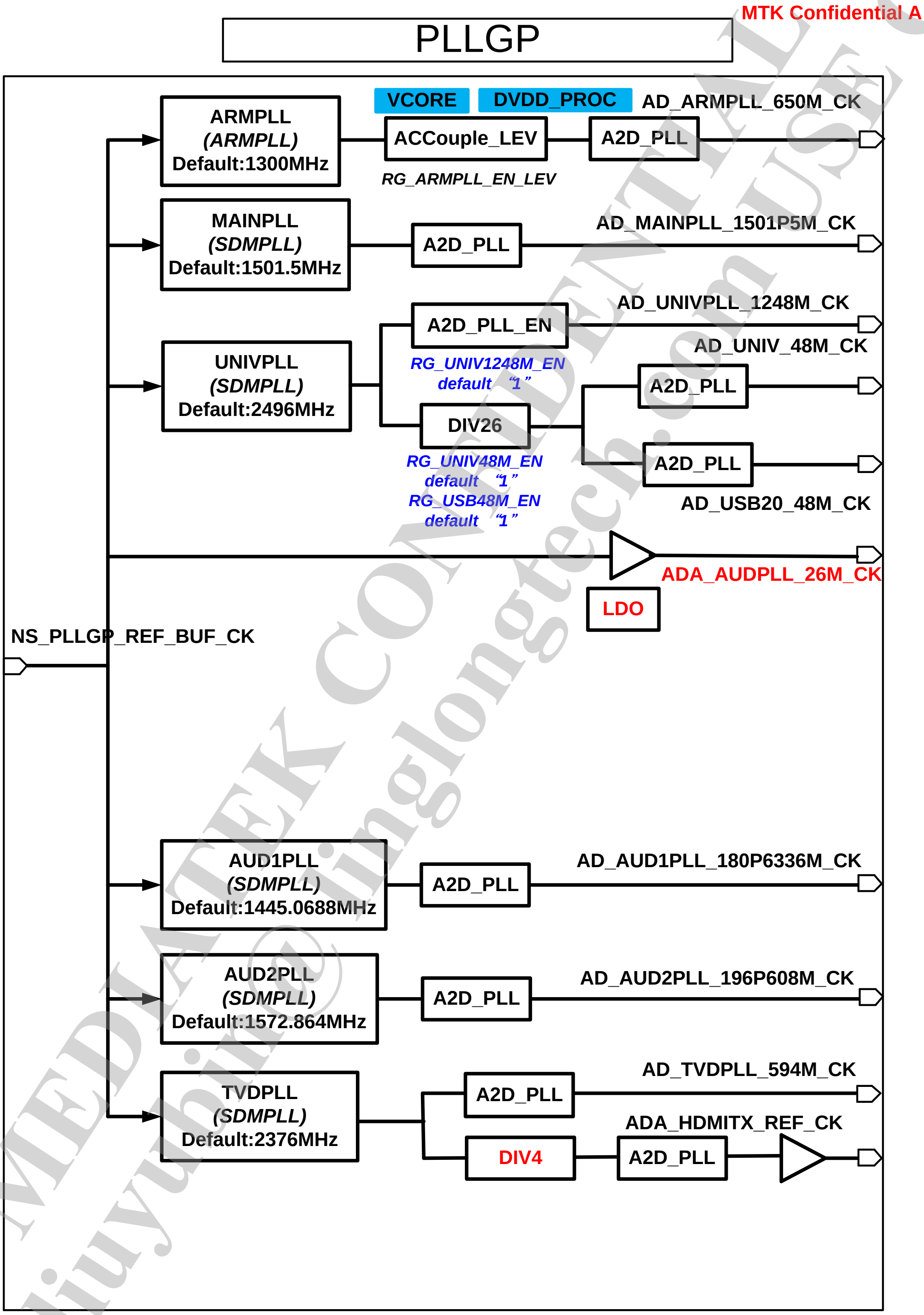


Figure 3-3. PLL Core Block Diagram

3.2.2 PLL Functional Specifications

Please check Charpter2.5 Analog Baseband.

3.2.2.1 PLL Power-on Sequence

The default setting to turn on PLLs is SW mode. SW has to turn on each PLL signal step-by- step. An automatic PLL power on sequence control is implemented to automatically turn on PLL from sleep mode to normal mode. SW has to set to HW mode to use the automatic PLL power-on sequence control.

PLL_EN is used to turn on each PLL, and PLLs only needs this step to be turned on.

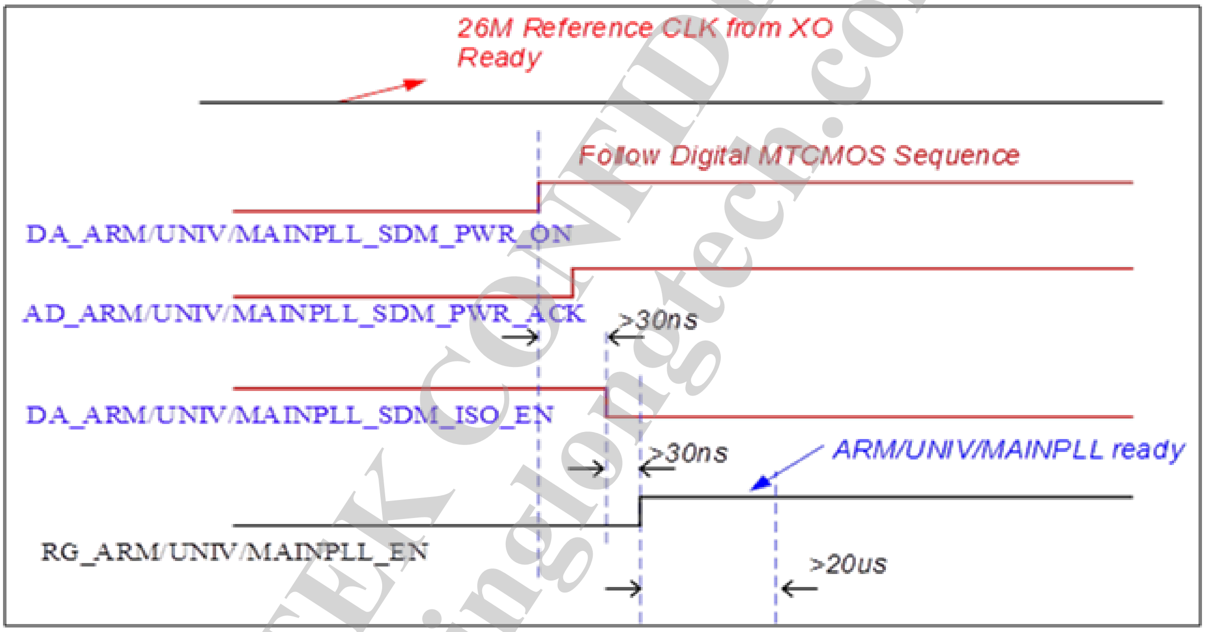

26M Reference CLK from XO Ready
Follow Digital MTCMOS Sequence
DA_ARM/UNIV/MAINPLL_SDM_PWR_ON
AD_ARM/UNIV/MAINPLL_SDM_PWR_ACK
>30ns
DA_ARM/UNIV/MAINPLL_SDM_ISO_EN
>30ns
ARM/UNIV/MAINPLL ready
RG_ARM/UNIV/MAINPLL_EN
>20us

Figure 3-4.PLL Power-on Sequence

3.2.3 Register Definitions

For register details refer to chapter 1.2 in “MT8516A Application Processor Registers.”

3.3 Top Clock Generator (TOPCKGEN)

3.3.1 Introduction

This chapter describes the entire clock topology of MT8516A and the digital clock management architecture. In MT8516A, a three-level clock structure is adopted to optimize the clock tree power. PLL and top_clock_ctrl (CKSYS) represent top-level clock generation for each subsystem domain. In each subsystem domain, there is a global clock control (global_con) module to generate the second level clock for sub module level. In the module level, there is a power compiler generated clock for the leave side F/F. Figure 3-5 shows the architecture and hierarchy.

3.3.2 Features

TOPCKGEN is responsible for generating the following clock signals:

- Free clock generation for 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 system

3.3.3 Block Diagram

In addition, as shown in Figure 3-5, top_clock_ctrl can be separated into three parts, PLL_DIV, CLK_SW, and Top Global Con. PLL_DIV generates clock sources for the AP platform. CLK_SW selects suitable clock frequency for all operational clocks, and Top Global Con controls top-level clock gating/DCM control. DCM is the Digital Clock Manager used for saving the power of clock tree. Top Global Con has HW EMI DCM and HW Bus DCM functions.

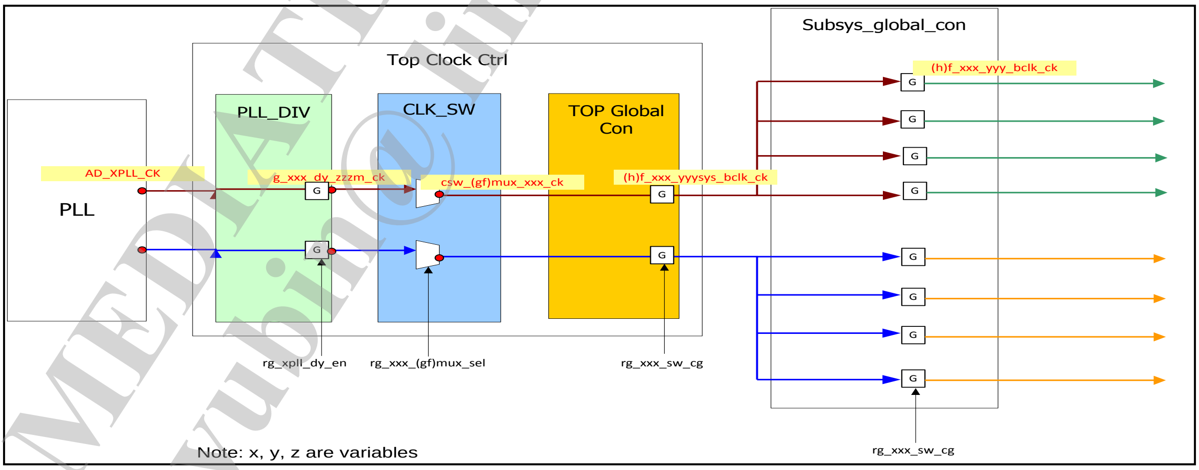


Figure 3-5.Clock Architecture and Hierarchy

3.3.4 Theory of Operations

PLL_DIV is for dividing the analog PLL output high speed clock into all possible clock speed required by each module or sub-system. After that, CLK_SW selects a speed from all the possible clock speed for each module or sub-system depending on the application scenario or limitation on hardware design.

Subsystem Global Con is responsible for generating module clocks within the sub-system. In addition, it is capable of controlling the module clock on/off or DCM. Most of the module clocks in Subsys Global Con use SW register on/off control. HW DCM is majorly used for bus fabric component clock management.

In MT8516A, hardware automatic clock gating or slow-down management scheme is implemented to save the power of clock tree when the targeted HW circuits are in the IDLE state. In other words, the detection of IDLE or Busy is performed by clock DCM HW circuits, which only requires a few interactions with SW (to enable the feature and set up some function parameters). There are many hardware DCM features in MT8516A.

3.3.5 Programming Guide

3.3.5.1 Frequency Meter

There are two frequency meters embedded inside TOPCKGEN.
1. Set fmeter_en to 1'b1.
2. Choose frequency meter source by clk_dbg_cfg[0]
3. Choose target clock by changing abist_clk_sel / ckgen_clk_sel.
4. Change ckgen_k1 for dividing target clock (optional).
5. Change reference clock by changing clk_exc (optional).
6. Change ckgen_load_cnt (optional).
7. Trigger frequency meter by set ckgen_tri_cal = 1b'1.
8. Wait until ckgen_tri_cal = 1'b0.
9. Read frequency meter result from ckgen_cal_cnt.
   freq(target) = (ckgen_k1 + 1)*[freq(reference clock)*ckgen_cal_cnt]/(ckgen_load_cnt + 1)

3.3.6 Register Definitions

For register details refer to chapter 1.3 in "MT8516A Application Processor Registers."

3.4 Frequency Hopping Control (FHCTL)

3.4.1 Introduction

The frequency hopping controller helps AP resolve 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 in ABB. The harmonic of all clock frequency may de-sense the band of RF system.

3.4.2 Features

The frequency hopping controller receives the command from the CPU to trigger two mechanisms:

- Spread spectrum clocking
- Frequency hopping

In MT8516A, there are three hopping PLLs:

Name	Capability	Range
ARMPLL	Hopping, SSC	{-8%,0}
MAINPLL	Hopping, SSC	{-8%,0}
MEMPLL	Hopping, SSC	{-8%,0}

3.4.3 Block Diagram

Whenever MCUSYS enters sleep mode, SRCLKENA from the sleep controller is de-asserted. The SRCLKENA from MCUSYS controls the power supply for the 13MHz/26MHz TCVCXO via the on-chip PMU. When the signal is de-asserted, LDO for TCVCXO in PMU will be turned off, and 13MHz/26MHz clock will stop.

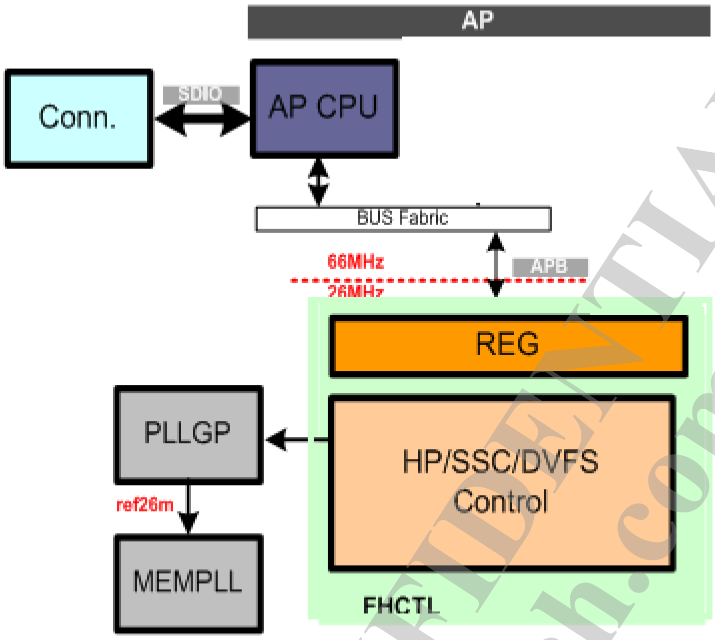


Figure 3-6. Frequency Hopping Controller Block Diagram

3.4.4 Register Definitions

For register details refer to chapter 1.4 on “MT8516A Application Processor Registers.”

3.5 Top Reset Generate Unit (TOPRGU)

3.5.1 Introduction

The top reset generator unit (TOPRGU) generates reset signals and distributes them to each system. A watchdog timer is also included in this module.

3.5.2 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

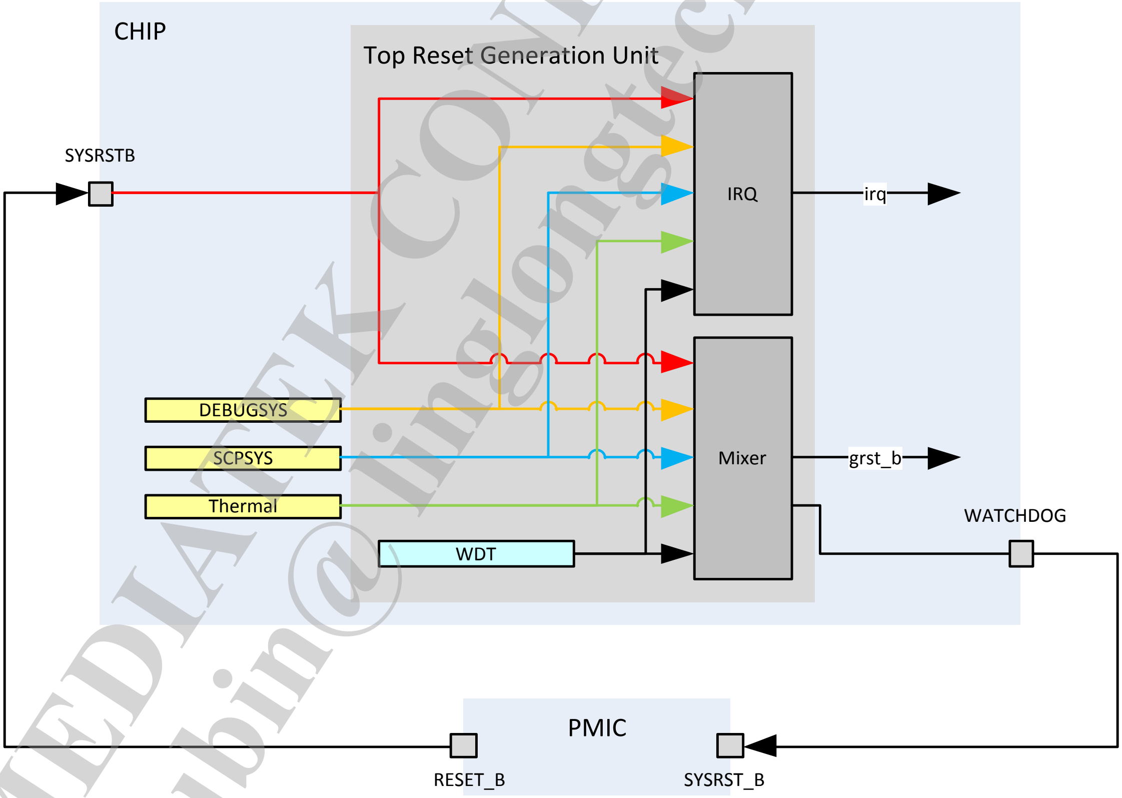


Figure 3-7. Top Reset Generation Unit Block Diagram

3.5.4 Register Definitions

For register details refer to chapter 1.5 in “MT8516A Application Processor Registers.”

3.6 MTCMOS Domains

3.6.1 Power Domain Introduction

The figure below shows all power blocks in MT8516A, including CPU/DDR/CONN.

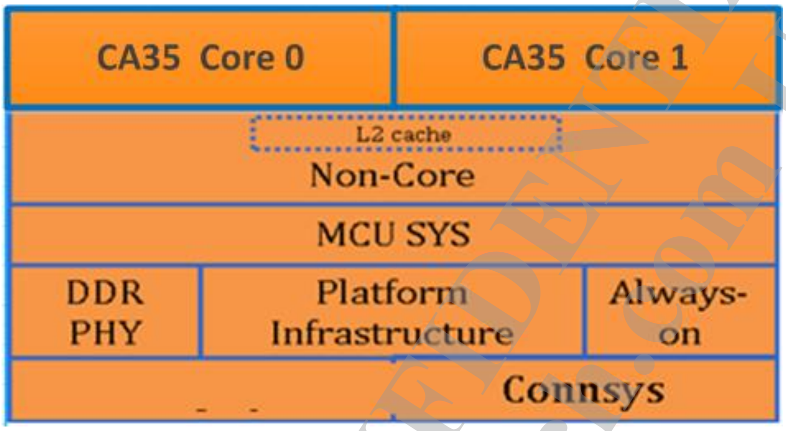


Figure 3-8. MT8516A Power Domain Block Diagram

3.6.2 MCUSYS MTCMOS

In MT8516A, fine grain MTCMOS control and DVFS control are provided to achieve high performance with low power in MCUSYS. There is one cluster in MCUSYS with Quad-core CA35. Total 4 MTCMOS domains are listed in the table below

Table 3-5. MTCMOS domain in MCUSYS

Domain	Description	Boot-up Power State
CA35-CPU0	CA35 CPU0 power domain	ON
CA35-CPU1	CA35 CPU1 power domain	OFF
CA35-CPUTOP	CA35 CPUTOP power domain	ON
MCUSYS	MCUSYS power domain	ON
(*) clock of this power domain is gated by default with reset asserted.		

3.6.3 Other MTCMOS Subsystems

All the other MTCMOS domains reside in VCORE power domain.

Table 3-6. VCORE MTCMOS Domains

Domain	Description	Boot-up Power State
INFRA	Infra domain	ON
CONN	Connsys domain (2 domains)	OFF

Domain	Description	Boot-up Power State
DDRPHY	DDR Phy domain	ON

3.6.4 DVFS

CPU DVFS are in MT8516A, but no DVFS in SOC.

3.6.5 CPU DVFS

There are three different power sources in CPU: VPROC, VSRAM_PROC, and VCORE. These power sources are driven from in-house PMIC (MT6392). DVFS table is shown below. For CA35, the minimum voltage is 1.15V. CA35_OPP1 provides the maximum performance of CA35 running at 1.3GHz and CA35_OPP5 aims at light-loading scenario which CA35 is running at 598MHz. VSRAM_PROC is slightly higher than VPROC for the same OPPs. PTPOD will be applied in these power domains, and thus the real voltage value, derived from PTPOD controller, depends on each chip with different corners.

Table 3-7. CPU DVFS

Power Domain	OPPs	Voltage (V)	Frequency (Mhz)	VSRAM (V)	STA Record
	CA35_OPP1	1.25-delta	1300	1.25-delta	FY = 1300Mhz SB = 1500Mhz
	CA35_OPP2	1.25-delta	1196	1.25-delta	
	CA35_OPP3	1.20-delta	1040	1.20-delta	
	CA35_OPP4	1.15-delta	747.5	1.15-delta	
	CA35_OPP5	1.15-delta	598	1.15-delta	

3.6.6 Power Mode Scenarios

Although there is DCM in MT8516A to reduce the power in the subsystem, to further improve the system level low power, we implement a system power manager (SPM) to control the top level power by different power scenarios. Table 3-8 shows the Android power state. All the power management scenarios are based on the Android power state. For more detail about CPU state, infra state and MM state, it is depicted in Table 3-9.

Table 3-8. Android Power State

Android Power State	Description	Backlight	CPU State	Infra State	MM State
Running	Normal operation	On	PTPOD/Turbo/Norm/Light	Normal	Normal

Android Power State	Description	Backlight	CPU State	Infra State	MM State
User idle	State after a longer period of user inactivity	On	Idle	Normal	Normal
Early suspend	CPU running. No user visible feature	Off	Turbo	Normal/Deep	Pwrdown
Early suspend idle	State after a longer period of CPU inactivity. No user visible feature	Off	Idle/Dormat	Normal/Deep idle	Pwrdown
Suspend	Sleep State, no thread running	Off	Pwrdown	Normal/Deep idle	Pwrdown
Power off	Power down	OFF	Pwrdown	Pwrdown	Pwrdown

Table 3-9. Power Mode Scenarios

	MT8516A	MT8516A	MT8516A	MT8516A	MT8516A
AP state	normal	slow idle	deep idle	suspend	power off
PMIC state	normal	normal	normal	sleep	power off
ARM CPU state	running, WFI	WFI	dormant	shutdown	shutdown
Core num.	1~4	1	0	0	0
Vproc	1.15~ 1.25V	1.15~ 1.25V	0.85 V	0	0
Vcore	1.15 V[2]	1.15V	1.15V	0.85V	0
DDRPHY	on	on	off	off	off
dram operation clk	1600 Mbps	1600 Mbps	off	off	off
bus operation clk	133MHz	133MHz	off	off	off
OSC (26M/32K)	on/on	on/on	on/on	off/on	off/on
Infra mtcmos	on	on	on	off	off
HW latency[3]	0	0	~1ms	~1ms (suspend) / ~5ms (resume)	~20ms (off)/ ~100ms (on)
Shared buck It is estimated number and CPU operating time is not included.					

3.7 PMIC_WRAP

3.7.1 Introduction

The PMIC wrapper serves as the bridge for the communication of AP and PMIC.

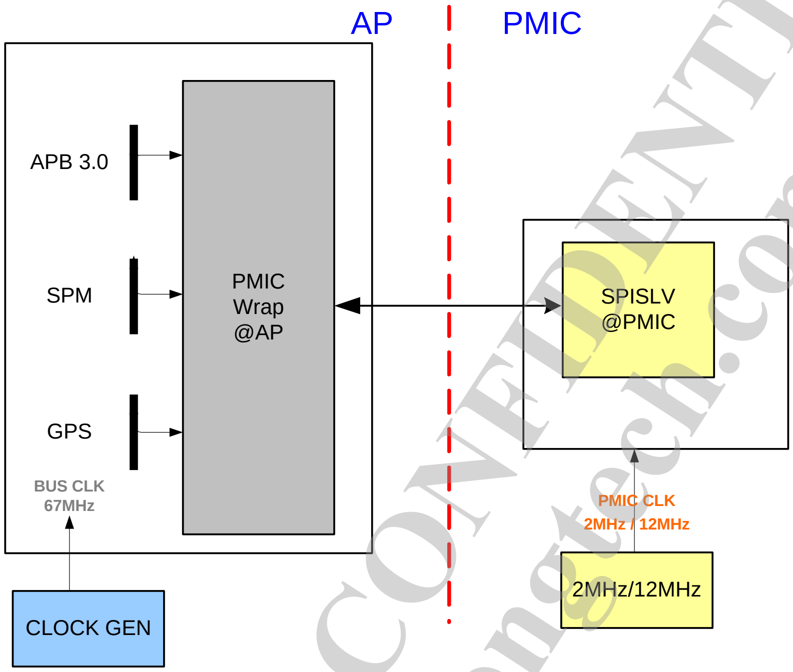


Figure 3-9. PMIC_WRAP Overview

3.7.2 Features

- Fast auto SPI format generator for PMIC register read/write
- APB3.0 bus lock scheme when SPI is busy
- Manually SPI format generator
- Dual I/O SPI mode
- Separated frequency between controller and SPI

3.7.3 PMIC_WRAP Block Diagram

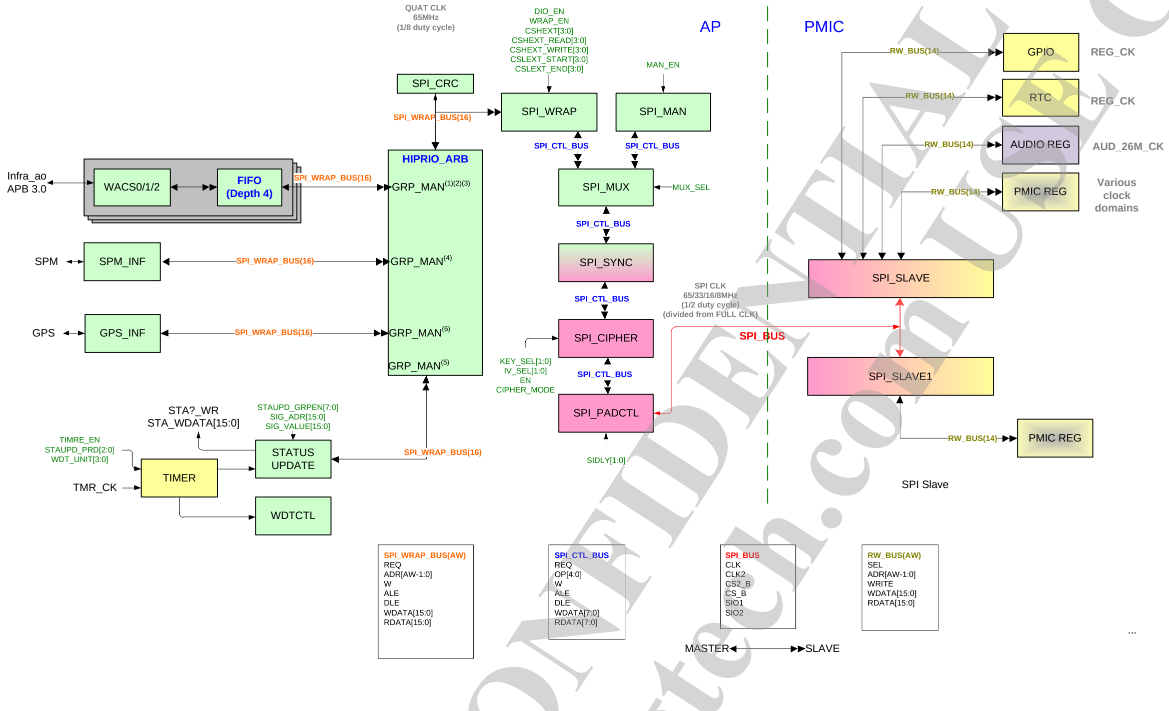


Figure 3-10. PMIC_WRAP Architecture

- 1-level arbitration is used to handle all PMIC read/write request.
- High-priority arbiter without FIFO is used to keep the latency of high priority request low.
- “STATUS UPDATE” is used to automatically read 4 registers @PMIC including CRC value and GPS data.
- SPI_WRAP transforms the read/write request into SPI formats.
- SPI_MAN can generate the SPI formats manually by control register.
- SPI_MUX selects the controller to be SPI_WRAP or SPI_MAN.
- SPI_SYNC is the asynchronous interface between controller and SPI_PADCTL.
- SPI_PADCTL controls the SPI pads to meet the SPI formats.
- Internal DCM saves power.

3.7.3.1 SPI Format

The pre-defined SPI format consists of two modes: single I/O mode and dual I/O mode. The single I/O mode always uses MOSI and MISO as output and input respectively. The dual I/O mode uses both MOSI and MISO to be output and input to achieve better channel use. The format conveys information of R/W direction, 15-bit address (bit 15 to bit 1) and 16-bit data. The operation waveform of SPI is illustrated as below. Parameter t_{CSLEXT_START}, t_{RDDMY}, t_{CSLEXT_END} and t_{CSHEXT} are fully configurable through command registers as illustrated below..

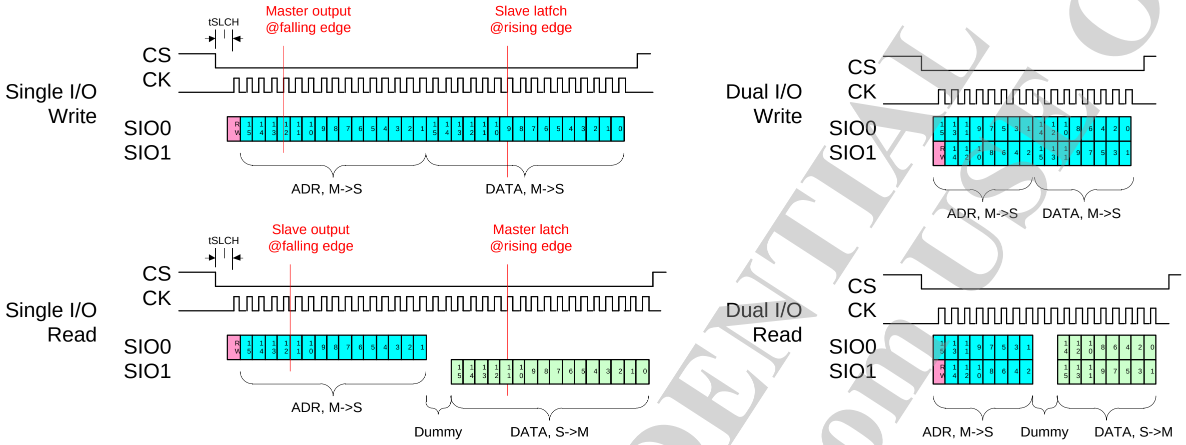


Figure 3-11. SPI Format

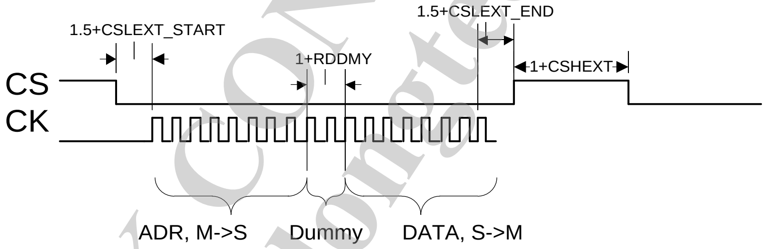


Figure 3-12. SPI Parameter Configuration

3.7.3.2 SPI Soft Reset Pattern

If the SPI slave or dewrapper is out of control, all register accesses will fail due to SPI transaction fails. In order to recover from this situation, a reset format is defined in Figure 3-13. It should be noted that in normal SPI transaction, there should not be any CK when CS=high. The SPI soft reset will apply to SPI slave (and DEWRAP) in PMIC.

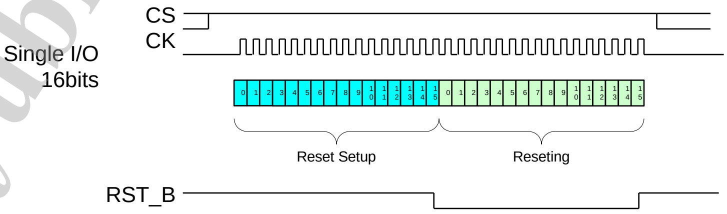


Figure 3-13. SPI Reset Pattern

3.7.4 Register Definitions

For register details refer to chapter 1.7 in “MT8516A Application Processor Registers.”

3.7.5 Programming Guide (Programming Model)

3.7.5.1 Scenario

3.7.5.1.1 SPI & WRAPPER Reset Flow

To reset wrapper @AP, SPI slave (and dewrapper) @PMIC, the recommended flow illustrated in Figure 3-14 below is used to guarantee the correctness of re-initialization. To reset SPI slave (and dewrapper) @PMIC, the reset pattern specified in Figure 3-13 is used. Refer to the example code below to generate SPI reset pattern.

Example code (SPI Reset Pattern)

```
`define MAN_DLE_RESTCNT 0x00030000;
WriteReg32( MAN_CMD, OP_WR | OP_CSL );
WriteReg32( MAN_CMD, OP_WR | OP_OUTS ); //to reset counter
WriteReg32( MAN_CMD, OP_WR | OP_CSH );
WriteReg32( MAN_CMD, OP_WR | OP_OUTS);
WriteReg32( MAN_CMD, OP_WR | OP_OUTS);
WriteReg32( MAN_CMD, OP_WR | OP_OUTS );
WriteReg32( MAN_CMD, OP_WR | OP_OUTS );
```

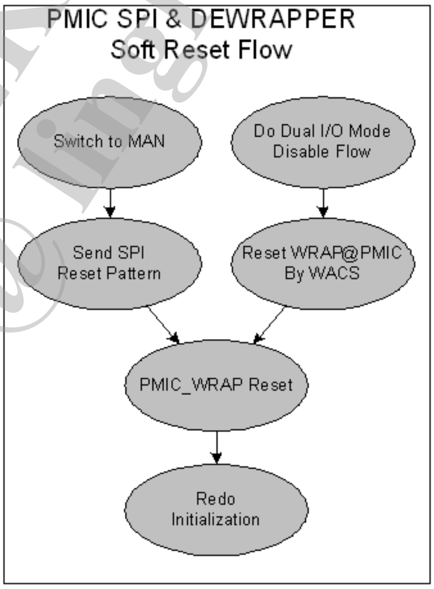


Figure 3-14.SPI and WRAPPER reset flow

3.7.5.1.2 Initialization

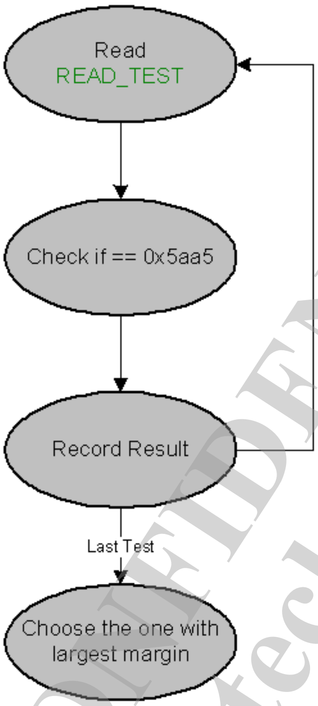


Figure 3-15.Initialization flow

4 MCU Bus and Fabric

4.1 MCU System

4.1.1 Introduction

The MCU system is a responsible for running the operating system and application programs in MT8516A. It consists of four Cortex-A35 cores into one cluster, and Generic Interrupt Controller (GIC). A 128-bit AXI bus is directly connected to External Memory Interface (EMI) to minimize the access latency to DRAM and provide sufficient memory bandwidth. The peripheral system and on-chip storage are bridged through AXI bus, and the outstanding capability of AXI protocol allows the system to exploit its maximum throughput from four CPU cores.

The MCU system supports DVFS technology which allows CPU to run at different frequency/voltage configurations for different application requirements. When in standby mode, MCUSYS can be completely shut down to further save power consumption and optimize the battery usage on mobile devices.

4.1.2 Cluster 0, Cortex-A53 Specifications

- Quad-core ARM® Cortex-A35 MPCore™ operating at 1.3 GHz
- Supports ARMv8-A architecture for both 32 and 64-bit execution state
- Supports NEON multimedia processing engine with SIMDv2/VFPv4 ISA
- Optional support ARMv8 Cryptographic extension
- 32KB L1 I-cache and 32KB L1 D-cache
- 512KB unified L2 cache for CPU cluster
- DVFS technology with adaptive operating voltage from 0.9V to 1.31V

4.1.3 Clock Modes between Clusters and AXI bus Fabric

The CPU and AXI bus fabric is synchronous and with integer clock ratio for example 1:1/1:2/1:4 for best system performance. CPU and AXI bus fabric also support Dynamic Clock Management (DCM) mechanism to dynamically turn off the clock when no transactions are on the bus interface. CPU, cluster and AXI bus fabric can also support DVFS technology to lower power consumption.

4.1.4 Interrupt Controller

MT8516A uses the ARM GIC400 interrupt controller for interrupt management. GIC400 is embedded inside MCU system to minimize the interrupt handling latency. For the interrupt connected to GIC400, see the table below for details. The GIC interrupts are separated into 2 categories, the Private Peripheral Interrupts (PPI) and Shared Peripheral Interrupts (SPI). PPI occupies the first 32 interrupt slots in GIC and are banked for each CPU core. SPI begins from the 33rd interrupt and is shared by all CPU cores.

Table 4-1. Interrupt Request List for Cortex-A35

GIC ID	*Interrupt source/name*	*Polarity*	*Trigger type*
0	Software generated interrupt 0	H	Edge
1	Software generated interrupt 1	H	Edge
2	Software generated interrupt 2	H	Edge
3	Software generated interrupt 3	H	Edge
4	Software generated interrupt 4	H	Edge
5	Software generated interrupt 5	H	Edge
6	Software generated interrupt 6	H	Edge
7	Software generated interrupt 7	H	Edge
8	Software generated interrupt 8	H	Edge
9	Software generated interrupt 9	H	Edge
10	Software generated interrupt 10	H	Edge
11	Software generated interrupt 11	H	Edge
12	Software generated interrupt 12	H	Edge
13	Software generated interrupt 13	H	Edge
14	Software generated interrupt 14	H	Edge
15	Software generated interrupt 15	H	Edge
16	Software generated interrupt 16	H	Edge
17	Software generated interrupt 17	H	Edge
18	Software generated interrupt 18	H	Edge
19	Software generated interrupt 19	H	Edge
20	Software generated interrupt 20	H	Edge
21	Software generated interrupt 21	H	Edge
22	Software generated interrupt 22	H	Edge
23	Software generated interrupt 23	H	Edge
24	Software generated interrupt 24	H	Edge
25	Software generated interrupt 25	H	Edge
26	Software generated interrupt 26	H	Edge
27	Software generated interrupt 27	H	Edge
28	Software generated interrupt 28	H	Edge
29	Software generated interrupt 29	H	Edge
30	Software generated interrupt 30	H	Edge
31	Software generated interrupt 31	H	Edge
32	mp0_NIRQOUT[0]	L	level
33	mp0_NIRQOUT[1]	L	level

GIC ID	Interrupt source/name	Polarity	Trigger type
34	mp0_NIRQOUT[2]	L	level
35	mp0_NIRQOUT[3]	L	level
36	(Reserved)	-	-
37	(Reserved)	-	-
38	(Reserved)	-	-
39	(Reserved)	-	-
40	mp0_NPMUIRQ[0]	L	level
41	mp0_NPMUIRQ[1]	L	level
42	mp0_NPMUIRQ[2]	L	level
43	mp0_NPMUIRQ[3]	L	level
44	(Reserved)	-	-
45	(Reserved)	-	-
46	(Reserved)	-	-
47	(Reserved)	-	-
48	mp0_NCNTHPIRQ[0]	L	level
49	mp0_NCNTHPIRQ[1]	L	level
50	mp0_NCNTHPIRQ[2]	L	level
51	mp0_NCNTHPIRQ[3]	L	level
52	(Reserved)	-	-
53	(Reserved)	-	-
54	(Reserved)	-	-
55	(Reserved)	-	-
56	mp0_NCNTVIRQ[0]	L	level
57	mp0_NCNTVIRQ[1]	L	level
58	mp0_NCNTVIRQ[2]	L	level
59	mp0_NCNTVIRQ[3]	L	level
60	(Reserved)	-	-
61	(Reserved)	-	-
62	(Reserved)	-	-
63	(Reserved)	-	-
64	mp0_NCNTPSIRQ[0]	L	level
65	mp0_NCNTPSIRQ[1]	L	level
66	mp0_NCNTPSIRQ[2]	L	level
67	mp0_NCNTPSIRQ[3]	L	level
68	(Reserved)	-	-
69	(Reserved)	-	-
70	(Reserved)	-	-
71	(Reserved)	-	-

GIC ID	Interrupt source/name	Polarity	Trigger type
72	mp0_NCNTPNSIRQ[0]	L	level
73	mp0_NCNTPNSIRQ[1]	L	level
74	mp0_NCNTPNSIRQ[2]	L	level
75	mp0_NCNTPNSIRQ[3]	L	level
76	(Reserved)	-	-
77	(Reserved)	-	-
78	(Reserved)	-	-
79	(Reserved)	-	-
80	mp0_NEXTERRIRQ	L	level
81	(Reserved)	-	-
82	mp0_CTIIRQ_SYNC[0]	L	level
83	mp0_CTIIRQ_SYNC[1]	L	level
84	mp0_CTIIRQ_SYNC[2]	L	level
85	mp0_CTIIRQ_SYNC[3]	L	level
86	(Reserved)	-	-
87	(Reserved)	-	-
88	(Reserved)	-	-
89	(Reserved)	-	-
90	(Reserved)	-	-
91	(Reserved)	-	-
92	(Reserved)	-	-
93	(Reserved)	-	-
94	(Reserved)	-	-
95	CCI_NERRORIRQ	L	level
96	xgpt_irq[0]	H	level
97	xgpt_irq[1]	H	level
98	xgpt_irq[2]	H	level
99	xgpt_irq[3]	H	level
100	xgpt_irq[4]	H	level
101	xgpt_irq[5]	H	level
102	xgpt_irq[6]	H	level
103	xgpt_irq[7]	H	level
104	usb_mcu_int_b	L	level
105	(Reserved)	L	level
106	(Reserved)	-	-
107	lowbattery_irq_b	L	edge
108	pwm_irq_b	L	level

GIC ID	Interrupt source/name	Polarity	Trigger type
109	ptp_therm_irq_b	L	level
110	msdc0_irq_b	L	level
111	msdc1_irq_b	L	level
112	i2c0_irqb	L	level
113	i2c1_irqb	L	level
114	i2c2_irqb	L	level
115	ptp_fsm_int_b	L	level
116	uart0_irq_b	L	level
117	uart1_irq_b	L	level
118	nfiecc_irq_b	L	level
119	nfi_irq_b	L	level
120	dma_irq_b[0]	L	level
121	dma_irq_b[1]	L	level
122	dma_irq_b[2]	L	level
123	dma_irq_b[3]	L	level
124	dma_irq_b[4]	L	level
125	dma_irq_b[5]	L	level
126	dma_irq_b[6]	L	level
127	dma_irq_b[7]	L	level
128	dma_irq_b[8]	L	level
129	dma_irq_b[9]	L	level
130	dma_irq_b[10]	L	level
131	dma_irq_b[11]	L	level
132	dma_irq_b[12]	L	level
133	dma_irq_b[13]	L	level
134	dma_irq_all_b	L	level
135	dma_fiq_all_b	L	level
136	spi1_irq_b	L	level
137	msdc0_wakeup_ps	H	edge
138	msdc1_wakeup_ps	H	edge
139	btif_irq_b	L	level
140	msdc2_wakeup_ps	H	edge
141	msdc2_irq_b	L	level
142	hdmi_sifm_irqb	L	level
143	ether_nic_wrap_irq_b	L	level
144	dcc_aparm_irq (debugtop_irq_b)	L	level

GIC ID	Interrupt source/name	Polarity	Trigger type
145	refresh_rate_int_pulse	L	level
146	apmcu_domain_irq_b	L	level
147	apmcu_decerr_irq_b	L	level
148	domain_abort	H	level
149	cq_dma_irq_b[0]	L	level
150	cq_dma_irq_b[1]	L	level
151	cq_dma_irq_b[2]	L	level
152	afe_irq_mcu_b	L	level
153	mmu_irq_b	L	level
154	mmu_sec_irq_b	L	level
155	bus_dbg_tracker_irq_b	L	level
156	ahbmon_irq_b	L	level
157	gce_irq_b	L	level
158	gce_secure_irq_b	L	level
159	trng_irq_b	L	level
160	spm_irq_b[0]	L	level
161	spm_irq_b[1]	L	level
162	spm_irq_b[2]	L	level
163	spm_irq_b[3]	L	level
164	irq_apxgpt	L	level
165	irrx_irq	H	level
166	arm_eint_irq	H	level
167	eint_direct_irq_b[0]	L	level
168	eint_direct_irq_b[1]	L	level
169	eint_direct_irq_b[2]	L	level
170	eint_direct_irq_b[3]	L	level
171	eint_direct_irq_b[4]	L	level
172	eint_direct_irq_b[5]	L	level
173	eint_direct_irq_b[6]	L	level
174	eint_direct_irq_b[7]	L	level
175	eint_direct_irq_b[8]	L	level
176	eint_direct_irq_b[9]	L	level
177	eint_direct_irq_b[10]	L	level
178	eint_direct_irq_b[11]	L	level
179	eint_event_b	L	level
180	cec_irq_b		
181	kp_irq_b	L	edge
182	sej_apxgpt_irq_b	L	Level

GIC ID	Interrupt source/name	Polarity	Trigger type
183	sej_wdt_irq_b	L	Level
184	accdet_irq_b	L	Level
207	hdmi_shell_irq_b	L	level
208	smi_larb0_irq_b	L	level
209	smi_larb1_irq_b	L	level
210	smi_larb2_irq_b	L	level
217	irq	H	level
218	pwr_irq_b	L	level
223	(Reserved)	-	-
224	(Reserved)	-	-
225	(Reserved)	-	-
226	bt_cvsd_int_b	L	level
227	conn2ap_btif_wakeup_out_b	L	level
228	wf_hif_int_b	L	level
229	conn_wdt_irq_b	L	edge
230	wdt_irq_b	L	edge
231	gcpu_irq_b	L	level
232	dmx_irq_b	L	level
233	gcpu_mmu_irq_b	L	level
234	gcpu_mmu_sec_irq_b	L	level
235	(Reserved)	-	-
236	pmic_wrap_int	L	Level
237	sys_timer_irq[0]	H	level
238	sys_timer_irq[1]	H	level
239	sys_timer_irq[2]	H	level
240	sys_timer_irq[3]	H	level
241	i2c3_irqb	L	level
242	usb_mcu_int_b_1p	L	level
243	uart2_irq_b	L	level
244	(Reserved)	-	-
245	(Reserved)	-	-
246	sys_cirq_event_b	L	level
247	mp_irq_oc[0]	H	level
248	mp_irq_oc[1]	H	level
249	mp_irq_oc[2]	H	level
250	mp_irq_oc[3]	H	level

4.1.5 Register Definitions

For register details refer to chapter 2.1 of “MT8516A Application Processor Registers.”

4.2 MCU Debug System (debugsys)

4.2.1 Introduction

MT8516A debug and trace system 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, we focus on target's debug and trace architecture design so that the existing DnT framework and the commercial ARM debugger can be used concurrently

4.2.2 References

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 specs can be found in the following ARM specifications:

- ARM Debug Interface V5.1
  Note: V5.1 is needed to be accurate for MT8516A. But V5 is ok as long as you don't need multi-drop SWD.
- 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 such as external emulators are able to halt and step processor cores, observe system state, and download code to the target device. The MT8516A 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 that 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, MDMCU, and DSP such that each processing system has its own communication channel. In practice, these can be used for RF calibration, test data, or generating stimulus to the target. The DCCs also share the same 2 pin interface as the emulators allowing for a highly efficient pin count.

4.2.4 Debug System Block Diagram

MT8516A DNT System is designed to provide debug and trace mechanism for CA35 MP cores. The debug and trace architecture is depicted in the following diagram.

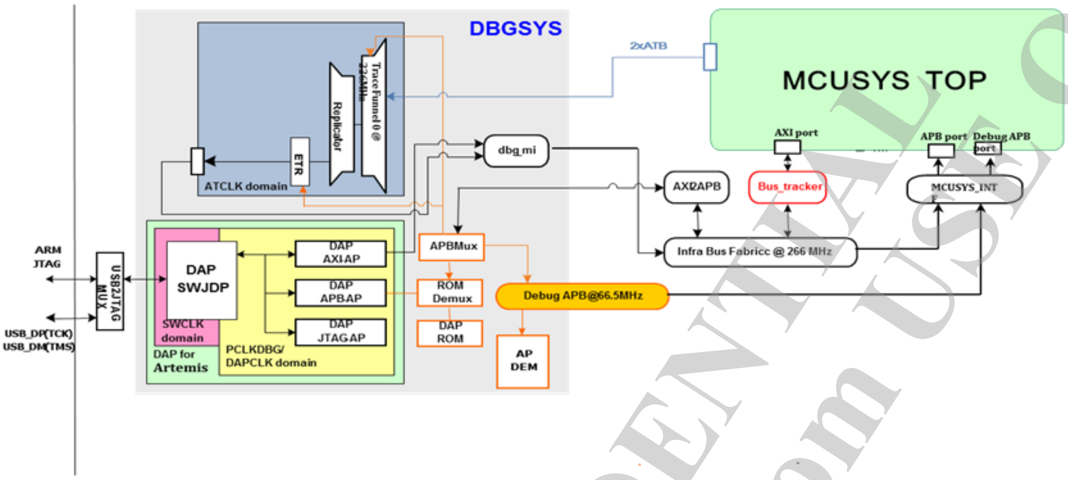


Figure 4-1. MT8516A Debug System Block Diagram

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

4.2.5 Application Processor (AP) Debug Subsystem

In general, modules not included in MD system belong to AP system. It includes APMCU subsystem, multimedia subsystem, peripherals and system bus fabric. The following list the clock frequency of major AP system bus fabric

Name	Frequency
AP AXI bus	66.5MHz
AP APB bus	66.5MHz
AP Debug APB bus	66.5MHz

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

4.2.6 Register Definitions

For register details refer to chapter 2.2 in "MT8516A Application Processor Registers."

4.3 System Interrupt Controller

4.3.1 Introduction

For processors like CA7 or CA9 which has embedded interrupt controllers (GIC), the part of the MCUSYS will need to keep feeding clock and power to make 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 become power ineffective. 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, the MCUSYS can be completely turned off to improve system power consumption without losing interrupts.

4.3.2 Features

SYS_CIRQ supports up to 146 interrupts which can configure following attributes individually.

- Polarity inversion
- Edge/level trigger selection

The 168 interrupts will feed through SYS_CIRQ and connect 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

Below is the system level block diagram of the system interrupt controller.

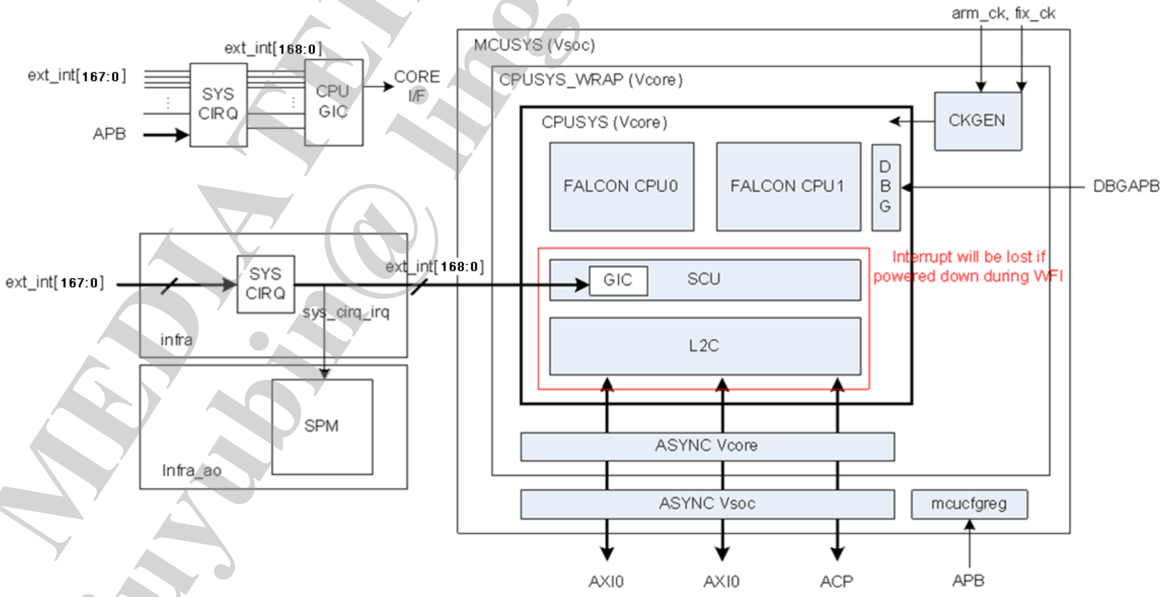


Figure 4-2. System Interrupt Controller System Level Block Diagram

The SYS_CIRQ controller is integrated in between MCUSYS and other interrupt sources as the second level interrupt controller. All interrupts are fed through SYS_CIRQ controller 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 the sleep mode, where GIC is power downed, the SYS_CIRQ controller will be enabled and monitor all edge-trigger interrupts (only edge-triggered interrupt will be lost in this scenario). When an edge-trigger interrupt is triggered, it will be recorded in SYS_CIRQ_STA register and can be restored to GIC by SW context restore or the SYS_CIRQ flush function.

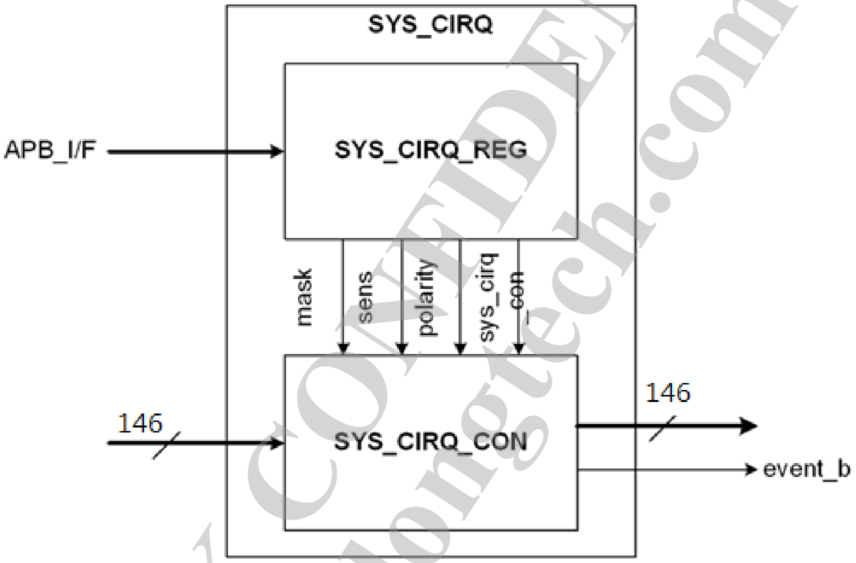


Figure 4-3. System Interrupt Controller Block Diagram

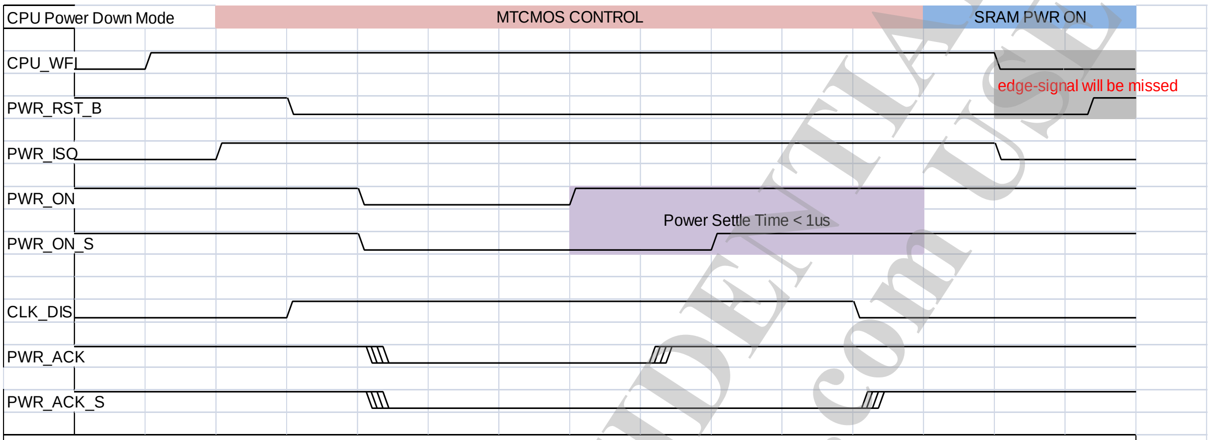
Figure 4-3 figure above shows the architecture of SYS_CIRQ. SYS_CIRQ_REG stores the mask/sensitivity/polarity attributes of each interrupt signal, and SYS_CIRQ_CON is used to mask and detect edge-triggered interrupts.

4.3.4 Register Definitions

For register details refer to chapter 2.3 of “MT8516A Application Processor Registers”.

4.3.5 Programming Guide

4.3.5.1 MCUSYS MTCMOS Sequence



Note:
1. MEM_CKISO should connect to PISO cell to isolate the CLOCK and CHIP SEL input ports of SRAM to LOW
2. All the MEM_PDN and MEM_SLEEP_B signals come from the SPM in always power on domain.
3. MEM_SLEEP_B should always keep HIGH when SRAM power down
4. All the signals are controled by CPU(Software) or SPM

4.4 External Interrupt Controller (EINTC)

4.4.1 Introduction

The 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 131 external interrupt signals and performs the following processes to the interrupt signals coming from external sources:

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

According to the register configuration, the external interrupt source will be forwarded to the Cortex-A7 built-in interrupt controller with different IRQ signals, eint_irq or eint_direct_irq. EINTC generates wakeup events to AP MCU.

4.4.3 EINTC Block Diagram

The external interrupt controller in MT8516A is shown here. Every functional block is controlled by the corresponding control registers defined in next section.

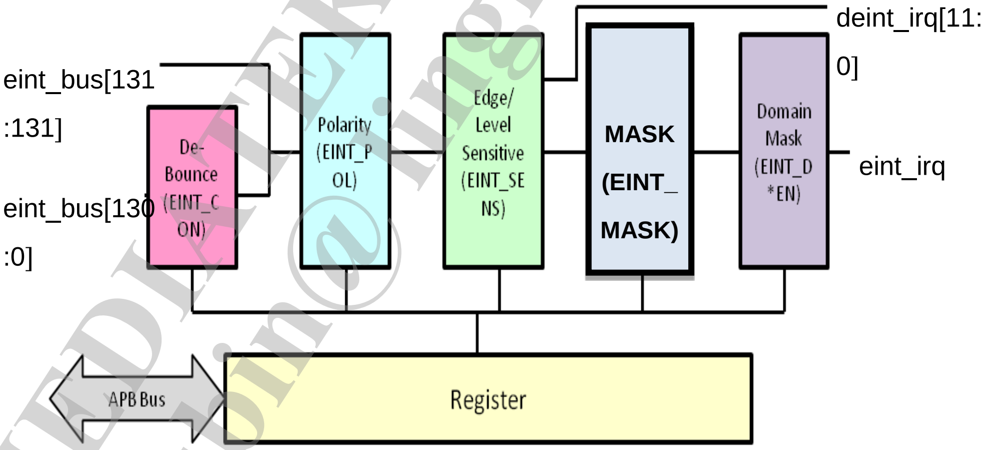


Figure 4-4. External Interrupt Controller Block Diagram

Normally the external interrupt source goes through the de-bounce unit which is driven by 32kHz clock and triggers the corresponding CPU with eint_irq. Therefore, the minimum latency from eint_bus to eint_irq is 30.52μs.

The following tables list the signal connections to the interrupt controller of CPU.

Table 4-2. External Interrupt Request Signal Connection

IRQ name	AP MCU INTC
eint_irq	IRQ[62]

Table 4-3. Domain Definitions

Domain number	Target CPU/DSP
0	Application CPU

4.4.4 Register Definitions

For register details refer to chapter 2.4 of “MT8516A Application Processor Registers.”

4.5 Infrastructure System Configuration Module (infrasys)

4.5.1 Introduction

The Infrastructure system contains bus fabric to connect all sub-systems like CPU, the Multimedia system, the Wi-Fi system, the peripheral system, etc. The Infrastructure system is also used to dispatch the register space for all function blocks, and we use the memory map to summarize all these space. The infrastructure system configuration module (INFRACFG) 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
- Top AXI bus fabric control signals
- Dynamic clock management control signals
- Dynamic clock management function

4.5.3 DCM Details

The dynamic clock management function is used to automatically slows down the clock frequency for power saving when the system is in idle state.

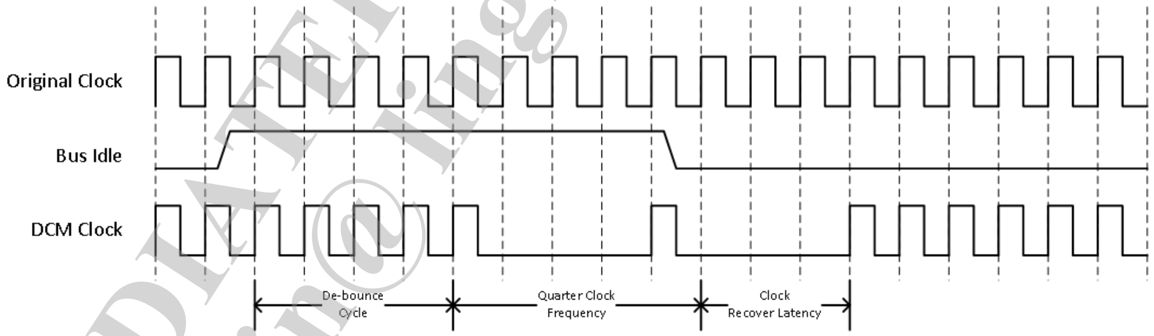


Figure 4-5below gives a sample clock waveform when DCM is activated. In this example, the clock frequency in DCM mode is set to quarter of the original clock. The ratio of clock frequency slow-down is controlled by the Top Clock Generator register rg_busdcmclk_ctrl (Refer to the Top Clock Generator registers in chapter 1.3 “MT8516A Application Processor Registers.”

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 issues. 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.

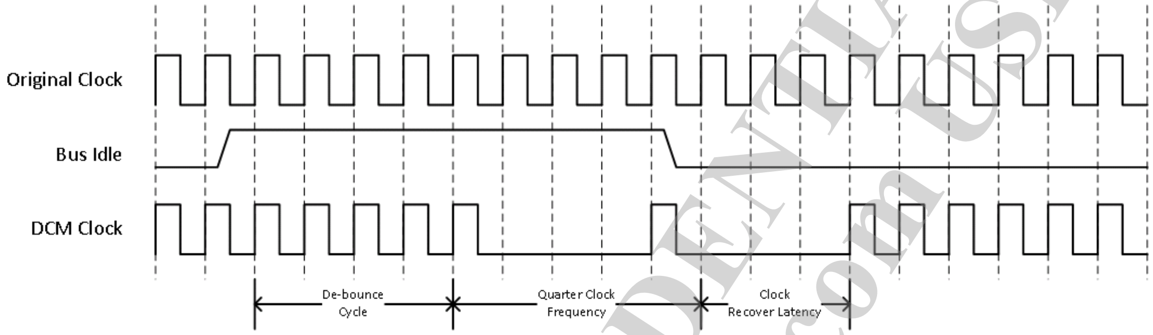


Figure 4-5. DCM in Action

4.5.4 AXI Fabric Control

The AXI fabric control registers help prevent the system bus from hanging up caused by improper access while some parts of the system are in the power-down state. See the figure below for the location of SI node 0 to SI node 2 and the sleep protector and find the corresponding control registers in 4.5.3 (referenced above).

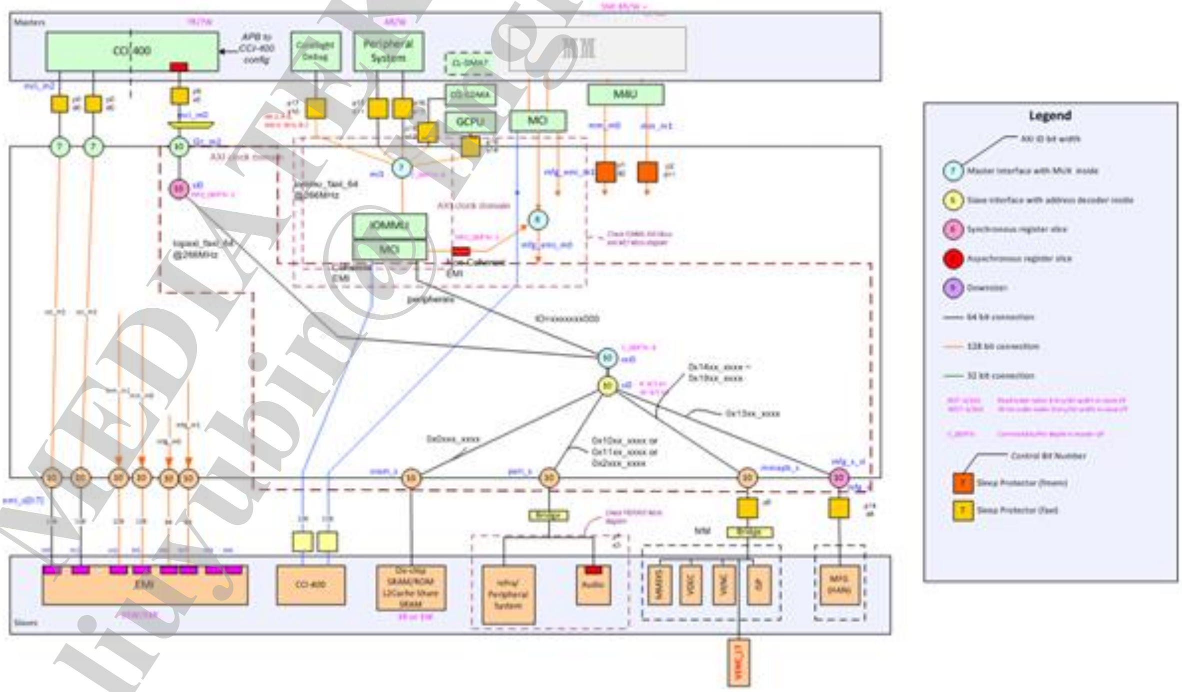


Figure 4-6. Top AXI Fabric and Control Blocks

4.5.5 MT8516A Memory Maps

Table 4-4. MT8516A Top Memory Map

Bank	Start Address	End Address	Attribute	Size	Devices
0	0x0000_0000	0x0001_7FFF	Normal	96KB	Boot ROM
	0x0001_8000	0x000F_FFFF	-		Reserved
	0x0010_0000	0x0010_FFFF	Normal	64KB	On-Chip SRAM
	0x0011_0000	0x001F_FFFF	-		Reserved
	0x0020_0000	0x0023_FFFF	Normal	256KB	Share SRAM
	0x0028_0000	0x03FF_FFFF	Reserved for MCUSYS		Reserved for MCUSYS
	0x0400_0000	0x07FF_FFFF	-		Reserved
	0x0800_0000	0x0800_000F	Device Memory	16B	Chip ID/HW, SW Version
	0x0800_0010	0x0FFF_FFFF	-		Reserved
1	0x1000_0000	0x10FF_FFFF	Depends on Settings	16MB	Infrastructure, Mixmode & MCU System
	0x1100_0000	0x11FF_FFFF	Depends on Settings	16MB	Peripheral System
	0x1200_0000	0x12FF_FFFF	Depends on Settings	16MB	Reserved
	0x1700_0000	0x17FF_FFFF	Depends on Settings	16MB	Reserved
	0x1800_0000	0x18FF_FFFF	Depends on Settings	16MB	CONN System
	0x1900_0000	0x1CFF_FFFF	Depends on Settings	64MB	NOR Flash
	0x1D00_0000	0x1FFF_FFFF	Depends on Settings	48MB	Reserved
2	0x2000_0000	0x2FFF_FFFF	Depends on Settings	256MB	Reserved
3	0x3000_0000	0x3FFF_FFFF	Depends on Settings	256MB	Reserved
4	0x4000_0000	0x4FFF_FFFF	Depends on Settings	3G	EMI
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			

Bank	Start Address	End Address	Attribute	Size	Devices
A	0xA000_0000	0xAFFF_FFFF			
B	0xB000_0000	0xBFFF_FFFF			
C	0xC000_0000	0xCFFF_FFFF			
D	0xD000_0000	0xDFFF_FFFF			
E	0xE000_0000	0xEFFF_FFFF			
F	0xF000_0000	0xFFFF_FFFF			

Table 4-5. Infrastructure System Memory Map

Start Address	End Address	Size	Module
0x1000_0000	0x1000_0FFF	4KB	topckgen
0x1000_1000	0x1000_1FFF	4KB	infracfg_ao
0x1000_2000	0x1000_2FFF	4KB	kp
0x1000_3000	0x1000_3FFF	4KB	pericfg
0x1000_4000	0x1000_4FFF	4KB	reserved
0x1000_5000	0x1000_5FFF	4KB	gpio
0x1000_6000	0x1000_6FFF	4KB	spm
0x1000_7000	0x1000_7FFF	4KB	toprgu
0x1000_8000	0x1000_8FFF	4KB	apxgpt
0x1000_9000	0x1000_9FFF	4KB	efusec
0x1000_A000	0x1000_AFFF	4KB	sej
0x1000_B000	0x1000_BFFF	4KB	ap_cirq_eint
0x1000_C000	0x1000_CFFF	4KB	reserved
0x1000_D000	0x1000_DFFF	4KB	sys_timer
0x1000_E000	0x1000_EFFF	4KB	reserved
0x1000_F000	0x1000_FFFF	4KB	pmic_wrap
0x1001_0000	0x1001_0FFF	4KB	infra_ao_device_apc
0x1001_1000	0x1001_1FFF	4KB	reserved
0x1001_2000	0x1001_2FFF	4KB	infra_top_mbist_ctrl
0x1001_3000	0x1001_3FFF	4KB	reserved
0x1001_4000	0x1001_4FFF	4KB	io_config t
0x1001_5000	0x1001_5FFF	4KB	io_config b
0x1001_6000	0x1001_6FFF	4KB	io_config l
0x1001_7000	0x1001_7FFF	4KB	io_config r
0x1001_8000	0x1001_8FFF	4KB	apmixed
0x1001_8F00	0x1001_8FFF	256B	fhctl
0x1001_9000	0x1001_9FFF	4KB	irrx
0x1001_A000	0x1001_AFFF	4KB	cec
0x1001_B000	0x1001_BFFF	4KB	addcet

Table 4-6. Infrastructure System Memory Map

Start Address	End Address	Size	Module
0x1020_0000	0x1020_0FFF	4KB	mcusys_cfgreg
0x1020_1000	0x1020_1FFF	4KB	mcusys_cfgreg2
0x1020_2000	0x1020_2FFF	4KB	sys_cirq
0x1020_3000	0x1020_3FFF	4KB	m4u_cfgreg
0x1020_4000	0x1020_4FFF	4KB	device_apc
0x1020_5000	0x1020_5FFF	4KB	emi
0x1020_6000	0x1020_6FFF	4KB	dramc
0x1020_7000	0x1020_7FFF	4KB	dramc_conf
0x1020_8000	0x1020_8FFF	4KB	ddrphy
0x1020_9000	0x1020_9FFF	4KB	sramrom
0x1020_A000	0x1020_AFFF	4KB	gce
0x1020_B000	0x1020_BFFF	4KB	bus_dbg_tracker
0x1020_C000	0x1020_CFFF	4KB	trng
0x1020_D000	0x1020_DFFF	4KB	gcpu
0x1020_E000	0x1020_EFFF	4KB	gcpu_mmu
0x1020_F000	0x1020_FFFF	4KB	infracfg_reg
0x1021_0000	0x1021_0FFF	4KB	BSI
0x1021_1000	0x1021_1FFF	4KB	DDRPHY_conf1
0x1021_2000	0x1021_2FFF	4KB	DDRPHY_conf2

Table 4-7. Peripheral System Memory Map

Start Address	End Address	Size	Module
0x1100_0000	0x1100_0FFF	4KB	ap_dma
0x1100_1000	0x1100_1FFF	4KB	nfi
0x1100_2000	0x1100_2FFF	4KB	nfi
0x1100_3000	0x1100_3FFF	4KB	auxadc
0x1100_4000	0x1100_4FFF	4KB	Reserved
0x1100_5000	0x1100_5FFF	4KB	uart
0x1100_6000	0x1100_6FFF	4KB	uart
0x1100_7000	0x1100_7FFF	4KB	uart
0x1100_8000	0x1100_8FFF	4KB	pwm
0x1100_9000	0x1100_9FFF	4KB	i2c
0x1100_A000	0x1100_AFFF	4KB	i2c
0x1100_B000	0x1100_BFFF	4KB	i2c
0x1100_C000	0x1100_CFFF	4KB	spi
0x1100_D000	0x1100_DFFF	4KB	therm_ctrl
0x1100_E000	0x1100_EFFF	4KB	BTIF
0x1100_F000	0x1100_FFFF	4KB	Reserved
0x1101_0000	0x1101_0FFF	4KB	flashif
0x1101_1000	0x1101_1FFF	4KB	hdmi_sifm

Start Address	End Address	Size	Module
0x1101_2000	0x1101_2FFF	4KB	i2c
0x1101_3000	0x1101_3FFF	4KB	Reserved

Table 4-8. Peripheral System Map

Start Address	End Address	Size	Module
0x1110_0000	0x1110_FFFF	64KB	USB0
0x1111_0000	0x1111_FFFF	64KB	USBSIF
0x1112_0000	0x1112_FFFF	64KB	MSDC0
0x1113_0000	0x1113_FFFF	64KB	MSDC1
0x1114_0000	0x1114_FFFF	64KB	Audio AHB slave
0x1115_0000	0x1115_FFFF	64KB	AHB MON ABORT
0x1116_0000	0x1116_FFFF	64KB	Reserved
0x1117_0000	0x1117_FFFF	64KB	MSDC2
0x1118_0000	0x1118_FFFF	64KB	ethernet
0x1119_0000	0x1119_FFFF	64KB	USB0
0x111A_0000	0x111A_FFFF	64KB	STC

4.5.6 Multimedia System Memory Maps

Table 4-9. Audio System Memory Map

Start Address	End Address	Size	Devices
0x1114_0000	0x1114_FFFF	64KB	Audio system configuration
0x1401_9000	0x1401_9FFF	4KB	Reserved
0x1401_A000	0x1401_AFFF	4KB	Reserved
0x1401_B000	0x1401_BFFF	4KB	HDMI

4.5.7 Register Definitions

For register details refer to chapter 2.5 of “MT8516A Application Processor Registers.”

4.6 On-Chip Memory Controller

The on-chip memory controller provides 96KB boot ROM and 64KB SRAM resources. The following table is the memory map of boot ROM and on-chip SRAM.

Table 4-10. On-chip Memory Controller Memory Map

Bank	Start address	End address	Size	Device
0	0x0000_0000	0x0001_7FFF	96B	Boot ROM
	0x0010_0000	0x0010_FFFF	64KB	On-chip SRAM
	0x0800_0000	0x0800_000F	16B	Chip ID/HW, SW version

4.6.1 On-Chip Memory Controller Block Diagram

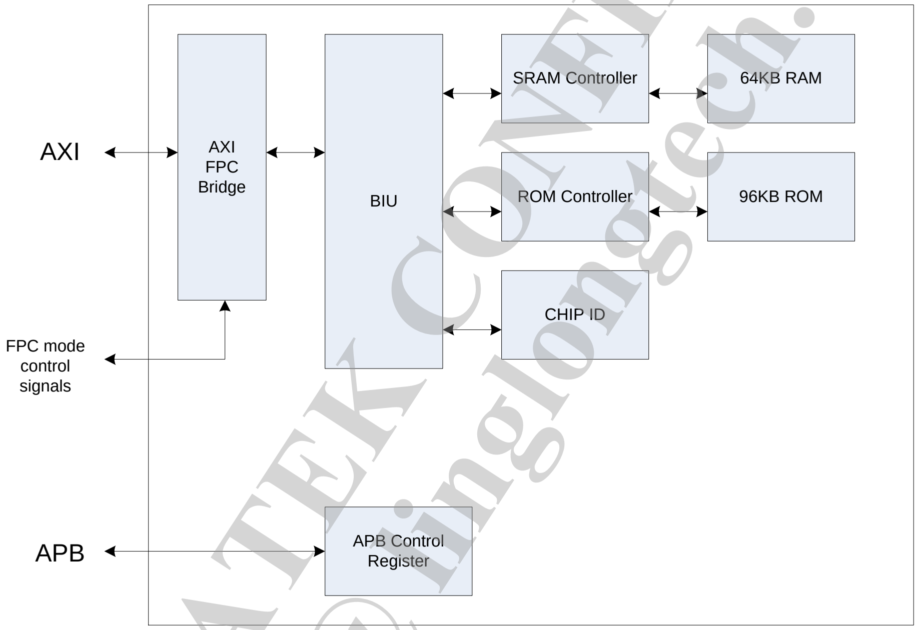


Figure 4-7. On-Chip Memory Controller Block Diagram

The on-chip memory controller consists of a SRAM controller, a ROM controller, an AXI-FPC bus bridge, a bus interface unit, setting register and chip ID unit. Detailed functionality is described in the following sections.

4.6.2 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 MCDI (multi-core-deep-idle), it is the bootstrap for suspend/resume CPU

4.6.3 BOOT ROM FPC Mode

Boot ROM FPC mode is mainly used in Function Pattern mode. When the chip is trapped into 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.

4.7 External Memory Interface (EMI)

4.7.1 Introduction

External Memory Interface (EMI) is a sophisticated protocol communication interface between external memory and other devices. The EMI controller schedules requests from the masters and issues commands to DRAMC in an efficiency way. The EMI controller conducts flow control for DRAMC and masters to avoid DRAM stall or data overflow or underflow. It also minimizes the latency of processor path to enhance the performance and increase the DRAM efficiency. To reduce power, the clock of the EMI controller is gated when EMI does not find any transaction.

4.7.2 Features

The EMI controller receives AXI master commands and issues them to the DRAM controller. It supports all AXI transaction type commands except for the fixed and cache commands. There are plenty of schedule options to schedule the command, which are:

- Starvation control
- Bandwidth limiter
- High priority
- Page hit control
- Read/write turn around prevent control

4.7.3 EMI Block Diagram

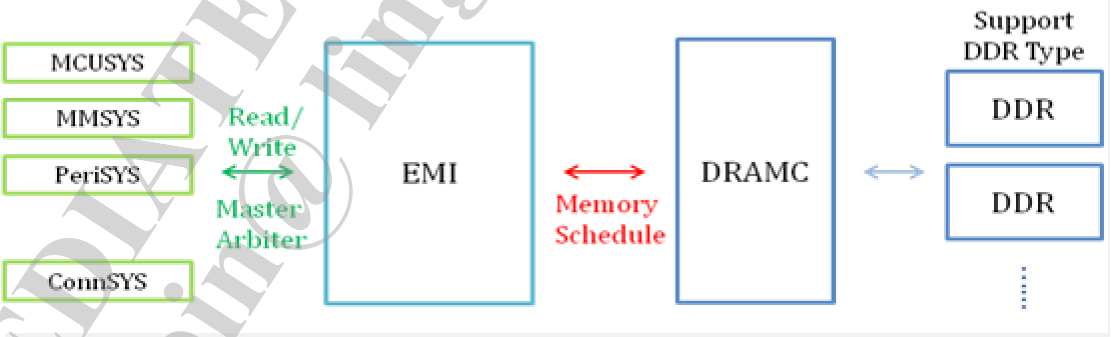


Figure 4-8. EMI/DRAM Controller Top Connection

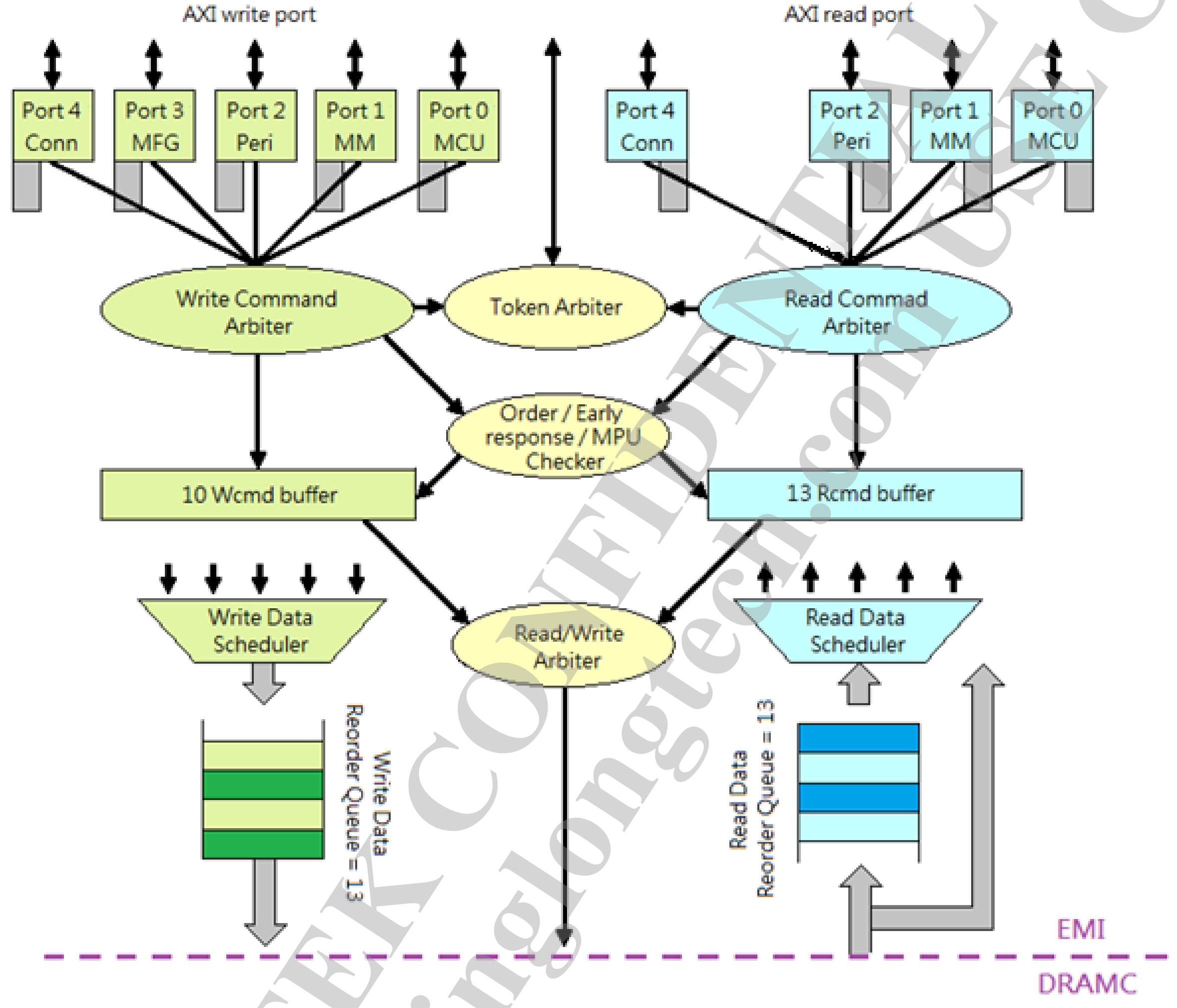


Figure 4-9. EMI Architecture

In this project, the EMI controller consists of AXI slaves, read/write command buffers, read/write data buffers, collector arbiters, read/write arbiter, command sequencers. The EMI controller connects five AXI ports and supports connecting two rank DRAM devices. For MCU, multimedia systems, 128-bit AXI ports are provided for the connection. Besides, two 64-bit AXI ports are provided for connecting to peripheral system and connectivity system.

4.7.4 Theory of Operations

EMI is a sophisticated protocol communication interface which also consists of AXI slaves, read/write buffers, collector arbiters, channel arbiters, command sequencers, and DRAM controllers. The degree of efficiency in which EMI can access memory will contribute a certain percentage to the overall system performance.

This EMI adopts SEDA (Scalable EMI/DRAMC Architecture). On the input side of the EMI, we have AXI slave ports which connect to AXI masters within the system. Slave ports connect to read/write command buffer arbiter. Read collector arbiter with corresponding thirteen collector buffers for read transactions are used and write collector arbiter with corresponding ten collector buffers for read transactions are used.

The command collector buffers are for command buffering, as well as priority encoding. After the read command or write command with write data are received, the buffer will issue the request signal to the channel arbiter. MI stores read/write data in SRAM to reduce area.
Channel arbiter uses the read/write buffers encoded priority code for arbitration. The arbiter will decode the priority and select one command to send it into DRAM controller. SEDA EMI enhance read/write grouping algorithm to improve DDR efficiency.

Inside the DRAM controller, a command sequencer is used to sequence the commands in execution. The command sequencer will follow DRAM protocols to issue all transaction. And the commands will transfer to external memories via PHY. In DRAM controller, we use APB interface to program registers. Then we can do DRAM initialization or other parameters setting.

4.7.5 Register Definitions

For register details refer to chapter 2.6 of “MT8516A Application Processor Registers.”

4.7.6 Programming Guide

To start EMI function, we have to program the settings in the following manner.

- Program supporting one channel DRAM.
- According to DRAM type, programing DRAM address mapping types like column bit number, row bit number, and bank bit number. And we have to put address map as “rank, row, ban, col” or “rank, row, ban[1:0], bank_group, col”.
- We have to set the DRAM bit number as 32 or 16 bits data bus.
- Program the latency for each AXI interface that we will set the request as high priority when the age counter expires. (In the beginning, we treat all the requests from one AXI port as the same ID.)
- Allocate the bandwidth requirement for each AXI port, and set the bandwidth limiter value. And we have to guarantee the total bandwidth that we set doesn’t exceed one hundred percent.
- Set the bandwidth limiter for each AXI ports as soft limit or hard limit.
- Program the security region addresses and numbers.

4.8 DRAM Controller (DRAMC)

4.8.1 Introduction

The DRAMC module can process the command which come from EMI to DRAM , and take special control to the DRAM.

4.8.2 Features

MT8516A supports DRAM type:

- LPDDR2 32bit
- LPDDR3 32bit
- DDR3 16bit/32bit
- DDR3 Asymmetry 512M+256M
- DDR4 32bit

4.8.3 Block Diagram

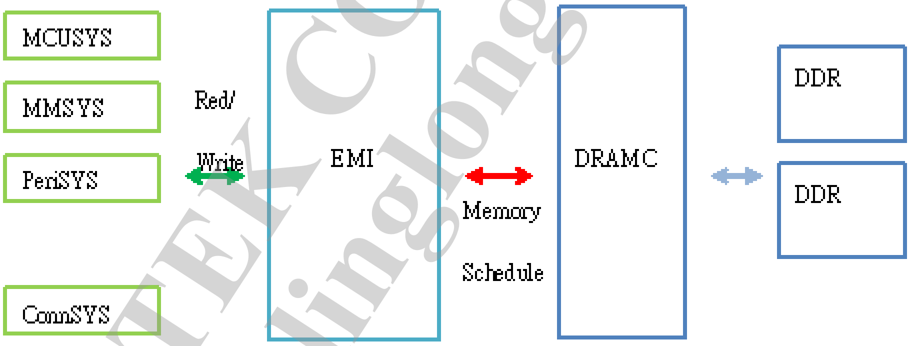


Figure 4-10. EMI/DRAM Controller Top Connection

The major DRAM controller blocks are command decoder, command pool, bus scheduler, timing controller DDR PHY and response generator.

The requests from Arbiter are pushed to command pool to wait for execution in order. The bus scheduler inspects the precharge/active pool and command FIFO and decides which DRAM bus command, e.g. PRECHARGE, ACTIVE, READ or WRITE, is issued to the DRAM bus. The goals of bus scheduler are to raise the bus utilization rate and lower the response latency. The timing control unit is responsible for the integrity of DRAM bus timing such as pre-charge to active delay (tRP), active to command delay (tRCD) and bus turnaround time. The bus scheduler refers to the information and choose the next DRAM bus command. The DRAM interface unit is responsible for generating DRAM bus commands, transmitting data and DQS to DDR DRAM and receiving data and DQS from DDR

DRAM. The response generator produces the response signals for all DRAM agents such as DLE and RDAT.

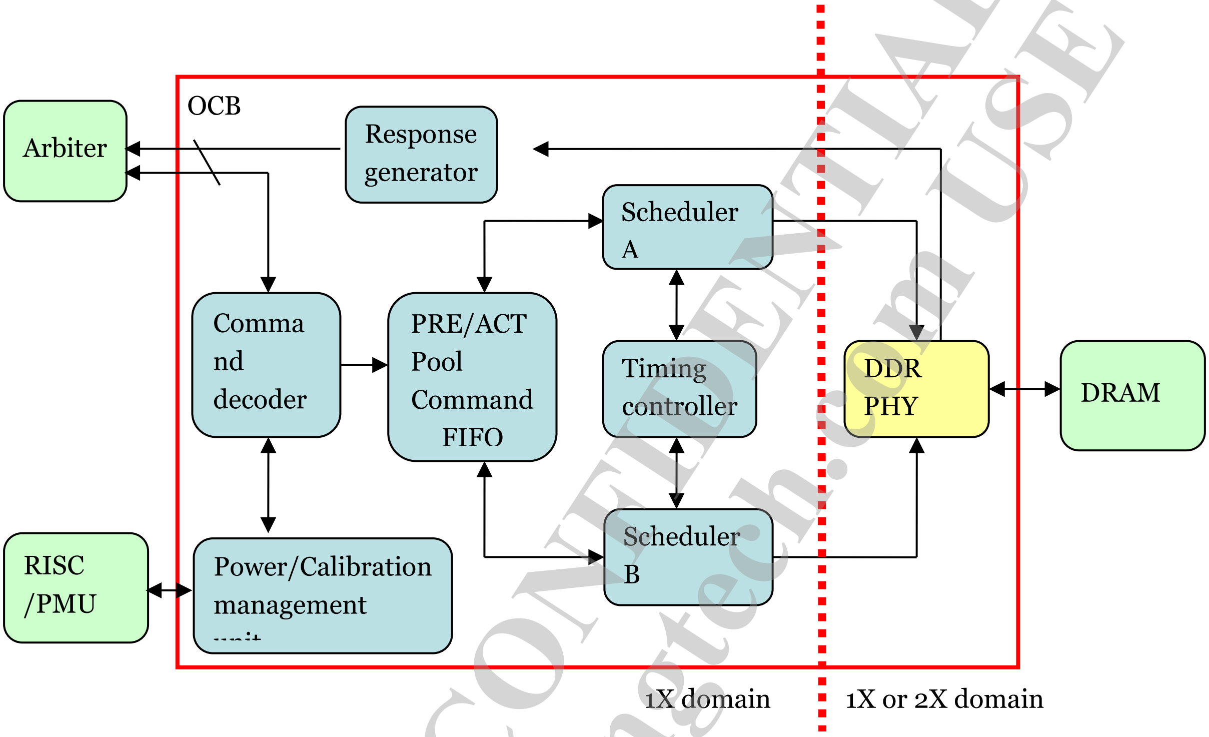


Figure 4-2. DRAM Controller Block Diagram

4.8.4 Theory of Operations

DRAMC reads command from EMI, and then reads/writes data from/to DRAM according to the command.

4.8.5 Register Definitions

For register details refer to chapter 2.7 of “MT8516A Application Processor Registers”.

4.8.6 Programming Guide

To start DRAMC function, program the settings in the following manner.

- Set up DRAM AC timing parameter.
- Follow DRAM spec to complete DRAM initialization including mode register programming.
- Enable calibration for DQ/DQS window.
- Set up refresh rate counter.
- Normal operation.

4.9 DDRPHY

4.9.1 Introduction

The DDRPHY module processes the command to adapt the protocol of DRAM interface, including signals and their sequence.

See the table below for the DRAM bus signals:

Table 4-11. DRAM Bus Signal List (refer to DRAMC side)(LPDDR3/LPDDR2)

Signal name	Type	Description
CK0/CK1	Input	DRAM clock signal
CK0#/CK1#	Input	DRAM clock invert signal
MA[9:0]	Input	Address for all memories/CA bus for LPDDR3
CKE/CKE1	Input	Clock enable signal for DRAM
CS# [1:0]	Input	RANK1~RANK0 selection signal
DQ[31:0]	I/O	Data bus for LPDDR3
DQM[1:0]	Input	Data mask
DQS[3:0]	I/O	Data strobe
DQS#[3:0]	I/O	Differential data strobe in LPDDR3
REXTDN	I/O	Output driving calibration

Table 4-12. DRAM Bus Signal List (refer to DRAMC side)(DDR4)(16bit DRAM)

Signal name	Type	Description
CK0	Input	DRAM clock signal
CK0#	Input	DRAM clock invert signal
MA[13:0]	Input	Address for all memories/CA bus
RAS_n	Input	Command signal
CAS_n/A15	Input	Command signal or address[15]
WE_n/A14	Input	Command signal or address[14]
ACT_n	Input	For Active command
BA[1:0]	Input	Bank address
BG[0]	Input	Bank group address
CKE	Input	Clock enable signal for DRAM
CS#	Input	selection signal
DQ[31:0]	I/O	Data bus
DQM[3:0]	Input	Data mask
DQS[3:0]	I/O	Data strobe
DQS#[3:0]	I/O	Differential data strobe
REXTDN	I/O	Output driving calibration

Table 4-13. DRAM Bus Signal List (refer to DRAMC side)(DDR3)

Signal name	Type	Description
CK0	Input	DRAM clock signal
CK0#	Input	DRAM clock invert signal
MA[15:0]	Input	Address for all memories/CA bus
RAS_n	Input	Command signal
CAS_n	Input	Command signal
WE_n	Input	Command signal
BA[2:0]	Input	Bank address
CKE	Input	Clock enable signal for DRAM
CS#	Input	selection signal
DQ[31:0]	I/O	Data bus
DQM[3:0]	Input	Data mask
DQS[3:0]	I/O	Data strobe
DQS#[3:0]	I/O	Differential data strobe
REXTDN	I/O	Output driving calibration

See below for the DRAM bus command truth table:

Table 4-14. DRAM Bus Command Truth Table (LPDDR3)

		SDR Command Pins			DDR CA pins (10)										
SDRAM Command	NVM Command	CKE		CS_N	CA0	CA1	CA2	CA3	CA4	CA5	CA6	CA7	CA8	CA9	CK_t EDGE
		CK_t(n-1)	CK_t(n)												
MRW	MRW	H	H	L	L	L	L	L	MA0	MA1	MA2	MA3	MA4	MA5	↑
				X	MA6	MA7	OP0	OP1	OP2	OP3	OP4	OP5	OP6	OP7	↓
MRR	MRR	H	H	L	L	L	L	H	MA0	MA1	MA2	MA3	MA4	MA5	↑
				X	MA6	MA7	X								↓
Refresh (per bank)[11]	-	H	H	L	L	L	H	L	X						↑
				X	X										↓
Refresh (all bank)	-	H	H	L	L	L	H	H	X						↑
				X	X										↓
Enter Self Refresh	Enter Power Down	H	L	L	L	L	H	X							↑
		X		X	X										↓
Activate (bank)	Activate (row buffer)	H	H	L	L	H	R8/a15	R9/a16	R10/a17	R11/a18	R12/a19	BA0	BA1	BA2	↑
				X	R0/a5	R1/a6	R2/a7	R3/a8	R4/a9	R5/a10	R6/a11	R7/a12	R13/a13	R14/a14	↓
Write (bank)	Write (RDB)	H	H	L	H	L	L	RFU	RFU	C1	C2	BA0	BA1	BA2	↑
				X	AP[3,4]	C3	C4	C5	C6	C7	C8	C9	C10	C11	↓
Read (bank)	Read (RDB)	H	H	L	H	L	H	RFU	RFU	C1	C2	BA0	BA1	BA2	↑
				X	AP[3,4]	C3	C4	C5	C6	C7	C8	C9	C10	C11	↓
Precharge (pre bank, all bank)	Preactive (RAB)	H	H	L	H	H	L	H	AB/a30	X/a31	X/a32	BA0	BA1	BA2	↑
				X	X/a20	X/a21	X/a22	X/a23	X/a24	X/a25	X/a26	X/a27	X/a28	X/a29	↓
BST	BST	H	H	L	H	H	L	L	X						↑
				X	X										↓
Enter Deep Power Down	Enter Power Down	H	L	L	H	H	L	X							↑
		X		X	X										↓
NOP	NOP	H	H	L	H	H	H	X							↑
				X	X										↓
Maintain PD, SREF, DPD (NOP)	Maintain Power Down (NOP)	L	L	L	H	H	H	X							↑
				X	X										↓
NOP	NOP	H	H	H	X										↑
				X	X										↓
Maintain PD, SREF, DPD (NOP)	Maintain Power Down (NOP)	L	L	H	X										↑
				X	X										↓
Enter Power Down	Enter Power Down	H	L	H	X										↑
		X		X	X										↓
Exit PD, SREF, DPD	Exit Power Down	L	H	H	X										↑
		X		X	X										↓

These tables are applied when CKE is asserted at the clock cycle before CS# is asserted. Read and write accesses to the DDR SDRAM are burst-oriented. The accesses start at a selected location and continue for a programmed number of locations in a programmed sequence. The accesses begin with the registration of an ACTIVE command, followed by a READ or WRITE command. The address bits registered coincident with the ACTIVE command are used to select the bank and row to be accessed. The address bits registered coincident with the READ or WRITE command are used to select the bank and the starting column location for the burst access.

As with standard SDRAMs, the pipelined, multi-bank architecture of DDR SDRAMs allows for concurrent operation, thereby providing high effective bandwidth by hiding row precharge and activation time.

The DDR SDRAM operates from a differential clock (CK and CK#). Commands (address and control signals) are registered at every positive and negative edges of CK for LPDDR3. The input data are registered on both edges of DQS, and the output data are referenced to both edges of DQS, as well as to both edges of CK. DQS is center-aligned with data for WRITEs. Without DLL inside mobile DRAM's (LPDDR3), DQS is not edge-aligned with data for READs.

The commands for LPDDR3 SDRAM are encoded in MA0 ~ MA9 and transfer at double rate of clock frequency such as DQ.

4.9.2 References

- LPDDR3 spec: http://www.jedec.org/download/search/JESD209-3C.pdf
- DDR4 spec: http://www.jedec.org/standards-documents/docs/jesd79-4a
- DDR3 spec: http://www.jedec.org/standards-documents/docs/jesd-79-3d

4.9.3 Features

MT8516A supports DRAM type:

-	LPDDR2	32bit		@1066M
-	LPDDR3	32bit		@1600M
-	DDR3	16bit/32bit		@1600M
-	DDR3	Asymmetry	512M+256M	@1600M
-	DDR4	32bit		@1600M

4.9.4 Block Diagram

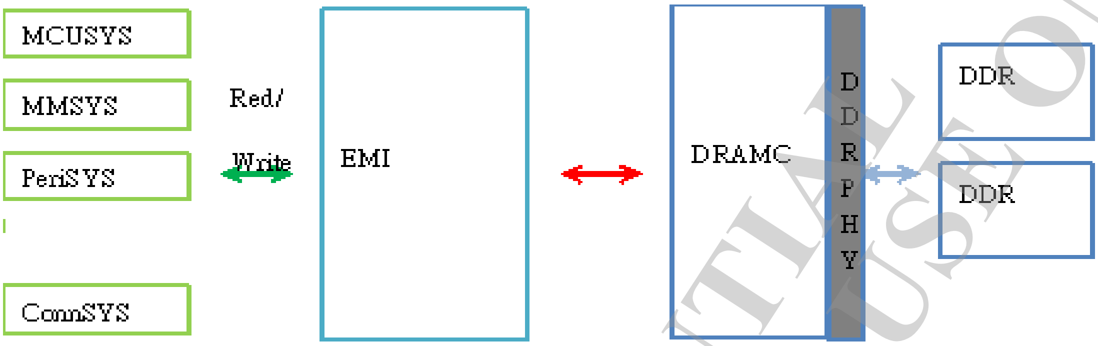


Figure 4-11. EMI/DRAM Controller Top Connection

DDRPHY connects DRAMC and DRAM. The major blocks of DDRPHY are command process module, data process module and PLL. PLL provides clock for DRAMC/EMI and DDRPHY, CA process module processes the command to DRAM PAD ,and DQ process module translates the write data and receives the rx data from DRAM.

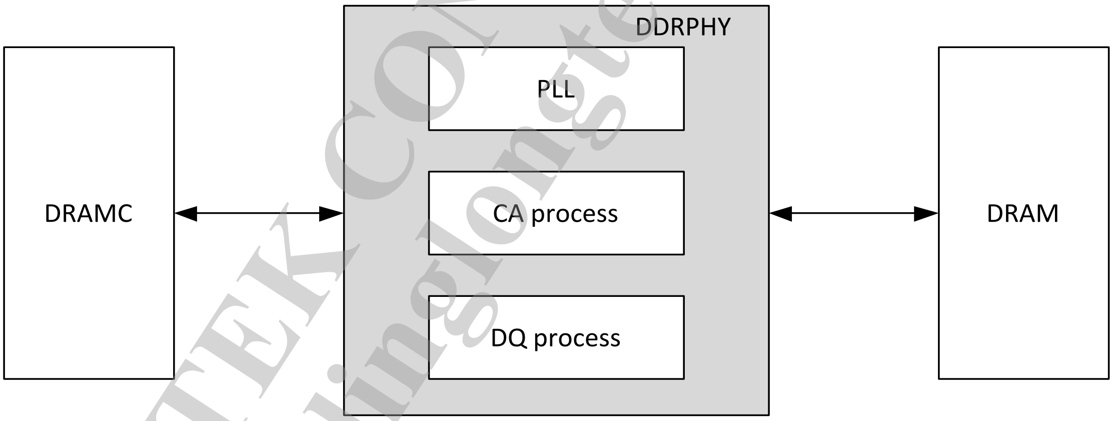


Figure 4-12. DDRPHY Block Diagram

4.9.5 Theory of Operations

Process the data rate to 1600M.

4.9.6 Register Definitions

For register details refer to chapter 2.8 of “MT8516A Application Processor Registers.”

4.9.7 Programming Guide

To start DRAMC function, we have to program the settings in the following manner.

- Set up DRAM AC timing parameter.
- Follow DRAM spec to complete DRAM initialization including mode register programming.
- Enable calibration for DQ/DQS window.
- Set up refresh rate counter.
- Normal operation.

4.10 AP_DMA

4.10.1 Introduction

There is always a DMA in a platform. The purpose of DMA is performing data transfer between different slaves. There are several slaves in a platform, and the major one is external memory, e.g. DRAM. There are also internal SRAM and some slave ports for the peripheral to transfer data. For saving software efforts, DMA delivers a virtual FIFO concept to help the software maintain read and write pointer when the software accesses data from a ring buffer. As the bus goes more and more efficient, the old DMA still utilizes the AHB bus protocol and may decrease its performance. Another problem is that when the old DMA meets byte alignment addresses or byte alignment sizes, it will need some software efforts to help solve head and tail non word alignment problems or let DMA to simply issues single-1-byte requests to conquer the byte-alignment problem. This will harm the overall system because the single-1-byte transaction is quite inefficient. The DMA efficiency is now improved by increasing its bus efficiency, including data buffering and overcoming byte alignment problems.

4.10.2 Features

APDMA has the following DMA engines.

- GDMA DMA engine*2
- I2C DMA engine*3
- BTIF TX DMA engine*1
- BTIF RX DMA engine*1
- UART1 TX DMA engine*1
- UART1 RX DMA engine*1
- UART2 TX DMA engine*1
- UART2 RX DMA engine*1
- UART3 TX DMA engine*1
- UART3 RX DMA engine*1
- HIF1 DMA engine*1

The DMA engines and corresponding peripheral devices are listed below.

Table 4-15. Relationship between Engines and Devices

Engine	Peripheral device
GDMA1	-
GDMA2	Connsys(sdctl)
I2C_1 ~ I2C_3	I2C_1 ~ I2C_3
UART1 ~ UART3	UART1 ~ UART3
BTIF	BTIF
HIF1	MDI slave

4.10.3 AP_DMA Block Diagram

The figure below is the basic AP_DMA block diagram. There are total 14 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 is re-directed to related peripherals, e.g. HIF, 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 both global registers and local registers existing in every individual DMA channel.

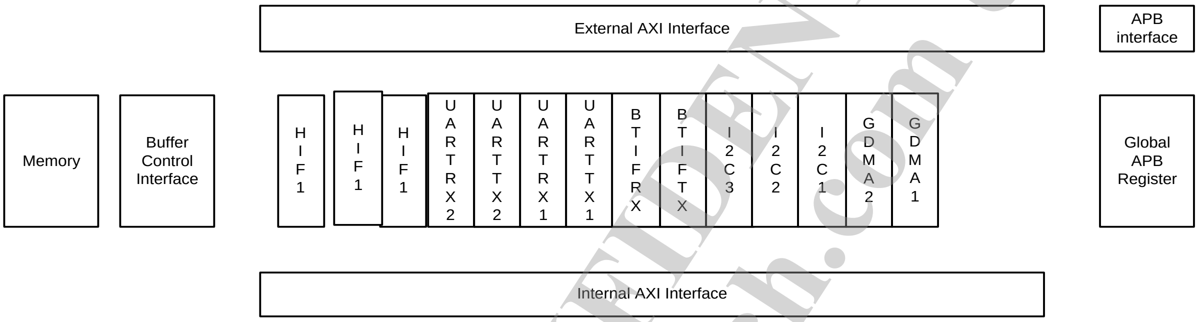


Figure 4-13. AP_DMA Block Diagram

4.10.4 Register Definitions

For register details refer to chapter 2.9 of “MT8516A Application Processor Registers.”

4.10.5 Programming Guide

4.10.5.1 Peripheral DMA with Burst Length equals 1 (e.g. I2C)

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)
    - Set interrupt enable=1 .
- AP_P_DMA_I2C_*_INT_EN
    - Wait for interrupt.

Clear interrupt flag.

- AP_P_DMA_I2C_*_INT_FLAG

4.10.5.2 Peripheral DMA with Configurable Burst Length (e.g. HIF)

Configure DMA registers.

- AP_P_DMA_HIF_*_CON (dir, burst_length)

- AP_P_DMA_HIF_*_MEM_ADDR
- AP_P_DMA_HIF_*_LEN

Set interrupt enable=1 .

- AP_P_DMA_HIF_*_INT_EN
    - ○ Set enable=1.
- AP_P_DMA_HIF_*_EN

    Wait for interrupt.

    Clear interrupt flag.
- AP_P_DMA_HIF_*_INT_FLAG

4.10.5.3 Virtual FIFO DMA TX (ex. UART_TX, BTIF_TX)

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

    Write data to EMI and update SW write_pointer.
- AP_P_DMA_UART_*_TX_VFF_WPT

    Clear interrupt (repeat step 3-6 till finished).
- AP_P_DMA_UART_*_TX_INT_FLAG

    Set interrupt enable =1.
- AP_P_DMA_UART_*_TX_INT_EN

    Set enable=1 (first time).
- AP_P_DMA_UART_*_TX_EN

    Wait for interrupt.

    Set stop=1 if finished.
- AP_P_DMA_UART_*_STOP

4.10.5.4 Virtual FIFO DMA RX (e.g. UART_RX, BTIF_RX)

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

    Set interrupt enable =1.
- AP_P_DMA_UART_*_RX_INT_EN

    Set enable=1 (first time).
- AP_P_DMA_UART_*_RX_EN

  Wait for interrupt (repeat step 4-6 till finished).

  Read data from EMI; update SW read_pointer.
- AP_P_DMA_UART_*_RX_VFF_RPT

  Clear interrupt flag.
- AP_P_DMA_UART_*_RX_INT_FLAG

  Set stop=1 if finished.

- AP_P_DMA_UART_*_RX_STOP

4.10.5.5 General DMA

Configure DMA registers.

- AP_P_DMA_G_DMA_*_CON (dir, burst length)
- AP_P_DMA_G_DMA_*_SRC_ADDR
- AP_P_DMA_G_DMA_*_DST_ADDR
- AP_P_DMA_G_DMA_*_LEN

  Set interrupt enable=1 .
- AP_P_DMA_G_DMA_*_INT_EN

  Set enable=1.
- AP_P_DMA_G_DMA_*_EN

  Wait for interrupt.

  Clear interrupt flag.
- AP_P_DMA_G_DMA_*_INT_FLAG

4.11 Blue Tooth Interface

4.11.1 Introduction

The Bluetooth Interface (BTIF) is designed in SOC (BT+GSM) to take the place of the UART interface between BT chip and baseband chip. Similar to the UART design, BTIF is an APB slave and can transmit or receive data by MCU access or through DMA/VFIFO.

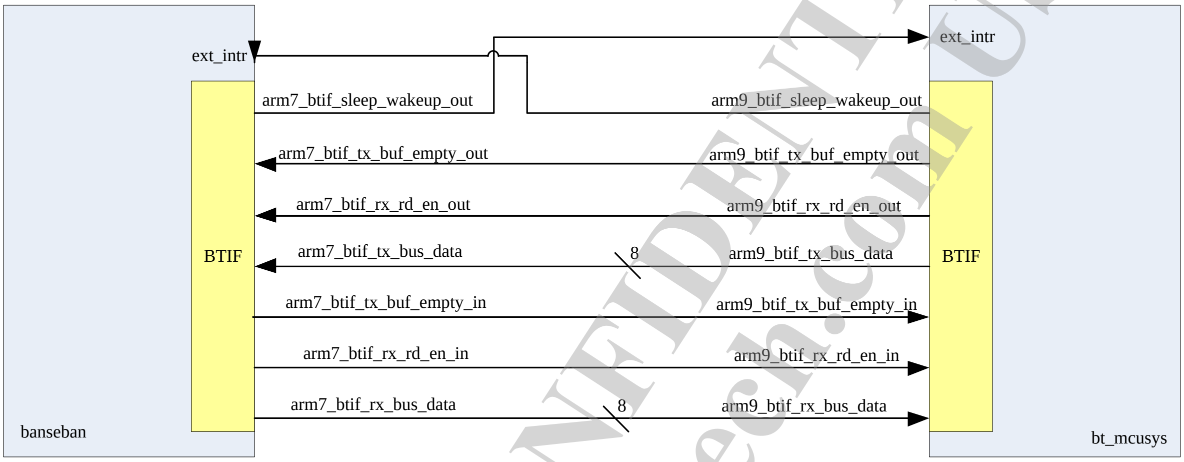


Figure 4-14. Interface Connection between BT and Baseband System

4.11.2 BTIF Block Diagram

Detailed block diagram of BTIF is shown below. Table 4-16 below details the design partitian. The Control Register (CR) of BTIF can be set by MCU or DMA through APB interface. The CR of BTIF in btif_inntr_reg sets up FIFO enable, interrupt, wakeup event and so on. Btif_tx_fifo and btif_rx_fifo are used to temporarily store the transmitted and received data. The BTIF transmission is asynchronous handshake. Therefore, btif_rx_async is required to sync the received data.

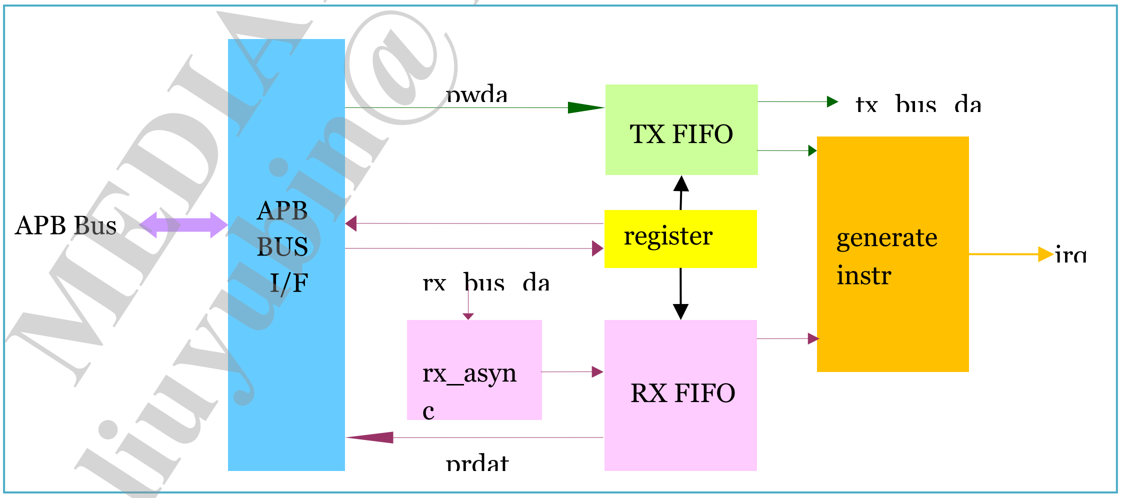


Figure 4-15. BTIF Block Diagram

Table 4-16. BTIF Design Partition

Sub module name (hier1)	Description
btif_intr_reg	APB bus configures UART register and generates interrupt.
btif_rx_async	Synchronously receives control and data signal
btif_rx_fifo	Receives data from baseband
btif_tx_fifo	Transmits data to baseband

4.11.3 Theory of Operations

The table below lists the function of BTIF for tests. There are four test items. The first is to test the transmit data; the second received data and the third the interrupt functionality. Finally, the register is configured to verify the settings of BTIF.

Table 4-17. BTIF Functions

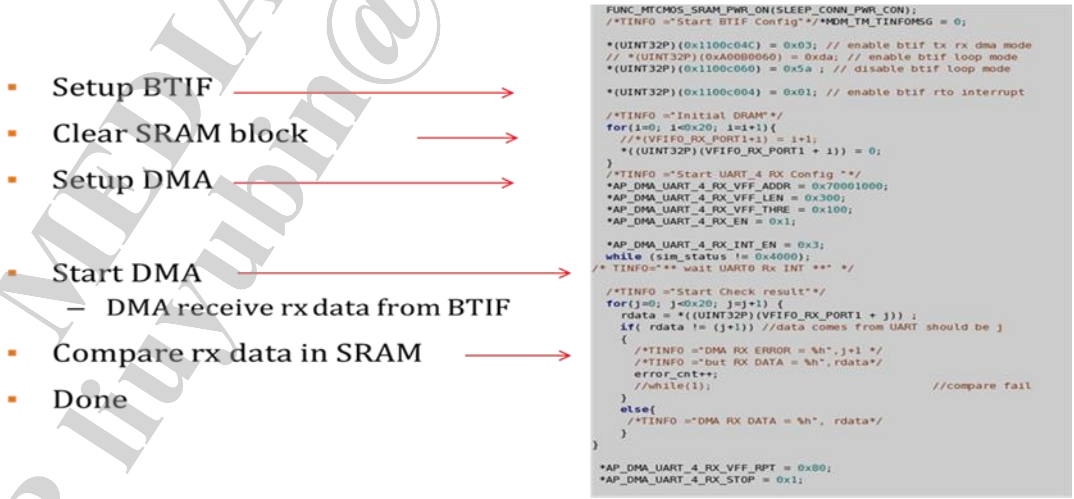
Item	Main function	Description
1	TX FIFO	Transmits data to baseband
2	RX FIFO	Receives data from baseband
3	Interrupt	Generates BTIF interrupt
4	Register	APB bus configures BTIF register.

4.11.4 Register Definitions

For register details refer to chapter 2.10 of "MT8516A Application Processor Register."

4.11.5 Programming Guide

The software programming guide is shown here.



- Setup BTIF
- Clear SRAM block
- Setup DMA
- Start DMA
  - DMA receive rx data from BTIF
- Compare rx data in SRAM
- Done

```
FUNC_MTCMOS_SRAM_PWR_ON(SLEEP_CONN_PWR_CON);
/*TINFO ="Start BTIF Config"*/*MDM_TM_TINFOMSG = 0;

*(UINT32P)(0x1100c04C) = 0x03; // enable btif tx rx dma mode
// *(UINT32P)(0xA00B0060) = 0xda; // enable btif loop mode
*(UINT32P)(0x1100c060) = 0x5a ; // disable btif loop mode

*(UINT32P)(0x1100c004) = 0x01; // enable btif rto interrupt

/*TINFO ="Initial DRAM"*/
for(i=0; i<0x20; i=i+1){
  //*(VFIFO_RX_PORT1+i) = i+1;
  *((UINT32P)(VFIFO_RX_PORT1 + i)) = 0;
}
/*TINFO ="Start UART_4 RX Config "*/
*AP_DMA_UART_4_RX_VFF_ADDR = 0x70001000;
*AP_DMA_UART_4_RX_VFF_LEN = 0x300;
*AP_DMA_UART_4_RX_VFF_THRE = 0x100;
*AP_DMA_UART_4_RX_EN = 0x1;

*AP_DMA_UART_4_RX_INT_EN = 0x3;
while (sim_status != 0x4000);
/* TINFO="** wait UART0 Rx INT **" */

/*TINFO ="Start Check result"*/
for(j=0; j<0x20; j=j+1) {
  rdata = *((UINT32P)(VFIFO_RX_PORT1 + j)) ;
  if( rdata != (j+1)) //data comes from UART should be j
  {
    /*TINFO ="DMA RX ERROR = %h",j+1 */
    /*TINFO ="but RX DATA = %h",rdata*/
    error_cnt++;
    //while(1);                            //compare fail
  }
  else{
   /*TINFO ="DMA RX DATA = %h", rdata*/
  }
}

*AP_DMA_UART_4_RX_VFF_RPT = 0x80;
*AP_DMA_UART_4_RX_STOP = 0x1;
```

In simulating the whole chip, four patterns are used to verify BTIF, as shown in the following table.

Table 4-18. Test Paterns for Whole Chip Simulation

Test list	Test name	Description
conn_mcu_btif.list	conn_btif	Tests DMA + BTIF transmitted and received data
conn_mcu_btif_toggle.list	conn_btif_toggle	Tests toggle coverage of interface
conn_mcu_btif_swrst.list	conn_btif_swrst	Tests interrupt
scpsys_wakeup_event.list	conn_btif_wake_ap	Tests wakeup signal to interrupt other subsys

4.12 Command Cue DMA (CQ_DMA)

CQDMA is a general DMA which transfers data between sram and dram.

There are two general DMA engines in CQDMA.

4.12.1 Block Diagram

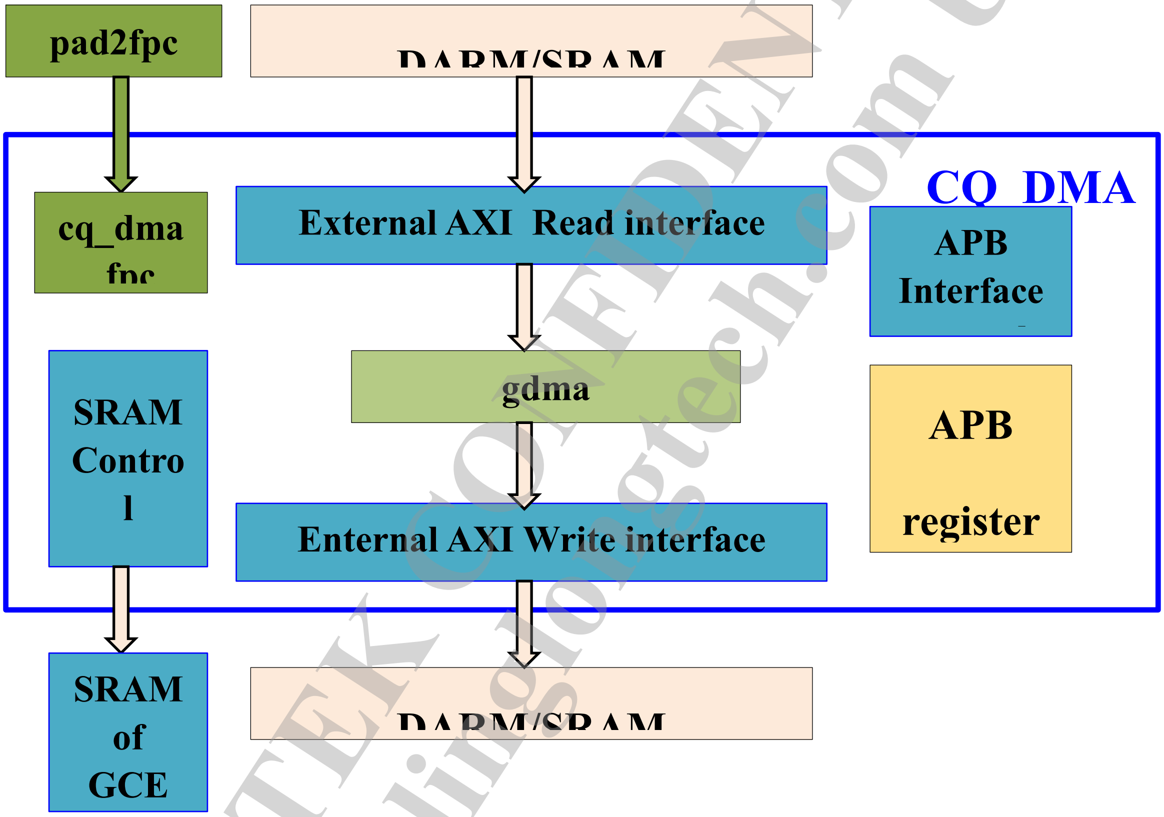


Figure 4-16. CQ_DMA Block Diagram

4.12.2 Register Definition

For register details refer to section 2.11 in “MT8516A Application Processor Registers”.

4.12.3 DMA Function Programming Guide

1. Enable interrupt Enable
   *CQ_DMA_G_DMA_INT_EN =1

2. Config DMA source address and destination address
   *CQ_DMA_G_DMA_SRC_ADDR
   *CQ_DMA_G_DMA_DST_ADDR

3. Config DMA length
   *CQ_DMA_G_DMA_LEN

4. Enable DMA to trigger DMA engine
   *CQ_DMA_G_DMA_EN = 1

5. Wait for Interrupt and Clear
   *CQ_DMA_G_DMA_INT_FLAG = 0

5 Peripherals

5.1 GPIO

5.1.1 Introduction

MT8516A offers 124 general-purpose I/O pins. By setting up the control registers, the MCU software can control the direction, output value and read the input values on these pins. The GPIOs are multiplexed with other functions to reduce the pin count.

5.1.2 Features

The figure below is GPIO block diagram. Each GPIO controls the auxiliary mode by programming the GPIO_MODEx command register.

GPIO_DIR, GPIO_DOUT and GPIO_PULLEN are also programmable by the same method of GPIO_MODE.

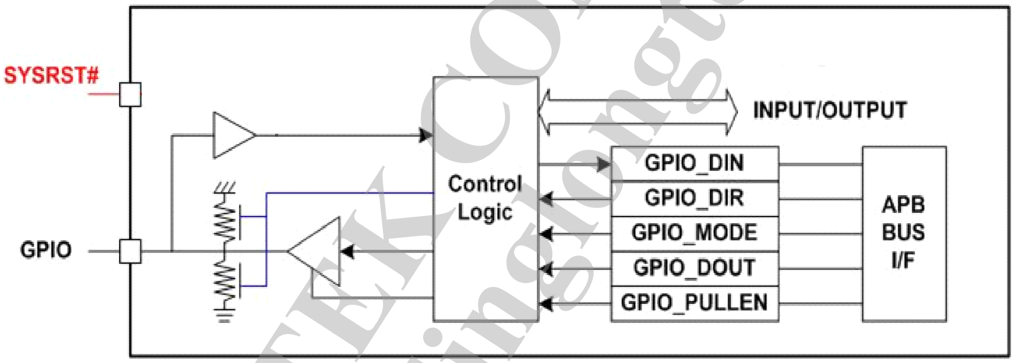


Figure 5-1. GPIO Block Diagram

5.1.3 Block Diagram

The table here shows the mapping table of aux. name, mode number, CU/CD (controllable pull-up and pull-down) and driving capability.

Table 5-1. GPIO Aux Functions

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
EINT0	0	GPIO0	IO	CU,CD	4/8/12/16mA
	1	PWM_B	O	CU,CD	4/8/12/16mA
	2	DPI_CK	O	CU,CD	4/8/12/16mA
	3	I2S2_BCK	O	CU,CD	4/8/12/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	4	EXT_TXD0	O	CU,CD	4/8/12/16mA
	5	-	-	CU,CD	4/8/12/16mA
	6	SQICS	O	CU,CD	4/8/12/16mA
	7	DBG_MON_A[6]	IO	CU,CD	4/8/12/16mA
EINT1	0	GPIO1	IO	CU,CD	4/8/12/16mA
	1	PWM_C	O	CU,CD	4/8/12/16mA
	2	DPI_D12	O	CU,CD	4/8/12/16mA
	3	I2S2_DI	I	CU,CD	4/8/12/16mA
	4	EXT_TXD1	O	CU,CD	4/8/12/16mA
	5	CONN_MCU_TDO	O	CU,CD	4/8/12/16mA
	6	SQISO	IO	CU,CD	4/8/12/16mA
	7	DBG_MON_A[7]	IO	CU,CD	4/8/12/16mA
EINT2	0	GPIO2	IO	CU,CD	4/8/12/16mA
	1	CLKM0	O	CU,CD	4/8/12/16mA
	2	DPI_D13	O	CU,CD	4/8/12/16mA
	3	I2S2_LRCK	O	CU,CD	4/8/12/16mA
	4	EXT_TXD2	O	CU,CD	4/8/12/16mA
	5	CONN_MCU_DBGACK_N	O	CU,CD	4/8/12/16mA
	6	SQISI	IO	CU,CD	4/8/12/16mA
	7	DBG_MON_A[8]	IO	CU,CD	4/8/12/16mA
EINT3	0	GPIO3	IO	CU,CD	4/8/12/16mA
	1	CLKM1	O	CU,CD	4/8/12/16mA
	2	DPI_D14	O	CU,CD	4/8/12/16mA
	3	SPI_MI	I	CU,CD	4/8/12/16mA
	4	EXT_TXD3	O	CU,CD	4/8/12/16mA
	5	CONN_MCU_DBGI_N	I	CU,CD	4/8/12/16mA
	6	SQIWP	IO	CU,CD	4/8/12/16mA
	7	DBG_MON_A[9]	IO	CU,CD	4/8/12/16mA
EINT4	0	GPIO4	IO	CU,CD	4/8/12/16mA
	1	CLKM2	O	CU,CD	4/8/12/16mA
	2	DPI_D15	O	CU,CD	4/8/12/16mA
	3	SPI_MO	O	CU,CD	4/8/12/16mA
	4	EXT_TXC	I	CU,CD	4/8/12/16mA
	5	CONN_MCU_TCK	I	CU,CD	4/8/12/16mA
	6	CONN_MCU_AICE_JCKC	I	CU,CD	4/8/12/16mA
	7	DBG_MON_A[10]	IO	CU,CD	4/8/12/16mA
EINT5	0	GPIO5	IO	CU,CD	4/8/12/16mA
	1	UCTS2	I	CU,CD	4/8/12/16mA
	2	DPI_D16	O	CU,CD	4/8/12/16mA
	3	SPI_CSB	O	CU,CD	4/8/12/16mA
	4	EXT_RXER	I	CU,CD	4/8/12/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	5	CONN_MCU_TDI	I	CU,CD	4/8/12/16mA
	6	CONN_TEST_CK	I	CU,CD	4/8/12/16mA
	7	DBG_MON_A[11]	IO	CU,CD	4/8/12/16mA
EINT6	0	GPIO6	IO	CU,CD	4/8/12/16mA
	1	URTS2	O	CU,CD	4/8/12/16mA
	2	DPI_D17	O	CU,CD	4/8/12/16mA
	3	SPI_CLK	O	CU,CD	4/8/12/16mA
	4	EXT_RXC	I	CU,CD	4/8/12/16mA
	5	CONN_MCU_TRST_B	I	CU,CD	4/8/12/16mA
	6	MM_TEST_CK	I	CU,CD	4/8/12/16mA
	7	DBG_MON_A[12]	IO	CU,CD	4/8/12/16mA
EINT7	0	GPIO7	IO	CU,CD	4/8/12/16mA
	1	SQIRST	IO	CU,CD	4/8/12/16mA
	2	DPI_D6	O	CU,CD	4/8/12/16mA
	3	SDA1_0	IO	CU,CD	4/8/12/16mA
	4	EXT_RXDV	I	CU,CD	4/8/12/16mA
	5	CONN_MCU_TMS	I	CU,CD	4/8/12/16mA
	6	CONN_MCU_AICE_JMSC	IO	CU,CD	4/8/12/16mA
	7	DBG_MON_A[13]	IO	CU,CD	4/8/12/16mA
EINT8	0	GPIO8	IO	CU,CD	4/8/12/16mA
	1	SQICK	O	CU,CD	4/8/12/16mA
	2	CLKM3	O	CU,CD	4/8/12/16mA
	3	SCL1_0	IO	CU,CD	4/8/12/16mA
	4	EXT_RXD0	I	CU,CD	4/8/12/16mA
	5	ANT_SEL0	O	CU,CD	4/8/12/16mA
	6	DPI_D7	O	CU,CD	4/8/12/16mA
	7	DBG_MON_A[14]	IO	CU,CD	4/8/12/16mA
EINT9	0	GPIO9	IO	CU,CD	4/8/12/16mA
	1	CLKM4	O	CU,CD	4/8/12/16mA
	2	SDA2_0	IO	CU,CD	4/8/12/16mA
	3	EXT_FRAME_SYNC	I	CU,CD	4/8/12/16mA
	4	EXT_RXD1	I	CU,CD	4/8/12/16mA
	5	ANT_SEL1	O	CU,CD	4/8/12/16mA
	6	DPI_D8	O	CU,CD	4/8/12/16mA
	7	DBG_MON_A[15]	IO	CU,CD	4/8/12/16mA
EINT10	0	GPIO10	IO	CU,CD	4/8/12/16mA
	1	CLKM5	O	CU,CD	4/8/12/16mA
	2	SCL2_0	IO	CU,CD	4/8/12/16mA
	3	EXT_FRAME_SYNC	I	CU,CD	4/8/12/16mA
	4	EXT_RXD2	I	CU,CD	4/8/12/16mA
	5	ANT_SEL2	O	CU,CD	4/8/12/16mA
	6	DPI_D9	O	CU,CD	4/8/12/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	7	DBG_MON_A[16]	IO	CU,CD	4/8/12/16mA
EINT11	0	GPIO11	IO	CU,CD	4/8/12/16mA
	1	CLKM4	O	CU,CD	4/8/12/16mA
	2	PWM_C	O	CU,CD	4/8/12/16mA
	3	CONN_TEST_CK	I	CU,CD	4/8/12/16mA
	4	ANT_SEL3	O	CU,CD	4/8/12/16mA
	5	DPI_D10	O	CU,CD	4/8/12/16mA
	6	EXT_RXD3	I	CU,CD	4/8/12/16mA
	7	DBG_MON_A[17]	IO	CU,CD	4/8/12/16mA
EINT12	0	GPIO12	IO	CU,CD	4/8/12/16mA
	1	CLKM5	O	CU,CD	4/8/12/16mA
	2	PWM_A	O	CU,CD	4/8/12/16mA
	3	SPDIF_OUT	O	CU,CD	4/8/12/16mA
	4	ANT_SEL4	O	CU,CD	4/8/12/16mA
	5	DPI_D11	O	CU,CD	4/8/12/16mA
	6	EXT_TXEN	O	CU,CD	4/8/12/16mA
	7	DBG_MON_A[18]	IO	CU,CD	4/8/12/16mA
EINT13	0	GPIO13	IO	CU,CD	4/8/12/16mA
	1	-	-	CU,CD	4/8/12/16mA
	2	-	-	CU,CD	4/8/12/16mA
	3	TSF_IN	I	CU,CD	4/8/12/16mA
	4	ANT_SEL5	O	CU,CD	4/8/12/16mA
	5	DPI_D0	O	CU,CD	4/8/12/16mA
	6	SPDIF_IN	I	CU,CD	4/8/12/16mA
	7	DBG_MON_A[19]	IO	CU,CD	4/8/12/16mA
EINT14	0	GPIO14	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_DO1	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	TDM_RX_MCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	4	ANT_SEL1	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	CONN_MCU_DBGACK_N	O	CU,CD	2/4/6/8/10/12/14/16mA
	6	NCLE	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[8]	IO	CU,CD	2/4/6/8/10/12/14/16mA
EINT15	0	GPIO15	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_LRCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	TDM_RX_BCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	4	ANT_SEL2	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	CONN_MCU_DBGI_N	I	CU,CD	2/4/6/8/10/12/14/16mA
	6	NCEB1	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[9]	IO	CU,CD	2/4/6/8/10/12/14/16mA
EINT16	0	GPIO16	IO	CU,CD	2/4/6/8/10/12/14/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	1	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_BCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	TDM_RX_LRCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	4	ANT_SEL3	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	CONN_MCU_TRST_B	I	CU,CD	2/4/6/8/10/12/14/16mA
	6	NCEB0	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[10]	IO	CU,CD	2/4/6/8/10/12/14/16mA
EINT17	0	GPIO17	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_MCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	TDM_RX_DI	I	CU,CD	2/4/6/8/10/12/14/16mA
	4	IDDIG	I	CU,CD	2/4/6/8/10/12/14/16mA
	5	ANT_SEL4	O	CU,CD	2/4/6/8/10/12/14/16mA
	6	NREB	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[11]	IO	CU,CD	2/4/6/8/10/12/14/16mA
EINT18	0	GPIO18	IO	CU,CD	2/4/6/8mA
	1	-	-	CU,CD	2/4/6/8mA
	2	USB_DRVVBUS	O	CU,CD	2/4/6/8mA
	3	I2S3_LRCK	O	CU,CD	2/4/6/8mA
	4	CLKM1	O	CU,CD	2/4/6/8mA
	5	ANT_SEL3	O	CU,CD	2/4/6/8mA
	6	I2S2_BCK	O	CU,CD	2/4/6/8mA
	7	DBG_MON_A[20]	IO	CU,CD	2/4/6/8mA
EINT19	0	GPIO19	IO	CU,CD	2/4/6/8mA
	1	UCTS1	I	CU,CD	2/4/6/8mA
	2	IDDIG	I	CU,CD	2/4/6/8mA
	3	I2S3_BCK	O	CU,CD	2/4/6/8mA
	4	CLKM2	O	CU,CD	2/4/6/8mA
	5	ANT_SEL4	O	CU,CD	2/4/6/8mA
	6	I2S2_DI	I	CU,CD	2/4/6/8mA
	7	DBG_MON_A[21]	IO	CU,CD	2/4/6/8mA
EINT20	0	GPIO20	IO	CU,CD	4/8/12/16mA
	1	URTS1	O	CU,CD	4/8/12/16mA
	2	-	-	CU,CD	4/8/12/16mA
	3	I2S3_DO	O	CU,CD	4/8/12/16mA
	4	CLKM3	O	CU,CD	4/8/12/16mA
	5	ANT_SEL5	O	CU,CD	4/8/12/16mA
	6	I2S2_LRCK	O	CU,CD	4/8/12/16mA
	7	DBG_MON_A[22]	IO	CU,CD	4/8/12/16mA
EINT21	0	GPIO21	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	NRNB	I	CU,CD	2/4/6/8/10/12/14/16mA
	2	ANT_SEL0	O	CU,CD	2/4/6/8/10/12/14/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	3	I2S_8CH_DO4	O	CU,CD	2/4/6/8/10/12/14/16mA
	4	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[31]	IO	CU,CD	2/4/6/8/10/12/14/16mA
EINT22	0	GPIO22	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_DO2	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	TSF_IN	I	CU,CD	2/4/6/8/10/12/14/16mA
	4	USB_DRVVBUS	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	SPDIF_OUT	O	CU,CD	2/4/6/8/10/12/14/16mA
	6	NRE_C	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[12]	IO	CU,CD	2/4/6/8/10/12/14/16mA
EINT23	0	GPIO23	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_DO3	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	CLKM0	O	CU,CD	2/4/6/8/10/12/14/16mA
	4	IR	I	CU,CD	2/4/6/8/10/12/14/16mA
	5	SPDIF_IN	I	CU,CD	2/4/6/8/10/12/14/16mA
	6	NDQS_C	IO	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[13]	IO	CU,CD	2/4/6/8/10/12/14/16mA
EINT24	0	GPIO24	IO	CU,CD	4/8/12/16mA
	1	DPI_D20	O	CU,CD	4/8/12/16mA
	2	DPI_DE	O	CU,CD	4/8/12/16mA
	3	ANT_SEL1	O	CU,CD	4/8/12/16mA
	4	UCTS2	I	CU,CD	4/8/12/16mA
	5	PWM_A	O	CU,CD	4/8/12/16mA
	6	I2S0_MCK	O	CU,CD	4/8/12/16mA
	7	DBG_MON_A[0]	IO	CU,CD	4/8/12/16mA
EINT25	0	GPIO25	IO	CU,CD	4/8/12/16mA
	1	DPI_D19	O	CU,CD	4/8/12/16mA
	2	DPI_VSYNC	O	CU,CD	4/8/12/16mA
	3	ANT_SEL0	O	CU,CD	4/8/12/16mA
	4	URTS2	O	CU,CD	4/8/12/16mA
	5	PWM_B	O	CU,CD	4/8/12/16mA
	6	I2S_8CH_MCK	O	CU,CD	4/8/12/16mA
	7	DBG_MON_A[1]	IO	CU,CD	4/8/12/16mA
PWRAP_SPI0_MI	0	GPIO26	IO	CU,CD	2/4/6/8mA
	1	PWRAP_SPI0_MO	IO	CU,CD	2/4/6/8mA
	2	PWRAP_SPI0_MI	IO	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
PWRAP_SPI0_MO	0	GPIO27	IO	CU,CD	2/4/6/8mA
	1	PWRAP_SPI0_MI	IO	CU,CD	2/4/6/8mA
	2	PWRAP_SPI0_MO	IO	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
PWRAP_INT	0	GPIO28	IO	CU,CD	2/4/6/8mA
	1	I2S0_MCK	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	I2S_8CH_MCK	O	CU,CD	2/4/6/8mA
	5	I2S2_MCK	O	CU,CD	2/4/6/8mA
	6	I2S3_MCK	O	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
PWRAP_SPI0_CK	0	GPIO29	IO	CU,CD	2/4/6/8mA
	1	PWRAP_SPI0_CK	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
PWRAP_SPI0_CSN	0	GPIO30	IO	CU,CD	2/4/6/8mA
	1	PWRAP_SPI0_CSN	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
RTC32K_CK	0	GPIO31	IO	CU,CD	2/4/6/8mA
	1	RTC32K_CK	I	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	7	-	-	CU,CD	2/4/6/8mA
WATCHDOG	0	GPIO32	IO	CU,CD	2/4/6/8mA
	1	WATCHDOG	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
SRCLKENA	0	GPIO33	IO	CU,CD	2/4/6/8mA
	1	SRCLKENA0	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
URXD2	0	GPIO34	IO	CU,CD	4/8/12/16mA
	1	URXD2	I	CU,CD	4/8/12/16mA
	2	DPI_D5	O	CU,CD	4/8/12/16mA
	3	UTXD2	O	CU,CD	4/8/12/16mA
	4	DBG_SCL	IO	CU,CD	4/8/12/16mA
	5	-	-	CU,CD	4/8/12/16mA
	6	I2S2_MCK	O	CU,CD	4/8/12/16mA
	7	DBG_MON_B[0]	IO	CU,CD	4/8/12/16mA
UTXD2	0	GPIO35	IO	CU,CD	4/8/12/16mA
	1	UTXD2	O	CU,CD	4/8/12/16mA
	2	DPI_HSYNC	O	CU,CD	4/8/12/16mA
	3	URXD2	I	CU,CD	4/8/12/16mA
	4	DBG_SDA	IO	CU,CD	4/8/12/16mA
	5	DPI_D18	O	CU,CD	4/8/12/16mA
	6	I2S3_MCK	O	CU,CD	4/8/12/16mA
	7	DBG_MON_B[1]	IO	CU,CD	4/8/12/16mA
MRG_CLK	0	GPIO36	IO	CU,CD	4/8/12/16mA
	1	MRG_CLK	O	CU,CD	4/8/12/16mA
	2	DPI_D4	O	CU,CD	4/8/12/16mA
	3	I2S0_BCK	IO	CU,CD	4/8/12/16mA
	4	I2S3_BCK	O	CU,CD	4/8/12/16mA
	5	PCM0_CLK	O	CU,CD	4/8/12/16mA
	6	IR	I	CU,CD	4/8/12/16mA
	7	DBG_MON_A[2]	IO	CU,CD	4/8/12/16mA
MRG_SYNC	0	GPIO37	IO	CU,CD	4/8/12/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	1	MRG_SYNC	O	CU,CD	4/8/12/16mA
	2	DPI_D3	O	CU,CD	4/8/12/16mA
	3	I2S0_LRCK	IO	CU,CD	4/8/12/16mA
	4	I2S3_LRCK	O	CU,CD	4/8/12/16mA
	5	PCM0_SYNC	O	CU,CD	4/8/12/16mA
	6	EXT_COL	IO	CU,CD	4/8/12/16mA
	7	DBG_MON_A[3]	IO	CU,CD	4/8/12/16mA
MRG_DI	0	GPIO38	IO	CU,CD	4/8/12/16mA
	1	MRG_DI	I	CU,CD	4/8/12/16mA
	2	DPI_D1	O	CU,CD	4/8/12/16mA
	3	I2S0_DI	I	CU,CD	4/8/12/16mA
	4	I2S3_DO	O	CU,CD	4/8/12/16mA
	5	PCM0_DI	I	CU,CD	4/8/12/16mA
	6	EXT_MDIO	IO	CU,CD	4/8/12/16mA
	7	DBG_MON_A[4]	IO	CU,CD	4/8/12/16mA
MRG_DO	0	GPIO39	IO	CU,CD	4/8/12/16mA
	1	MRG_DO	O	CU,CD	4/8/12/16mA
	2	DPI_D2	O	CU,CD	4/8/12/16mA
	3	I2S0_MCK	O	CU,CD	4/8/12/16mA
	4	I2S3_MCK	O	CU,CD	4/8/12/16mA
	5	PCM0_DO	O	CU,CD	4/8/12/16mA
	6	EXT_MDC	O	CU,CD	4/8/12/16mA
	7	DBG_MON_A[5]	IO	CU,CD	4/8/12/16mA
KPROW0	0	GPIO40	IO	CU,CD	2/4/6/8mA
	1	KPROW0	IO	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	IMG_TEST_CK	I	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[4]	IO	CU,CD	2/4/6/8mA
KPROW1	0	GPIO41	IO	CU,CD	2/4/6/8mA
	1	KPROW1	IO	CU,CD	2/4/6/8mA
	2	IDDIG	I	CU,CD	2/4/6/8mA
	3	EXT_FRAME_SYNC	I	CU,CD	2/4/6/8mA
	4	MFG_TEST_CK	I	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[5]	IO	CU,CD	2/4/6/8mA
KPCOL0	0	GPIO42	IO	CU,CD	2/4/6/8mA
	1	KPCOL0	IO	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[6]	IO	CU,CD	2/4/6/8mA
KPCOL1	0	GPIO43	IO	CU,CD	2/4/6/8mA
	1	KPCOL1	IO	CU,CD	2/4/6/8mA
	2	USB_DRVVBUS	O	CU,CD	2/4/6/8mA
	3	EXT_FRAME_SYNC	I	CU,CD	2/4/6/8mA
	4	TSF_IN	I	CU,CD	2/4/6/8mA
	5	DFD_NTRST_XI	I	CU,CD	2/4/6/8mA
	6	UDI_NTRST_XI	I	CU,CD	2/4/6/8mA
	7	DBG_MON_B[7]	IO	CU,CD	2/4/6/8mA
JTMS	0	GPIO44	IO	CU,CD	2/4/6/8mA
	1	JTMS	IO	CU,CD	2/4/6/8mA
	2	CONN_MCU_TMS	I	CU,CD	2/4/6/8mA
	3	CONN_MCU_AICE_JMSC	IO	CU,CD	2/4/6/8mA
	4	GPUDFD_TMS_XI	I	CU,CD	2/4/6/8mA
	5	DFD_TMS_XI	I	CU,CD	2/4/6/8mA
	6	UDI_TMS_XI	I	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
JTCK	0	GPIO45	IO	CU,CD	2/4/6/8mA
	1	JTCK	I	CU,CD	2/4/6/8mA
	2	CONN_MCU_TCK	I	CU,CD	2/4/6/8mA
	3	CONN_MCU_AICE_JCKC	I	CU,CD	2/4/6/8mA
	4	GPUDFD_TCK_XI	I	CU,CD	2/4/6/8mA
	5	DFD_TCK_XI	I	CU,CD	2/4/6/8mA
	6	UDI_TCK_XI	I	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
JTDI	0	GPIO46	IO	CU,CD	2/4/6/8mA
	1	JTDI	I	CU,CD	2/4/6/8mA
	2	CONN_MCU_TDI	I	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	GPUDFD_TDI_XI	I	CU,CD	2/4/6/8mA
	5	DFD_TDI_XI	I	CU,CD	2/4/6/8mA
	6	UDI_TDI_XI	I	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
JTDO	0	GPIO47	IO	CU,CD	2/4/6/8mA
	1	JTDO	O	CU,CD	2/4/6/8mA
	2	CONN_MCU_TDO	O	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	4	GPUDFD_TDO	O	CU,CD	2/4/6/8mA
	5	DFD_TDO	O	CU,CD	2/4/6/8mA
	6	UDI_TDO	O	CU,CD	2/4/6/8mA
	7	-	-	CU,CD	2/4/6/8mA
SPI_CS	0	GPIO48	IO	CU,CD	2/4/6/8mA
	1	SPI_CSB	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	I2S0_DI	I	CU,CD	2/4/6/8mA
	4	I2S2_BCK	O	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_A[23]	IO	CU,CD	2/4/6/8mA
SPI_CK	0	GPIO49	IO	CU,CD	2/4/6/8mA
	1	SPI_CLK	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	I2S0_LRCK	IO	CU,CD	2/4/6/8mA
	4	I2S2_DI	I	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_A[24]	IO	CU,CD	2/4/6/8mA
SPI_MI	0	GPIO50	IO	CU,CD	2/4/6/8mA
	1	SPI_MI	I	CU,CD	2/4/6/8mA
	2	SPI_MO	O	CU,CD	2/4/6/8mA
	3	I2S0_BCK	IO	CU,CD	2/4/6/8mA
	4	I2S2_LRCK	O	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_A[25]	IO	CU,CD	2/4/6/8mA
SPI_MO	0	GPIO51	IO	CU,CD	2/4/6/8mA
	1	SPI_MO	O	CU,CD	2/4/6/8mA
	2	SPI_MI	I	CU,CD	2/4/6/8mA
	3	I2S0_MCK	O	CU,CD	2/4/6/8mA
	4	I2S2_MCK	O	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_A[26]	IO	CU,CD	2/4/6/8mA
SDA1	0	GPIO52	IO	CD	-
	1	SDA1_0	IO	CD	-
	2	-	-	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	6	-	-	CD	-
	7	-	-	CD	-
SCL1	0	GPIO53	IO	CD	-
	1	SCL1_0	IO	CD	-
	2	-	-	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-
DISP_PWM	0	GPIO54	IO	CU,CD	2/4/6/8mA
	1	DISP_PWM	O	CU,CD	2/4/6/8mA
	2	PWM_B	O	CU,CD	2/4/6/8mA
	3	-	-	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[2]	IO	CU,CD	2/4/6/8mA
I2S_DATA_IN	0	GPIO55	IO	CU,CD	2/4/6/8mA
	1	I2S0_DI	I	CU,CD	2/4/6/8mA
	2	UCTS0	I	CU,CD	2/4/6/8mA
	3	I2S3_DO	O	CU,CD	2/4/6/8mA
	4	I2S_8CH_DO1	O	CU,CD	2/4/6/8mA
	5	PWM_A	O	CU,CD	2/4/6/8mA
	6	I2S2_BCK	O	CU,CD	2/4/6/8mA
	7	DBG_MON_A[28]	IO	CU,CD	2/4/6/8mA
I2S_LRCK	0	GPIO56	IO	CU,CD	2/4/6/8mA
	1	I2S0_LRCK	IO	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	I2S3_LRCK	O	CU,CD	2/4/6/8mA
	4	I2S_8CH_LRCK	O	CU,CD	2/4/6/8mA
	5	PWM_B	O	CU,CD	2/4/6/8mA
	6	I2S2_DI	I	CU,CD	2/4/6/8mA
	7	DBG_MON_A[29]	IO	CU,CD	2/4/6/8mA
I2S_BCK	0	GPIO57	IO	CU,CD	2/4/6/8mA
	1	I2S0_BCK	IO	CU,CD	2/4/6/8mA
	2	URTS0	O	CU,CD	2/4/6/8mA
	3	I2S3_BCK	O	CU,CD	2/4/6/8mA
	4	I2S_8CH_BCK	O	CU,CD	2/4/6/8mA
	5	PWM_C	O	CU,CD	2/4/6/8mA
	6	I2S2_LRCK	O	CU,CD	2/4/6/8mA
	7	DBG_MON_A[30]	IO	CU,CD	2/4/6/8mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
SDA0	0	GPIO58	IO	CD	-
	1	SDA0_0	IO	CD	-
	2	-	-	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-
SCL0	0	GPIO59	IO	CD	-
	1	SCL0_0	IO	CD	-
	2	-	-	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-
SDA2	0	GPIO60	IO	CD	-
	1	SDA2_0	IO	CD	-
	2	PWM_B	O	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-
SCL2	0	GPIO61	IO	CD	-
	1	SCL2_0	IO	CD	-
	2	PWM_C	O	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-
URXD0	0	GPIO62	IO	CU,CD	4/8/12/16mA
	1	URXD0	I	CU,CD	4/8/12/16mA
	2	UTXD0	O	CU,CD	4/8/12/16mA
	3	-	-	CU,CD	4/8/12/16mA
	4	-	-	CU,CD	4/8/12/16mA
	5	-	-	CU,CD	4/8/12/16mA
	6	-	-	CU,CD	4/8/12/16mA
	7	-	-	CU,CD	4/8/12/16mA
UTXD0	0	GPIO63	IO	CU,CD	4/8/12/16mA
	1	UTXD0	O	CU,CD	4/8/12/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	2	URXD0	I	CU,CD	4/8/12/16mA
	3	-	-	CU,CD	4/8/12/16mA
	4	-	-	CU,CD	4/8/12/16mA
	5	-	-	CU,CD	4/8/12/16mA
	6	-	-	CU,CD	4/8/12/16mA
	7	-	-	CU,CD	4/8/12/16mA
URXD1	0	GPIO64	IO	CU,CD	4/8/12/16mA
	1	URXD1	I	CU,CD	4/8/12/16mA
	2	UTXD1	O	CU,CD	4/8/12/16mA
	3	-	-	CU,CD	4/8/12/16mA
	4	-	-	CU,CD	4/8/12/16mA
	5	-	-	CU,CD	4/8/12/16mA
	6	-	-	CU,CD	4/8/12/16mA
	7	DBG_MON_A[27]	IO	CU,CD	4/8/12/16mA
UTXD1	0	GPIO65	IO	CU,CD	4/8/12/16mA
	1	UTXD1	O	CU,CD	4/8/12/16mA
	2	URXD1	I	CU,CD	4/8/12/16mA
	3	-	-	CU,CD	4/8/12/16mA
	4	-	-	CU,CD	4/8/12/16mA
	5	-	-	CU,CD	4/8/12/16mA
	6	-	-	CU,CD	4/8/12/16mA
	7	DBG_MON_A[31]	IO	CU,CD	4/8/12/16mA
LCM_RST	0	GPIO66	IO	CU,CD	2/4/6/8mA
	1	LCM_RST	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	I2S0_MCK	O	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[3]	IO	CU,CD	2/4/6/8mA
DSI_TE	0	GPIO67	IO	CU,CD	2/4/6/8mA
	1	DSI_TE	I	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	I2S_8CH_MCK	O	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	-	-	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[14]	IO	CU,CD	2/4/6/8mA
MSDC2_CMD	0	GPIO68	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC2_CMD	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_DO4	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	SDA1_0	IO	CU,CD	2/4/6/8/10/12/14/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	4	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	5	USB_SDA	IO	CU,CD	2/4/6/8/10/12/14/16mA
	6	I2S3_BCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[15]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC2_CLK	0	GPIO69	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC2_CLK	O	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_DO3	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	SCL1_0	IO	CU,CD	2/4/6/8/10/12/14/16mA
	4	DPI_D21	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	USB_SCL	IO	CU,CD	2/4/6/8/10/12/14/16mA
	6	I2S3_LRCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[16]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC2_DAT0	0	GPIO70	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC2_DAT0	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_DO2	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	DPI_D22	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	UTXD0	O	CU,CD	2/4/6/8/10/12/14/16mA
	6	I2S3_DO	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[17]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC2_DAT1	0	GPIO71	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC2_DAT1	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_DO1	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	PWM_A	O	CU,CD	2/4/6/8/10/12/14/16mA
	4	I2S3_MCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	URXD0	I	CU,CD	2/4/6/8/10/12/14/16mA
	6	PWM_B	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[18]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC2_DAT2	0	GPIO72	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC2_DAT2	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_LRCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	SDA2_0	IO	CU,CD	2/4/6/8/10/12/14/16mA
	4	DPI_D23	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	UTXD1	O	CU,CD	2/4/6/8/10/12/14/16mA
	6	PWM_C	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[19]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC2_DAT3	0	GPIO73	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC2_DAT3	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	I2S_8CH_BCK	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	SCL2_0	IO	CU,CD	2/4/6/8/10/12/14/16mA
	4	EXT_FRAME_SYNC	I	CU,CD	2/4/6/8/10/12/14/16mA
	5	URXD1	I	CU,CD	2/4/6/8/10/12/14/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	6	PWM_A	O	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[20]	IO	CU,CD	2/4/6/8/10/12/14/16mA
TDN3	0	GPI74	I	-	-
	1	TDN3	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
TDP3	0	GPI75	I	-	-
	1	TDP3	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
TDN2	0	GPI76	I	-	-
	1	TDN2	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
TDP2	0	GPI77	I	-	-
	1	TDP2	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
TCN	0	GPI78	I	-	-
	1	TCN	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
TCP	0	GPI79	I	-	-
	1	TCP	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
TDN1	0	GPI80	I	-	-
	1	TDN1	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
TDP1	0	GPI81	I	-	-
	1	TDP1	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
TDN0	0	GPI82	I	-	-
	1	TDN0	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
TDP0	0	GPI83	I	-	-
	1	TDP0	O	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDN0	0	GPI84	I	-	-
	1	RDN0	I	-	-

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDP0	0	GPI85	I	-	-
	1	RDP0	I	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDN1	0	GPI86	I	-	-
	1	RDN1	I	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDP1	0	GPI87	I	-	-
	1	RDP1	I	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RCN	0	GPI88	I	-	-
	1	RCN	I	-	-
	2	-	-	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RCP	0	GPI89	I	-	-
	1	RCP	I	-	-
	2	-	-	-	-
	3	-	-	-	-

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDN2	0	GPI90	I	-	-
	1	RDN2	I	-	-
	2	CMDAT8	I	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDP2	0	GPI91	I	-	-
	1	RDP2	I	-	-
	2	CMDAT9	I	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDN3	0	GPI92	I	-	-
	1	RDN3	I	-	-
	2	CMDAT4	I	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDP3	0	GPI93	I	-	-
	1	RDP3	I	-	-
	2	CMDAT5	I	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RCN_A	0	GPI94	I	-	-
	1	RCN_A	I	-	-
	2	CMDAT6	I	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	6	-	-	-	-
	7	-	-	-	-
RCP_A	0	GPI95	I	-	-
	1	RCP_A	I	-	-
	2	CMDAT7	I	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDN1_A	0	GPI96	I	-	-
	1	RDN1_A	I	-	-
	2	CMDAT2	I	-	-
	3	CMCSD2	I	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDP1_A	0	GPI97	I	-	-
	1	RDP1_A	I	-	-
	2	CMDAT3	I	-	-
	3	CMCSD3	I	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDN0_A	0	GPI98	I	-	-
	1	RDN0_A	I	-	-
	2	CMHSYNC	I	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-
RDP0_A	0	GPI99	I	-	-
	1	RDP0_A	I	-	-
	2	CMVSYNC	I	-	-
	3	-	-	-	-
	4	-	-	-	-
	5	-	-	-	-
	6	-	-	-	-
	7	-	-	-	-

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
CMDAT0	0	GPIO100	IO	CU,CD	2/4/6/8mA
	1	CMDAT0	I	CU,CD	2/4/6/8mA
	2	CMCSD0	I	CU,CD	2/4/6/8mA
	3	ANT_SEL2	O	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	TDM_RX_MCK	O	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[21]	IO	CU,CD	2/4/6/8mA
CMDAT1	0	GPIO101	IO	CU,CD	2/4/6/8mA
	1	CMDAT1	I	CU,CD	2/4/6/8mA
	2	CMCSD1	I	CU,CD	2/4/6/8mA
	3	ANT_SEL3	O	CU,CD	2/4/6/8mA
	4	CMFLASH	O	CU,CD	2/4/6/8mA
	5	TDM_RX_BCK	O	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[22]	IO	CU,CD	2/4/6/8mA
CMMCLK	0	GPIO102	IO	CU,CD	2/4/6/8mA
	1	CMMCLK	O	CU,CD	2/4/6/8mA
	2	-	-	CU,CD	2/4/6/8mA
	3	ANT_SEL4	O	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	TDM_RX_LRCK	O	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[23]	IO	CU,CD	2/4/6/8mA
CMPCLK	0	GPIO103	IO	CU,CD	2/4/6/8mA
	1	CMPCLK	I	CU,CD	2/4/6/8mA
	2	CMCSK	I	CU,CD	2/4/6/8mA
	3	ANT_SEL5	O	CU,CD	2/4/6/8mA
	4	-	-	CU,CD	2/4/6/8mA
	5	TDM_RX_DI	I	CU,CD	2/4/6/8mA
	6	-	-	CU,CD	2/4/6/8mA
	7	DBG_MON_B[24]	IO	CU,CD	2/4/6/8mA
MSDC1_CMD	0	GPIO104	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC1_CMD	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	SQICS	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[25]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC1_CLK	0	GPIO105	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC1_CLK	O	CU,CD	2/4/6/8/10/12/14/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	2	UDI_NTRST_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	3	DFD_NTRST_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	4	SQISO	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	GPUEJ_NTRST_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[26]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC1_DAT0	0	GPIO106	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC1_DAT0	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	UDI_TMS_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	3	DFD_TMS_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	4	SQISI	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	GPUEJ_TMS_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[27]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC1_DAT1	0	GPIO107	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC1_DAT1	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	UDI_TCK_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	3	DFD_TCK_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	4	SQIWP	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	GPUEJ_TCK_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[28]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC1_DAT2	0	GPIO108	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC1_DAT2	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	UDI_TDI_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	3	DFD_TDI_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	4	SQIRST	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	GPUEJ_TDI_XI	I	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[29]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC1_DAT3	0	GPIO109	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC1_DAT3	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	UDI_TDO	O	CU,CD	2/4/6/8/10/12/14/16mA
	3	DFD_TDO	O	CU,CD	2/4/6/8/10/12/14/16mA
	4	SQICK	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	GPUEJ_TDO	IO	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	DBG_MON_B[30]	IO	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_DAT7	0	GPIO110	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_DAT7	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	4	NLD7	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_DAT6	0	GPIO111	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_DAT6	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NLD6	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_DAT5	0	GPIO112	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_DAT5	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NLD4	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_DAT4	0	GPIO113	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_DAT4	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NLD3	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_RSTB	0	GPIO114	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_RSTB	O	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NLD0	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_CMD	0	GPIO115	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_CMD	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NALE	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_CLK	0	GPIO116	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_CLK	O	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NWEB	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_DAT3	0	GPIO117	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_DAT3	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NLD1	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_DAT2	0	GPIO118	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_DAT2	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NLD5	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_DAT1	0	GPIO119	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_DAT1	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	NLD8	IO	CU,CD	2/4/6/8/10/12/14/16mA
	5	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA
MSDC0_DAT0	0	GPIO120	IO	CU,CD	2/4/6/8/10/12/14/16mA
	1	MSDC0_DAT0	IO	CU,CD	2/4/6/8/10/12/14/16mA
	2	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	3	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	4	WATCHDOG	O	CU,CD	2/4/6/8/10/12/14/16mA
	5	NLD2	IO	CU,CD	2/4/6/8/10/12/14/16mA
	6	-	-	CU,CD	2/4/6/8/10/12/14/16mA
	7	-	-	CU,CD	2/4/6/8/10/12/14/16mA

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving
CEC	0	GPIO121	IO	CD	-
	1	CEC	IO	CD	-
	2	-	-	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-
HTPLG	0	GPIO122	IO	CD	-
	1	HTPLG	I	CD	-
	2	-	-	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-
HDMISCK	0	GPIO123	IO	CD	-
	1	HDMISCK	IO	CD	-
	2	-	-	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-
HDMISD	0	GPIO124	IO	CD	-
	1	HDMISD	IO	CD	-
	2	-	-	CD	-
	3	-	-	CD	-
	4	-	-	CD	-
	5	-	-	CD	-
	6	-	-	CD	-
	7	-	-	CD	-

The reset status of MT8516A pins is as shown below.

Table 5-2. GPIO Reset Status

Name	Reset					
	State	Aux	PU/PD	Driving	IES	SMT
PAD_EINT0	I	0	PD	4mA	1	0
PAD_EINT1	I	0	PD	4mA	1	0
PAD_EINT2	I	0	PD	4mA	1	0
PAD_EINT3	I	0	PD	4mA	1	0
PAD_EINT4	I	0	PD	4mA	1	0

Name	Reset					
	State	Aux	PU/PD	Driving	IES	SMT
PAD_EINT5	I	0	PD	4mA	1	0
PAD_EINT6	I	0	PD	4mA	1	0
PAD_EINT7	I	0	PD	4mA	1	0
PAD_EINT8	I	0	PD	4mA	1	0
PAD_EINT9	I	0	PD	4mA	1	0
PAD_EINT10	I	0	PD	4mA	1	0
PAD_EINT11	I	0	PD	4mA	1	0
PAD_EINT12	I	0	PD	4mA	1	0
PAD_EINT13	I	0	PD	4mA	1	0
PAD_EINT14	OL	6	PD	2mA	1	0
PAD_EINT15	OH	6	PU	2mA	1	0
PAD_EINT16	OH	6	PU	2mA	1	0
PAD_EINT17	OH	6	PU	2mA	1	0
PAD_EINT18	I	0	PD	2mA	1	0
PAD_EINT19	I	0	PD	2mA	1	0
PAD_EINT20	I	0	PD	2mA	1	0
PAD_EINT21	I	1	PU	2mA	1	0
PAD_EINT22	OL	6	PD	2mA	1	0
PAD_EINT23	OH	6	PU	2mA	1	0
PAD_EINT24	I	0	PD	4mA	1	0
PAD_EINT25	I	0	PD	4mA	1	0
PAD_PWRAP_SPI0_MI	I	1	PD	2mA	1	0
PAD_PWRAP_SPI0_MO	I	1	PD	2mA	1	0
PAD_PWRAP_INT	I	0	PD	2mA	1	0
PAD_PWRAP_SPI0_CK	OL	1	PD	2mA	1	0
PAD_PWRAP_SPI0_CSN	OH	1	PU	2mA	1	0
PAD_RTC32K_CK	I	1	PD	2mA	1	0
PAD_WATCHDOG	OL	1	PD	2mA	1	0
PAD_SRCLKENA	OH	1	PU	2mA	1	0
PAD_URXD2	I	0	PD	4mA	1	0
PAD_UTXD2	I	0	PD	4mA	1	0
PAD_MRG_CLK	I	0	PD	4mA	1	0
PAD_MRG_SYNC	I	0	PD	4mA	1	0
PAD_MRG_DI	I	0	PD	4mA	1	0
PAD_MRG_DO	I	0	PD	4mA	1	0
PAD_KPROW0		1	PD	2mA	1	0
PAD_KPROW1	I	0	PD	2mA	1	0
PAD_KPCOL0	I	0	PU	2mA	1	0
PAD_KPCOL1	I	0	PD	2mA	1	0
PAD_JTMS	I	1	PU	2mA	1	0
PAD_JTCK	I	1	PU	2mA	1	0

Name	Reset					
	State	Aux	PU/PD	Driving	IES	SMT
PAD_JTDI	I	1	PU	2mA	1	0
PAD_JTDO	OL	1	PD	2mA	1	0
PAD_SPI_CS	I	0	PD	2mA	1	0
PAD_SPI_CK	I	0	PD	2mA	1	0
PAD_SPI_MI	I	0	PD	2mA	1	0
PAD_SPI_MO	I	0	PD	2mA	1	0
PAD_SDA1	I	1	NO-PULL	2mA	1	0
PAD_SCL1	I	1	NO-PULL	2mA	1	0
PAD_DISP_PWM	I	0	PD	2mA	1	0
PAD_I2S_DATA_IN	I	0	PD	2mA	1	0
PAD_I2S_LRCK	I	0	PD	2mA	1	0
PAD_I2S_BCK	I	0	PD	2mA	1	0
PAD_SDA0	I	1	NO-PULL	2mA	1	0
PAD_SCL0	I	1	NO-PULL	2mA	1	0
PAD_SDA2	I	1	NO-PULL	2mA	1	0
PAD_SCL2	I	1	NO-PULL	2mA	1	0
PAD_URXD0	I	1	PU	2mA	1	0
PAD_UTXD0	OH	1	PU	2mA	1	0
PAD_URXD1	I	0	PD	2mA	1	0
PAD_UTXD1	I	0	PD	2mA	1	0
PAD_LCM_RST	I	0	PD	2mA	1	0
PAD_DSI_TE	I	0	PD	2mA	1	0
PAD_MSDC2_CMD	I	0	PD	2mA	1	0
PAD_MSDC2_CLK	I	0	PD	2mA	1	0
PAD_MSDC2_DAT0	I	0	PD	2mA	1	0
PAD_MSDC2_DAT1	I	0	PD	2mA	1	0
PAD_MSDC2_DAT2	I	0	PD	2mA	1	0
PAD_MSDC2_DAT3	I	0	PD	2mA	1	0
PAD_TDN3	-	-	-	-	-	-
PAD_TDP3	-	-	-	-	-	-
PAD_TDN2	-	-	-	-	-	-
PAD_TDP2	-	-	-	-	-	-
PAD_TCN	-	-	-	-	-	-
PAD_TCP	-	-	-	-	-	-
PAD_TDN1	-	-	-	-	-	-
PAD_TDP1	-	-	-	-	-	-
PAD_TDN0	-	-	-	-	-	-
PAD_TDP0	-	-	-	-	-	-
PAD_RDN0	-	-	-	-	-	-
PAD_RDP0	-	-	-	-	-	-
PAD_RDN1	-	-	-	-	-	-

Name	Reset					
	State	Aux	PU/PD	Driving	IES	SMT
PAD_RDP1	-	-	-	-	-	-
PAD_RCN	-	-	-	-	-	-
PAD_RCP	-	-	-	-	-	-
PAD_RDN2	-	-	-	-	-	-
PAD_RDP2	-	-	-	-	-	-
PAD_RDN3	-	-	-	-	-	-
PAD_RDP3	-	-	-	-	-	-
PAD_RCN_A	-	-	-	-	-	-
PAD_RCP_A	-	-	-	-	-	-
PAD_RDN1_A	-	-	-	-	-	-
PAD_RDP1_A	-	-	-	-	-	-
PAD_RDN0_A	-	-	-	-	-	-
PAD_RDP0_A	-	-	-	-	-	-
PAD_CMDAT0	I	0	PD	2mA	1	0
PAD_CMDAT1	I	0	PD	2mA	1	0
PAD_CMMCLK	OL	1	PD	2mA	1	0
PAD_CMPCLK	I	0	PD	2mA	1	0
PAD_MSDC1_CMD	I	1	PU	2mA	1	0
PAD_MSDC1_CLK	OL	1	PD	2mA	1	0
PAD_MSDC1_DAT0	I	1	PU	2mA	1	0
PAD_MSDC1_DAT1	I	1	PU	2mA	1	0
PAD_MSDC1_DAT2	I	1	PU	2mA	1	0
PAD_MSDC1_DAT3	I	1	PU	2mA	1	0
PAD_MSDC0_DAT7	I	1	PU	2mA	1	0
PAD_MSDC0_DAT6	I	1	PU	2mA	1	0
PAD_MSDC0_DAT5	I	1	PU	2mA	1	0
PAD_MSDC0_DAT4	I	1	PU	2mA	1	0
PAD_MSDC0_RSTB	OH	1	PU	2mA	1	0
PAD_MSDC0_CMD	I	1	PU	2mA	1	0
PAD_MSDC0_CLK	OL	1	PD	2mA	1	0
PAD_MSDC0_DAT3	I	1	PU	2mA	1	0
PAD_MSDC0_DAT2	I	1	PU	2mA	1	0
PAD_MSDC0_DAT1	I	1	PU	2mA	1	0
PAD_MSDC0_DAT0	I	1	PU	2mA	1	0
PAD_CEC	I	1	NO-PULL	2mA	1	0
PAD_HTPLG	I	1	NO-PULL	2mA	1	0
PAD_HDMISCK	I	1	NO-PULL	2mA	1	0
PAD_HDMISD	I	1	NO-PULL	2mA	1	0

The GPIO corresponding configure registers address and bit number are shown below.

Table 5-3. GPIO Configuration Registers Summary

Name	SMT	IES	PUPD/R1/R0	PULLEN	PULLSEL	DRV
PAD_EINT0	0x10005A00[2]	0x10005900[2]	-	0x10005500[0]	0x10005600[0]	0x10005D00[2:1]
PAD_EINT1	0x10005A00[2]	0x10005900[2]	-	0x10005500[1]	0x10005600[1]	0x10005D00[2:1]
PAD_EINT2	0x10005A00[2]	0x10005900[2]	-	0x10005500[2]	0x10005600[2]	0x10005D00[2:1]
PAD_EINT3	0x10005A00[2]	0x10005900[2]	-	0x10005500[3]	0x10005600[3]	0x10005D00[2:1]
PAD_EINT4	0x10005A00[2]	0x10005900[2]	-	0x10005500[4]	0x10005600[4]	0x10005D00[2:1]
PAD_EINT5	0x10005A00[2]	0x10005900[2]	-	0x10005500[5]	0x10005600[5]	0x10005D00[6:5]
PAD_EINT6	0x10005A00[2]	0x10005900[2]	-	0x10005500[6]	0x10005600[6]	0x10005D00[6:5]
PAD_EINT7	0x10005A00[3]	0x10005900[3]	-	0x10005500[7]	0x10005600[7]	0x10005D00[6:5]
PAD_EINT8	0x10005A00[3]	0x10005900[3]	-	0x10005500[8]	0x10005600[8]	0x10005D00[6:5]
PAD_EINT9	0x10005A00[3]	0x10005900[3]	-	0x10005500[9]	0x10005600[9]	0x10005D00[6:5]
PAD_EINT10	0x10005A00[3]	0x10005900[3]	-	0x10005500[10]	0x10005600[10]	0x10005D00[6:5]
PAD_EINT11	0x10005A00[12]	0x10005900[12]	-	0x10005500[11]	0x10005600[11]	0x10005D00[10:8]
PAD_EINT12	0x10005A00[12]	0x10005900[12]	-	0x10005500[12]	0x10005600[12]	0x10005D00[10:8]
PAD_EINT13	0x10005A00[12]	0x10005900[12]	-	0x10005500[13]	0x10005600[13]	0x10005D00[10:8]
PAD_EINT14	0x10005A00[13]	0x10005900[13]	0x10005E50[14:12]	-	-	0x10005D00[14:12]
PAD_EINT15	0x10005A00[13]	0x10005900[13]	0x10005E60[2:0]	-	-	0x10005D00[14:12]
PAD_EINT16	0x10005A00[13]	0x10005900[13]	0x10005E60[6:4]	-	-	0x10005D00[14:12]
PAD_EINT17	0x10005A00[13]	0x10005900[13]	0x10005E60[10:8]	-	-	0x10005D00[14:12]
PAD_EINT18	0x10005A10[10]	0x10005910[10]	-	0x10005510[2]	0x10005610[2]	0x10005D10[2:1]
PAD_EINT19	0x10005A10[10]	0x10005910[10]	-	0x10005510[3]	0x10005610[3]	0x10005D10[2:1]
PAD_EINT20	0x10005A10[10]	0x10005910[10]	-	0x10005510[4]	0x10005610[4]	0x10005D10[2:1]
PAD_EINT21	0x10005A00[13]	0x10005900[13]	0x10005E60[14:12]	-	-	0x10005D00[14:12]
PAD_EINT22	0x10005A00[13]	0x10005900[13]	0x10005E70[2:0]	-	-	0x10005D00[14:12]
PAD_EINT23	0x10005A00[13]	0x10005900[13]	0x10005E70[6:4]	-	-	0x10005D00[14:12]
PAD_EINT24	0x10005A00[12]	0x10005900[12]	-	0x10005510[8]	0x10005610[8]	0x10005D00[10:8]
PAD_EINT25	0x10005A00[12]	0x10005900[12]	-	0x10005510[9]	0x10005610[9]	0x10005D00[10:8]
PAD_PWRAP_SPI0_MI	0x10005A00[0]	0x10005900[0]	-	0x10005510[10]	0x10005610[10]	0x10005D10[6:5]
PAD_PWRAP_SPI0_MO	0x10005A00[0]	0x10005900[0]	-	0x10005510[11]	0x10005610[11]	0x10005D10[6:5]
PAD_PWRAP_INT	0x10005A00[0]	0x10005900[0]	-	0x10005510[12]	0x10005610[12]	0x10005D10[6:5]
PAD_PWRAP_SPI0_CK	0x10005A00[0]	0x10005900[0]	-	0x10005510[13]	0x10005610[13]	0x10005D10[6:5]
PAD_PWRAP_SPI0_CSN	0x10005A00[0]	0x10005900[0]	-	0x10005510[14]	0x10005610[14]	0x10005D10[6:5]
PAD_RTC32K_CK	0x10005A00[1]	0x10005900[1]	-	0x10005510[15]	0x10005610[15]	0x10005D10[10:9]
PAD_WATCHDOG	0x10005A00[1]	0x10005900[1]	-	0x10005520[0]	0x10005620[0]	0x10005D10[10:9]
PAD_SRCLKENA	0x10005A00[1]	0x10005900[1]	-	0x10005520[1]	0x10005620[1]	0x10005D10[10:9]
PAD_URXD2	0x10005A00[2]	0x10005900[2]	-	0x10005520[2]	0x10005620[2]	0x10005D10[14:13]
PAD_UTXD2	0x10005A00[2]	0x10005900[2]	-	0x10005520[3]	0x10005620[3]	0x10005D10[14:13]
PAD_MRG_CLK	0x10005A00[2]	0x10005900[2]	-	0x10005520[4]	0x10005620[4]	0x10005D20[2:1]
PAD_MRG_SYNC	0x10005A00[2]	0x10005900[2]	-	0x10005520[5]	0x10005620[5]	0x10005D20[2:1]
PAD_MRG_DI	0x10005A00[2]	0x10005900[2]	-	0x10005520[6]	0x10005620[6]	0x10005D20[2:1]
PAD_MRG_DO	0x10005A00[2]	0x10005900[2]	-	0x10005520[7]	0x10005620[7]	0x10005D20[2:1]
PAD_KPROW0	0x10005A10[11]	0x10005910[11]	0x10005E80[2:0]	-	-	0x10005D20[6:5]
PAD_KPROW1	0x10005A00[10]	0x10005900[10]	0x10005E80[6:4]	-	-	0x10005D20[10:9]
PAD_KPCOL0	0x10005A00[10]	0x10005900[10]	0x10005E90[2:0]	-	-	0x10005D20[10:9]

Name	SMT	IES	PUPD/R1/R0	PULLEN	PULLSEL	DRV
PAD_KPCOL1	0x10005A00[10]	0x10005900[10]	0x10005E90[6:4]	-	-	0x10005D20[10:9]
PAD_JTMS	0x10005A00[11]	0x10005900[11]	-	0x10005520[12]	0x10005620[12]	0x10005D20[14:13]
PAD_JTCK	0x10005A00[11]	0x10005900[11]	-	0x10005520[13]	0x10005620[13]	0x10005D20[14:13]
PAD_JTDI	0x10005A00[11]	0x10005900[11]	-	0x10005520[14]	0x10005620[14]	0x10005D20[14:13]
PAD_JTDO	0x10005A00[11]	0x10005900[11]	-	0x10005520[15]	0x10005620[15]	0x10005D20[14:13]
PAD_SPI_CS	0x10005A00[14]	0x10005900[14]	-	0x10005530[0]	0x10005630[0]	0x10005D30[2:1]
PAD_SPI_CK	0x10005A00[14]	0x10005900[14]	-	0x10005530[1]	0x10005630[1]	0x10005D30[2:1]
PAD_SPI_MI	0x10005A00[14]	0x10005900[14]	-	0x10005530[2]	0x10005630[2]	0x10005D30[2:1]
PAD_SPI_MO	0x10005A00[14]	0x10005900[14]	-	0x10005530[3]	0x10005630[3]	0x10005D30[2:1]
PAD_SDA1	0x10005A10[0]	0x10005910[0]	-	0x10005530[4]	0x10005630[4]	-
PAD_SCL1	0x10005A10[0]	0x10005910[0]	-	0x10005530[5]	0x10005630[5]	-
PAD_DISP_PWM	0x10005A10[2]	0x10005910[2]	-	0x10005530[6]	0x10005630[6]	0x10005D30[10:9]
PAD_I2S_DATA_IN	0x10005A10[4]	0x10005910[4]	-	0x10005530[7]	0x10005630[7]	0x10005D30[14:13]
PAD_I2S_LRCK	0x10005A10[4]	0x10005910[4]	-	0x10005530[8]	0x10005630[8]	0x10005D30[14:13]
PAD_I2S_BCK	0x10005A10[4]	0x10005910[4]	-	0x10005530[9]	0x10005630[9]	0x10005D30[14:13]
PAD_SDA0	0x10005A00[15]	0x10005900[15]	-	0x10005530[10]	0x10005630[10]	-
PAD_SCL0	0x10005A00[15]	0x10005900[15]	-	0x10005530[11]	0x10005630[11]	-
PAD_SDA2	0x10005A10[1]	0x10005910[1]	-	0x10005530[12]	0x10005630[12]	-
PAD_SCL2	0x10005A10[1]	0x10005910[1]	-	0x10005530[13]	0x10005630[13]	-
PAD_URXD0	0x10005A10[5]	0x10005910[5]	-	0x10005530[14]	0x10005630[14]	0x10005D40[10:9]
PAD_UTXD0	0x10005A10[5]	0x10005910[5]	-	0x10005530[15]	0x10005630[15]	0x10005D40[10:9]
PAD_URXD1	0x10005A10[5]	0x10005910[5]	-	0x10005540[0]	0x10005640[0]	0x10005D40[10:9]
PAD_UTXD1	0x10005A10[5]	0x10005910[5]	-	0x10005540[1]	0x10005640[1]	0x10005D40[10:9]
PAD_LCM_RST	0x10005A10[6]	0x10005910[6]	-	0x10005540[2]	0x10005640[2]	0x10005D40[10:9]
PAD_DSI_TE	0x10005A10[6]	0x10005910[6]	-	0x10005540[3]	0x10005640[3]	0x10005D40[10:9]
PAD_MSDC2_CMD	0x10005A30[2]	0x10005930[2]	0x10005E50[10:8]	-	-	0x10005D40[14:12]
PAD_MSDC2_CLK	0x10005A30[1]	0x10005930[1]	0x10005E50[6:4]	-	-	0x10005D50[2:0]
PAD_MSDC2_DAT0	0x10005A30[3]	0x10005930[6]	0x10005E40[6:4]	-	-	0x10005D50[6:4]
PAD_MSDC2_DAT1	0x10005A30[4]	0x10005930[5]	0x10005E40[10:8]	-	-	0x10005D50[6:4]
PAD_MSDC2_DAT2	0x10005A30[5]	0x10005930[4]	0x10005E40[14:12]	-	-	0x10005D50[6:4]
PAD_MSDC2_DAT3	0x10005A30[6]	0x10005930[3]	0x10005E50[2:0]	-	-	0x10005D50[6:4]
PAD_TDN3	-	-	-	-	-	-
PAD_TDP3	-	-	-	-	-	-
PAD_TDN2	-	-	-	-	-	-
PAD_TDP2	-	-	-	-	-	-
PAD_TCN	-	-	-	-	-	-
PAD_TCP	-	-	-	-	-	-
PAD_TDN1	-	-	-	-	-	-
PAD_TDP1	-	-	-	-	-	-
PAD_TDN0	-	-	-	-	-	-
PAD_TDP0	-	-	-	-	-	-
PAD_RDN0	-	-	-	-	-	-
PAD_RDP0	-	-	-	-	-	-

Name	SMT	IES	PUPD/R1/R0	PULLEN	PULLSEL	DRV
PAD_RDN1	-	-	-	-	-	-
PAD_RDP1	-	-	-	-	-	-
PAD_RCN	-	-	-	-	-	-
PAD_RCP	-	-	-	-	-	-
PAD_RDN2	-	-	-	-	-	-
PAD_RDP2	-	-	-	-	-	-
PAD_RDN3	-	-	-	-	-	-
PAD_RDP3	-	-	-	-	-	-
PAD_RCN_A	-	-	-	-	-	-
PAD_RCP_A	-	-	-	-	-	-
PAD_RDN1_A	-	-	-	-	-	-
PAD_RDP1_A	-	-	-	-	-	-
PAD_RDN0_A	-	-	-	-	-	-
PAD_RDP0_A	-	-	-	-	-	-
PAD_CMDAT0	0x10005A10[7]	0x10005910[7]	-	0x10005560[4]	0x10005660[4]	0x10005D50[10:9]
PAD_CMDAT1	0x10005A10[7]	0x10005910[7]	-	0x10005560[5]	0x10005660[5]	0x10005D50[10:9]
PAD_CMMCLK	0x10005A10[7]	0x10005910[7]	-	0x10005560[6]	0x10005660[6]	0x10005D50[10:9]
PAD_CMPCLK	0x10005A10[7]	0x10005910[7]	-	0x10005560[7]	0x10005660[7]	0x10005D50[10:9]
PAD_MSDC1_CMD	0x10005A20[12]	0x10005920[12]	0x10005E40[2:0]	-	-	0x10005D50[14:12]
PAD_MSDC1_CLK	0x10005A20[11]	0x10005920[11]	0x10005E30[14:12]	-	-	0x10005D60[2:0]
PAD_MSDC1_DAT0	0x10005A20[13]	0x10005930[0]	0x10005E20[14:12]	-	-	0x10005D60[6:4]
PAD_MSDC1_DAT1	0x10005A20[14]	0x10005920[15]	0x10005E30[2:0]	-	-	0x10005D60[6:4]
PAD_MSDC1_DAT2	0x10005A20[15]	0x10005920[14]	0x10005E30[6:4]	-	-	0x10005D60[6:4]
PAD_MSDC1_DAT3	0x10005A30[0]	0x10005920[13]	0x10005E30[10:8]	-	-	0x10005D60[6:4]
PAD_MSDC0_DAT7	0x10005A20[9]	0x10005920[9]	0x10005E10[14:12]	-	-	0x10005D70[2:0]
PAD_MSDC0_DAT6	0x10005A20[8]	0x10005920[8]	0x10005E10[10:8]	-	-	0x10005D70[2:0]
PAD_MSDC0_DAT5	0x10005A20[7]	0x10005920[7]	0x10005E10[6:4]	-	-	0x10005D70[2:0]
PAD_MSDC0_DAT4	0x10005A20[6]	0x10005920[6]	0x10005E10[2:0]	-	-	0x10005D70[2:0]
PAD_MSDC0_RSTB	0x10005A20[10]	0x10005920[10]	0x10005E20[10:8]	-	-	0x10005D70[6:4]
PAD_MSDC0_CMD	0x10005A20[1]	0x10005920[1]	0x10005E20[2:0]	-	-	0x10005D60[14:12]
PAD_MSDC0_CLK	0x10005A20[0]	0x10005920[0]	0x10005E20[6:4]	-	-	0x10005D60[10:8]
PAD_MSDC0_DAT3	0x10005A20[5]	0x10005920[5]	0x10005E00[14:12]	-	-	0x10005D70[2:0]
PAD_MSDC0_DAT2	0x10005A20[4]	0x10005920[4]	0x10005E00[10:8]	-	-	0x10005D70[2:0]
PAD_MSDC0_DAT1	0x10005A20[3]	0x10005920[3]	0x10005E00[6:4]	-	-	0x10005D70[2:0]
PAD_MSDC0_DAT0	0x10005A20[2]	0x10005920[2]	0x10005E00[2:0]	-	-	0x10005D70[2:0]
PAD_CEC	0x10005A10[9]	0x10005910[9]	-	0x10005570[9]	0x10005670[9]	-
PAD_HTPLG	0x10005A10[9]	0x10005910[9]	-	0x10005570[10]	0x10005670[10]	-
PAD_HDMISCK	0x10005A10[9]	0x10005910[9]	-	0x10005570[11]	0x10005670[11]	-
PAD_HDMISD	0x10005A10[9]	0x10005910[9]	-	0x10005570[12]	0x10005670[12]	-

5.1.4 Register Definitions

For register details refer to chapter 3.1 of “MT8516A Application Processor Registers.”

5.2 Peripheral Configuration Controller (pericfg)

5.2.1 Introduction

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

5.2.2 Features

Pericfg provides the following control signals to the functional blocks inside the peripheral system:

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

5.2.3 Pericfg Block Diagram

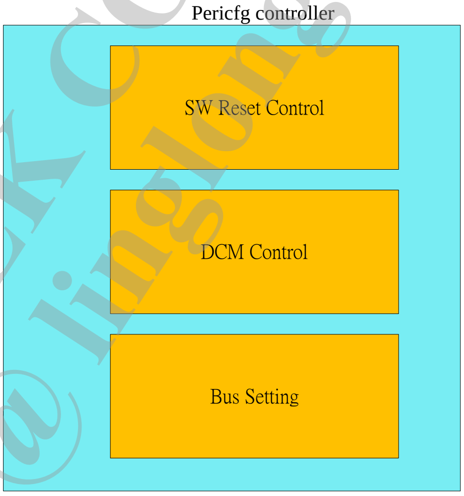


Figure 5-2. Pericfg Controller Block Diagram

5.2.4 Register Definitions

For register details refer to chapter 3.2 of “MT8516A Application Processor Registers”.

5.3 Keypad Scanner

5.3.1 General Description

The keypad supports two types of keypads: 2*2 single keys and 2*2 configurable double keys.

The 2*2 keypad can be divided into two parts:

- The keypad interface including 2 columns and 2 rows (see Figure 5-3 and Figure 5-4)
- The key detection block provides key pressed, key released and de-bounce mechanisms.

Each time the key is pressed or released, i.e. something different in the 8x8 matrix, the key detection block senses the change and recognizes if a key has been pressed or released. Whenever the key status changes and is stable, a KEYPAD IRQ will be issued. The MCU can then read the key(s) pressed directly in the KP_MEM1, KP_MEM2, KP_MEM3, KP_MEM4 and KP_MEM5 registers. To ensure the key pressed 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-pressed detection FSM.

This keypad detects one or two keys pressed simultaneously with any combination. Figure 5-7 Figure 5-8shows the one key pressed condition. Figure 5-8 (a) and (b) illustrate the cases of two keys pressed. 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 on the same column and the same row with other keys, the pressed key cannot be correctly decoded. For example, if there are three key 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.

The 2*2 double keypad supports a 2*2*2 = 8 keys matrix. The eight keys are divided into four sub groups, and each group consists of 2 keys and a 20 ohm resistor. Besides the limitation of the 2*2 keypad, 2*2 double keypad has another limitation, which is it cannot detect two keys pressed simultaneously when the two keys are in one group, i.e. the 2*2 double keypad cannot detect key 0 and key 1 pressed simultaneously or key 3 and key 4 pressed simultaneously.

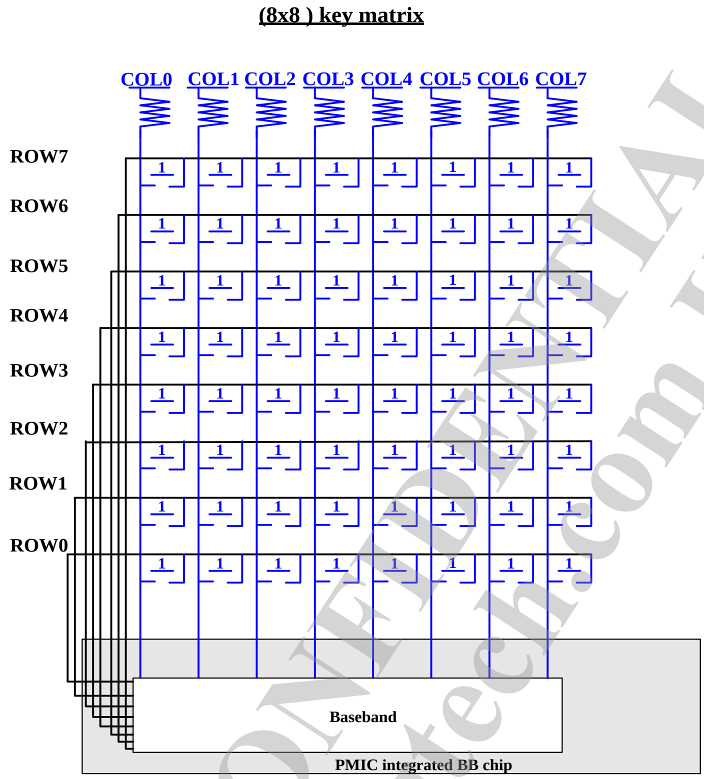


Figure 5-3. 2x2 Keypad Matrix (4 Keys)

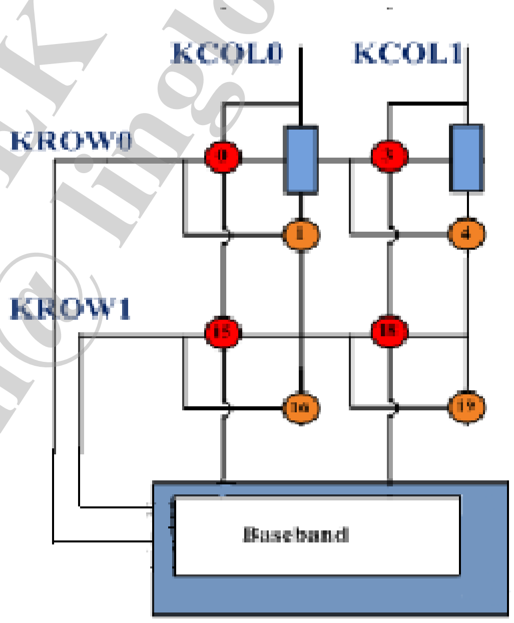


Figure 5-4. 2x2 Keypad Matrix (8 Keys)

5.3.2 Waveform

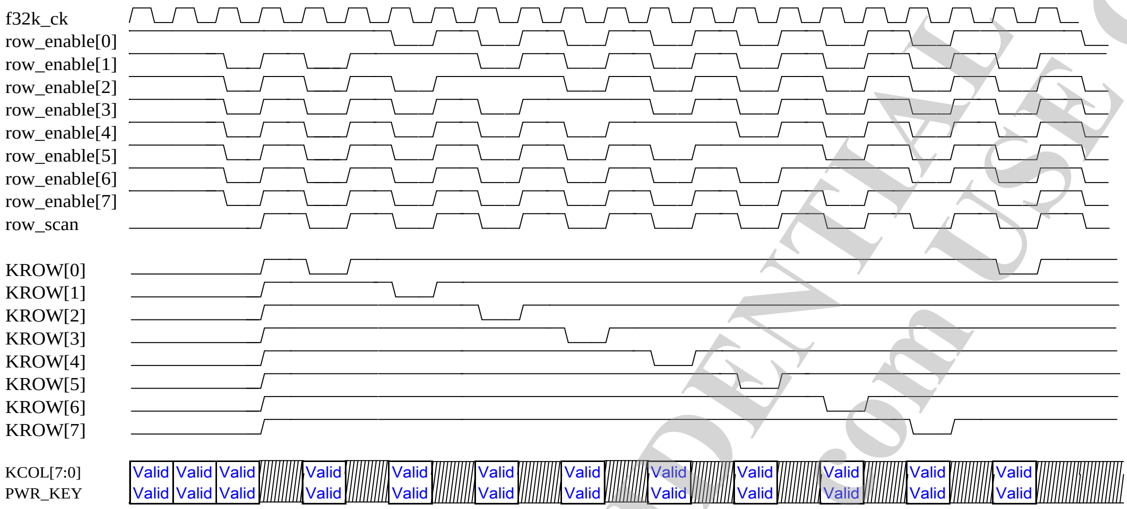


Figure 5-5. 8x8 Keypad Scan Waveform

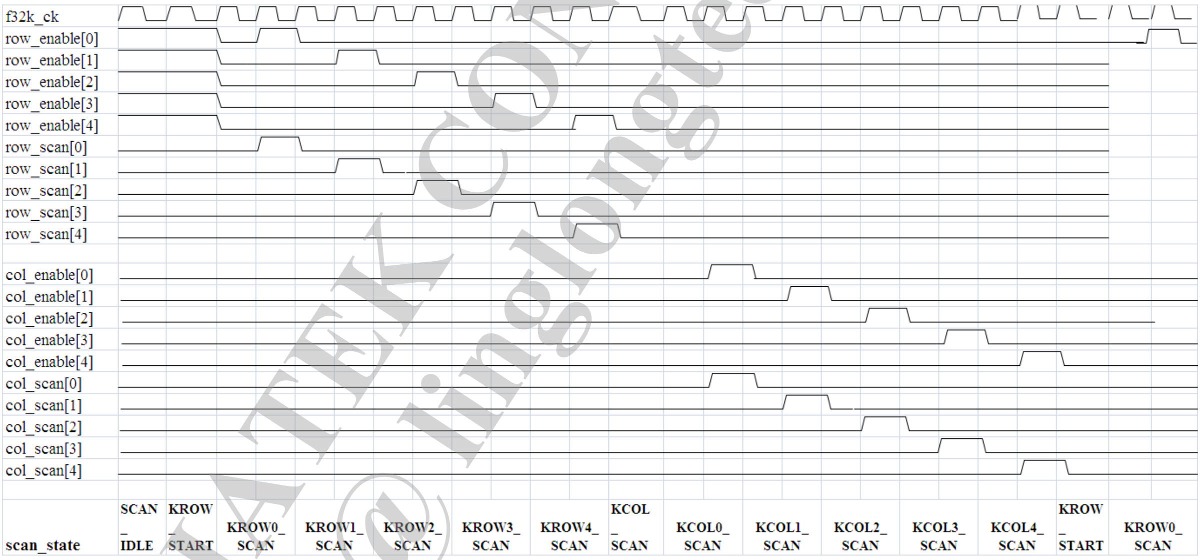


*Figure 5-6. 5*5 Keypad Scan Waveform*

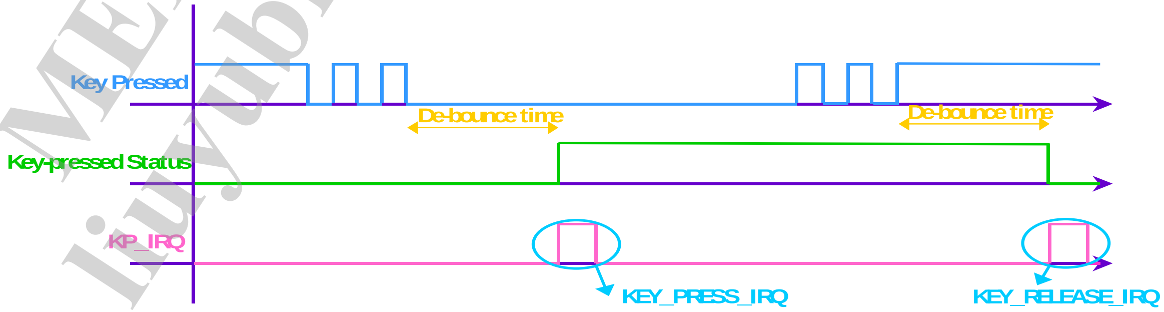


Figure 5-7. One Key Pressed with De-bounce Mechanism Denoted

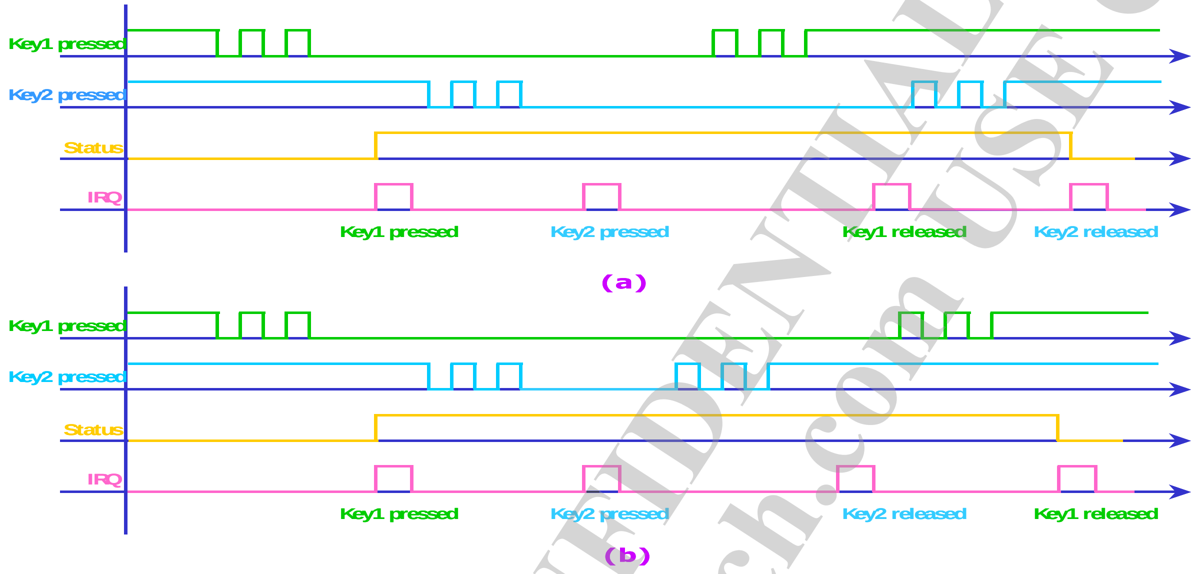


Figure 5-8. (a) Two Keys Pressed, Case 1; (b) Two Keys Pressed, Case 2

5.3.3 Register Definitions

For register details refer to chapter 3.3 in “MT8516A Application Processor Registers.”

5.4 UART

5.4.1 Introduction

The baseband chipset houses four UARTs. UARTs provide full duplex serial communication channels between the baseband chipset and external devices.

UART has both M16C450 and M16550A modes of operation, 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 M16550A, the UART supports word lengths from 5 to 8 bits, an optional parity bit and one or two stop bits and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Two modem control lines and a diagnostic loop-back mode are provided. UART also includes two DMA handshake lines, indicating when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the ten sources.

Note that UART is designed so that all internal operation is synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and industry standard 16550A device, which means that the core is not clock for clock identical to the original device.

After hardware reset, UART will be in M16C450 mode; its FIFOs can then be enabled and UART can enter M16550A mode. UART has further additional functions beyond the M16550A mode. Each of the extended functions can be selected individually under software control.

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

Hardware flow control
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.

Note that in order to enable the enhancements, the enhanced mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:4], FCR[5:4], cannot be written and MCR[7] cannot be read. The enhanced mode bit ensures that UART is backward compatible with the software that has been written for 16C450 and 16550A devices.

5.4.2 Features

- Provides three channels
- DMA, polling or interrupt operation
- Supports word lengths from 5 to 8 bits, with an optional parity bit and one or two stop bits
- Three UART ports for hardware automatic flow control (UART0, UART1, UART2)
- Supports baud rates from 110bps up to 961,200bps

- Baud rate auto detection function

5.4.3 UART Block Diagram

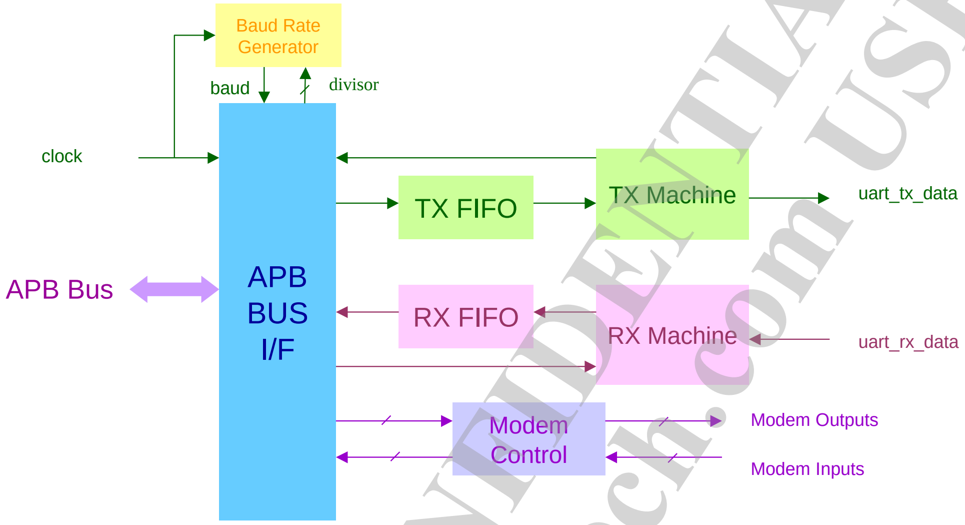


Figure 5-9. UART Block Diagram

5.4.4 Register Definitions

For register details refer to chapter 3.4 in “MT8516A Application Processor Registers.”

5.4.5 Programming Guide

5.4.5.1 Auto Baud Rate Detection

UART can detect the baud rate used automatically. Follow the steps below:

1. Set up register autobaud_en to start detecting the data.
2. Send data of ASCII code “AT” or “at” to UART from the connected host, e.g. the PC.
3. Check if the “AT” or “at” is received. If received, the setting is now already set for further transmission.

5.4.5.2 Transmission

Follow the steps below for UART transmission:

1. Use the auto baud function to set up the baud rate or set up the parameters by yourself. The settings needed can be found in register DLL, DLM ,HIGH SPEED.
2. After setting up the baud rate, start the transmission by filling the TX FIFO and receiving data from RX FIFO.
3. Virtual FIFO can also be used for the transmission. To use the virtual FIFO, you need APDMA settings (refer to details in the APDMA section).

5.5 USB 2.0 High Speed Controller

5.5.1 Introduction

The USB controller is configured for supporting 8 endpoints to receive packets and eight endpoints to send packets except for endpoint 0. These endpoints can be individually configured in the software to handle either Bulk transfers, Interrupt transfers or Isochronous transfers. There are eight DMA channels and the embedded RAM size is configurable size up to 8K bytes. The embedded RAM can be dynamically configured to each endpoint. As the host for point-to-point communications, the controller maintains a frame counter and automatically schedules SOF, Isochronous, Interrupt and Bulk transfers.

5.5.2 Feature List

The following table lists the unified USB IP features.

- USB 2.0 device
- Endpoint: 8Tx, 8Rx, EP0
- 8 DMA channels
- Embedded RAM up to 8KB
- UTM + 16b interface
- CPU slave interface—AHB Asynchronous design
- DMA master interface—AHB busy free asynchronous design

5.5.3 USB Controller Block Diagram

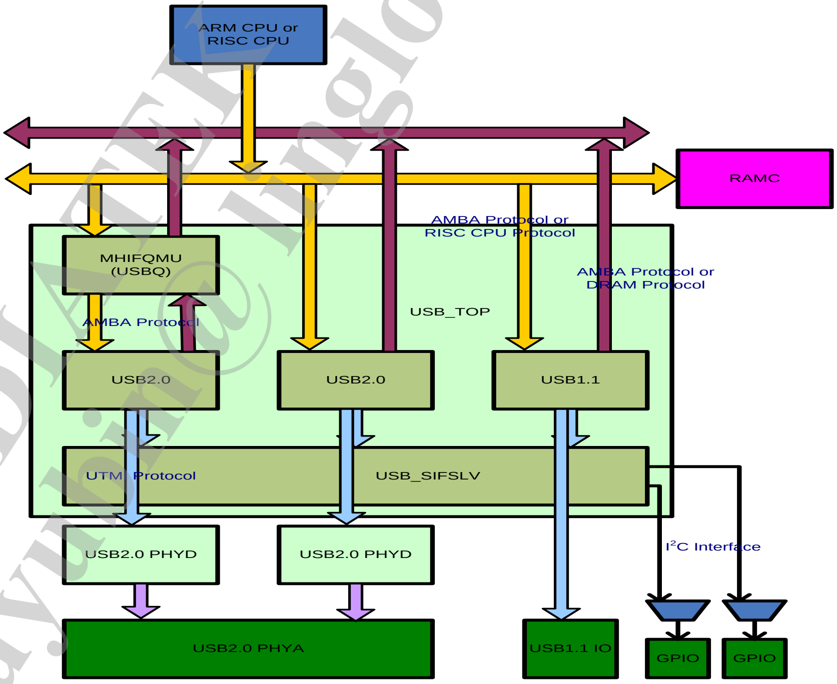


Figure 5-10. USB Controller Block Diagram

5.5.4 Register Definitions

For additional register details refer to chapter 3.5 in "MT8516A Application Processor Registers."

Registers accessed using byte manipulation are marked in blue columns. Byte accessing registers can be accessed using word manipulation. Word accessing registers cannot be accessed using the byte manipulation.

Register address	Register name	Manipulation (Byte/Word)	Acronym
Common Registers			
USB + 0000h	Function address register	Byte	FADDR
USB + 0001h	Power management register	Byte	POWER
USB + 0002h	Tx interrupt status register	Byte	INTRTX
USB + 0004h	Rx interrupt status register	Byte	INTRRX
USB + 0006h	Tx interrupt enable register	Byte	INTRTXE
USB + 0008h	Rx interrupt enable register	Byte	INTRRXE
USB + 000Ah	Common USB interrupts register	Byte	INTRUSB
USB + 000Bh	Common USB interrupts enable register	Byte	INTRUSBE
USB + 000Ch	Frame number register	Byte	FRAME
USB + 000Eh	Endpoint selecting index register	Byte	INDEX
USB + 000Fh	Test mode enable register	Byte	TESTMODE
Indexed EndPoint CSR Region *n stands for endpoint number.* *For example, endpoint 1's n = 1. Valid n = 1 ~ MaxEndPoint.* *MaxEndPoint is hardware configured and the maximum is 15.*			
USB + 0010h ~ USB + 001Fh	It maps to CSR EP0 ~ EPx depending on the INDEX register. For example, if INDEX is n, address 0010h ~ 001Fh are mapped to 0x(100+10*n)h ~ 0x(100+10*n+F)h.	Byte	Indexed CSR
USB + 0020h	USB endpoint 0 FIFO register	Byte	FIFO0
USB + 0020h +(n)*4 h	USB endpoint n FIFO register	Byte	FIFOn
OTG, Dynamic FIFO, Version Registers			
USB + 0060h	OTG device control register	Byte	DEVCTL
USB + 0061h	Power up counter register	Byte	PWRUPCNT
USB + 0062h	Tx FIFO size register	Byte	TXFIFOSZ
USB + 0063h	Rx FIFO size register	Byte	RXFIFOSZ
USB + 0064h	Tx FIFO address register	Byte	TXFIFOADD
USB + 0066h	Rx FIFO address register	Byte	RXFIFOADD
USB + 006Ch	Hardware capability register	Byte	HWCAPS
USB + 006Eh	Hardware sub version register	Byte	HWSVERS
Hardware Configuration, Special Setting Registers			
USB + 0070h	USB bus performance register 1	Byte	BUSPERF1

Register address	Register name	Manipulation (Byte/Word)	Acronym
USB + 0072h	USB bus performance register 2	Byte	BUSPERF2
USB + 0074h	USB bus performance register 3	Byte	BUSPERF3
USB + 0078h	Information about number of Tx and Rx register	Byte	EPINFO
USB + 0079h	Information about the width of RAM and the number of DMA channel register	Byte	RAMINFO
USB + 007Ah	Info. about delay to be applied register	Byte	LINKINFO
USB + 007Bh	Vbus pulsing charge register	Byte	VPLEN
USB + 007Ch	Time buffer available on HS transactions register	Byte	HS_EOF1
USB + 007Dh	Time buffer available on FS transactions register	Byte	FS_EOF1
USB + 007Eh	Time buffer available on LS transactions register	Byte	LS_EOF1
USB + 007Fh	Reset information register	Byte	RST_INFO
USB + 0080h	Rx data toggle set/status register	Word	RXTOG
USB + 0082h	Rx data toggle enable register	Word	RXTOGEN
USB + 0084h	Tx data toggle set/status register	Word	TXTOG
USB + 0086h	Tx data toggle enable register	Word	TXTOGEN
Level1 interrupt Control/Status registers			
USB + 00A0h	USB Level 1 interrupt status register	Byte	USB_L1INTS
USB + 00A4h	USB Level 1 interrupt unmask register	Byte	USB_L1INTM
USB + 00A8h	USB Level 1 interrupt polarity register	Byte	USB_L1INTP
USB + 00ACh	USB Level 1 interrupt control register	Byte	USB_L1INTC
Non-indexed EndPoint CSR Region *n stands for endpoint number.* *For example, endpoint 1's n = 1. Valid n = 1 ~ MaxEndPoint.* *MaxEndPoint is hardware configured and the maximum is 15.*			
USB + 0102h	EP0 control status register	Byte	CSR0
USB + 0108h	EP0 received bytes register	Byte	COUNT0
USB + 010Bh	NAK limit register	Byte	NAKLIMT0
USB + 010Fh	Core configuration register	Byte	CONFIGDATA
USB + 0100h +(n)*10h	TXMAP register	Byte	TXMAP(n)
USB + 0102h +(n)*10h	Tx CSR register	Byte	TXCSR(n)
USB + 0104h +(n)*10h	RXMAP register	Byte	RXMAP(n)
USB + 0106h +(n)*10h	Rx CSR register	Byte	RXCSR(n)
USB + 0108h +(n)*10h	Rx Count register	Byte	RXCOUNT(n)
USB + 010Ah +(n)*10h	TxType register	Byte	TXTYPE(n)
USB + 010Bh	TxInterval register	Byte	TXINTERVAL(n)

Register address	Register name	Manipulation (Byte/Word)	Acronym
+(n)*10h			
USB + 010Ch +(n)*10h	RxType register	Byte	RXTYPE(n)
USB + 010Dh +(n)*10h	RxInterval register	Byte	RXINTERVAL(n)
USB + 010Fh +(n)*10h	Configured FIFO size register	Byte	FIFOSIZE(n)
DMA Channels Control Registers *M stands for DMA channel number.* *For example, DMA channel 1's M = 1. Valid M = 1 ~ MaxDMAChannel.* *MaxDMAChannel is hardware configured and the maximum is 8.*			
USB + 0200h	DMA interrupt status register (word access only)	Word	DMA_INTR
USB + 0210h	DMA limiter register (word access only)	Word	DMA_LIMITER
USB + 0220h	DMA configuration register (word access only)	Word	DMA_CONFIG
USB + 0204h +(M-1)*10h	DMA channel M control register (word access only)	Word	DMA_CNTL_M
USB + 0208h +(M-1)*10h	DMA channel M address register (word access only)	Word	DMA_ADDR_M
USB + 020Ch +(M-1)*10h	DMA channel M byte count register (word access only)	Word	DMA_COUNT_M
EndPoint RX Packet Count Register *n stands for endpoint number. For example, endpoint 1's n = 1. Valid n = 1 ~ MaxEndPoint.* *MaxEndPoint is hardware configured and the maximum is 15.*			
USB + 0300h +(n)*4h	EPn RxPktCount register	Word	EPnRXPKTCOUNT
Host/Hub Control Registers (Host mode only registers) *n stands for endpoint number. For example, endpoint 1's n = 1. Valid n = 1 ~ MaxEndPoint.* *MaxEndPoint is hardware configured and maximum is 15.*			
USB + 0480h +8*n h	Transmit endpoint n function address	Word	TXFUNCADDR
USB + 0482h +8*n h	Transmit endpoint n hub/port address	Word	TXHUBADDR
USB + 0484h +8*n h	Receive endpoint n function address	Word	RXFUNCADDR
USB + 0486h +8*n h	Receive endpoint n hub/port address	Word	RXHUBADDR
Debug Function Registers			
USB + 0600h	Debug flag selection control (byte 0, 1, 2, 3)	Word	DFC0R, DFC1R
USB + 0604h	Timing test mode	Word	TM1
USB + 0605h	No response error count	Word	TM1
USB + 0606h	Debug flag UTMI11 sub group selection	Word	DFC2R
USB + 0608h	Hardware version control register	Word	HWVER_DATE

Register address	Register name	Manipulation (Byte/Word)	Acronym
USB + 0610h	Packet sequence record control/OpState record control	Word	PSR_CTRL/ OSR_CTRL
USB + 0611h ~ 0616h	Packet sequence record filter and trigger setting	Word	PSR_CTRL
USB + 0620h ~ 0637h	Debug register	Word	DBG_PRB
USB + 0640h ~ 065fh	Packet sequence PID data/OpState record Data	Word	PSR_DATA/ OSR_DATA
USB + 0684h	SRAM address register	Word	SRAMA
USB + 0688h	SRAM data register (word access only)	Word	SRAMD
USB + 0690h	RISC_SIZE register	Word	RISC_SIZE
USB + 0700h	Reserved register	Word	RESREG
USB + 0704h	HW TxPktRdy	Word	HWTPR
USB + 0708h	HW TxPktRdy enable register	Word	HWTPR_EN
USB + 070Ch	HW TxPktRdy error detection register	Word	HWTPR_ERR

5.6 USBPHY

The full USBPHY features include USB2.0 PHYD and PHYA macro control registers. It also includes a frequency meter for USB2.0 PHYA monitor clock. The registers can be accessed by I2C interface (FT). The default mode is accessing registers by AHB. After 0xfe (I2C access only) is configured to 8'h01, the register file will be in the I2C mode (accessing registers by I2C).

5.6.1 Features

- USB2.0
- USB2.0 PHYD control registers for PHYD macro setting
- USB2.0 PHYA control registers for PHYA characteristic tuning
- Force USB2.0 UTMI interface for FT tests
- Force USB2.0 PHY analog power-down in ATPG mode
- Frequency meter for USB2.0 PHYA monitor clock
- Accessing PHY registers by AHB slave interface
- Accessing PHY registers by I2C interface

5.6.2 Block Diagram

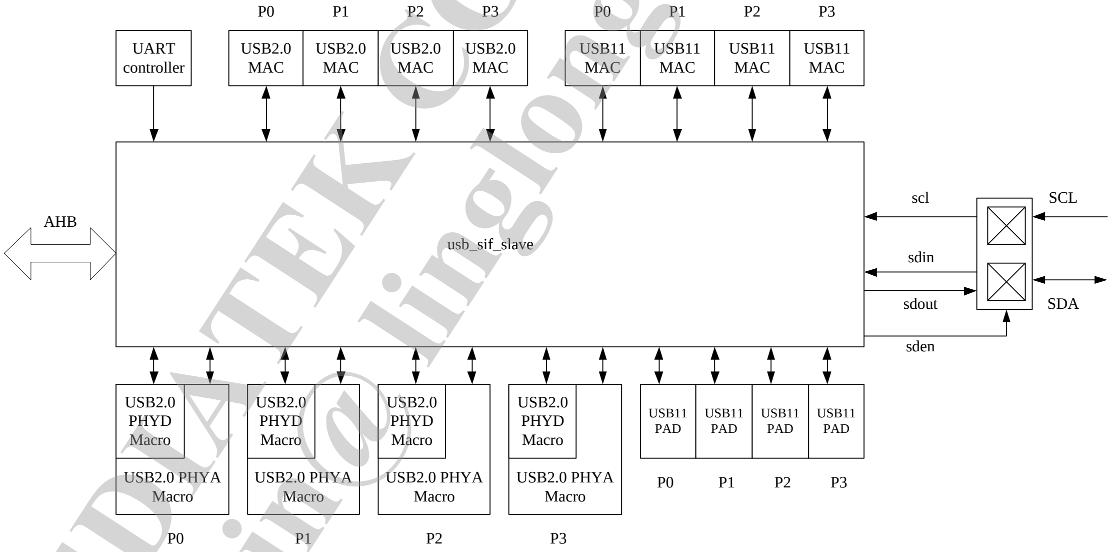


Figure 5-11. USBPHY RegFile Block Diagram

5.6.3 Register Definitions

For register details refer to chapter 3.6 in “MT8516A Application Processor Registers.”

Page base

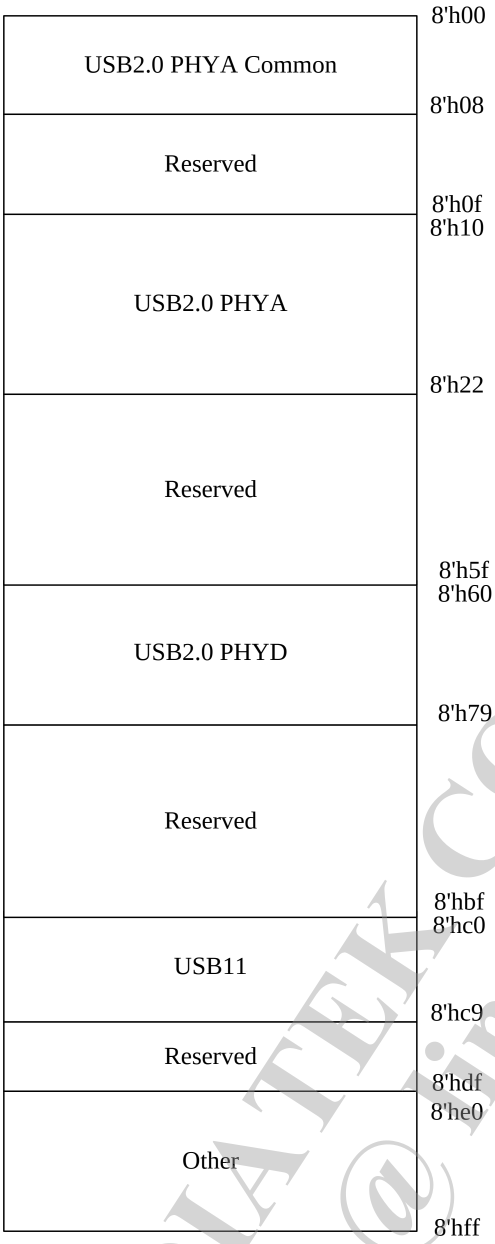
- Every page register contains 0x00h ~ 0xefh (USB2.0+ USB1.1)
- Frequency meter registers in one page (@ 0xff = 8’h0f)
- 0xf0~0xff: Global register
- 0xfe[0]: I2C mode, the default value is 0 (accessing register by AHB interface)
  - If switched to I2C mode, configuring 0xfe[0] to be 1’h1 will be required.
  - I2C access only
- 0xffh (RG_PAGE): I2C page register
  - I2C access only

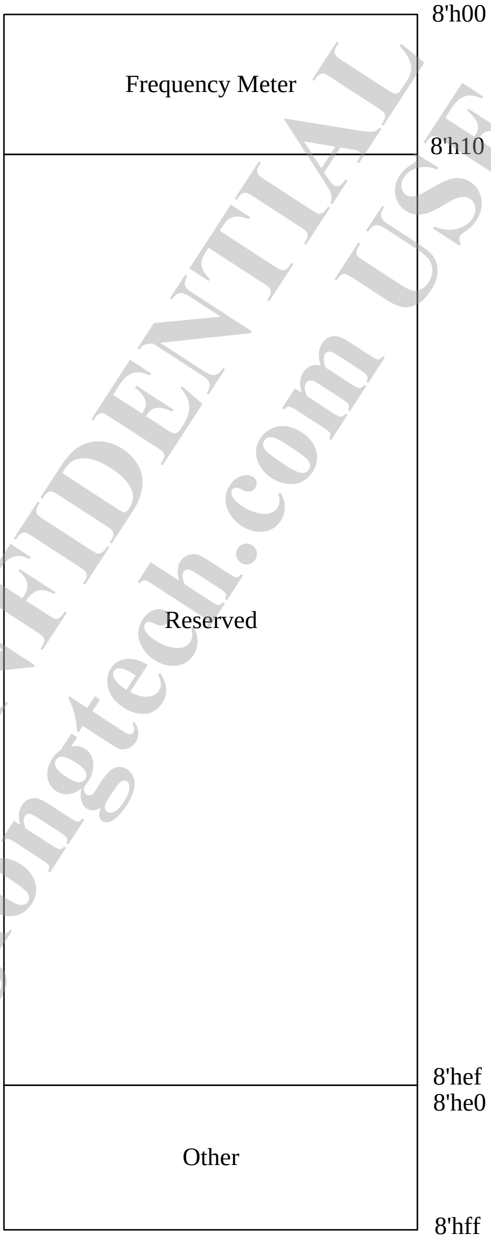
I2C

- Accessing different pages by setting up RG_PAGE
- Default device number: 7’h60
- (*If there are more than one hierarchy, the device number of the second hierarchy will be 7’h61.)
- Accessing different pages by setting up RG_PAGE (0xff)
- Port 0 USB PHY register page : RG_PAGE value : 8’h00
- Frequency Meter register page : RG_PAGE value : 8’h0f

AHB

- Accessing different pages by different base addresses
- Supporting one port. Base address: 800h
  - Port 0 register base address: 800h
  - Frequency meter registers base address: f00h


8'h00
USB2.0 PHYA Common
8'h08
Reserved
8'h0f
8'h10
USB2.0 PHYA
8'h22
Reserved
8'h5f
8'h60
USB2.0 PHYD
8'h79
Reserved
8'hbf
8'hc0
USB11
8'hc9
Reserved
8'hdf
8'he0
Other
8'hff


8'h00
Frequency Meter
8'h10
Reserved
8'hef
8'he0
Other
8'hff

5.7 SPI Interface Controller

5.7.1 Introduction

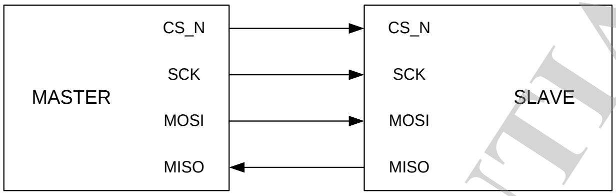


Figure 5-12. Pin Connection between SPI Master and SPI Slave

The SPI interface is a bit-serial, four-pin transmission protocol. The figure above is an example of the connection between the SPI master and SPI slave. The SPI interface controller is a master responsible of the data transmission with the slave.

5.7.2 SPI Block Diagram

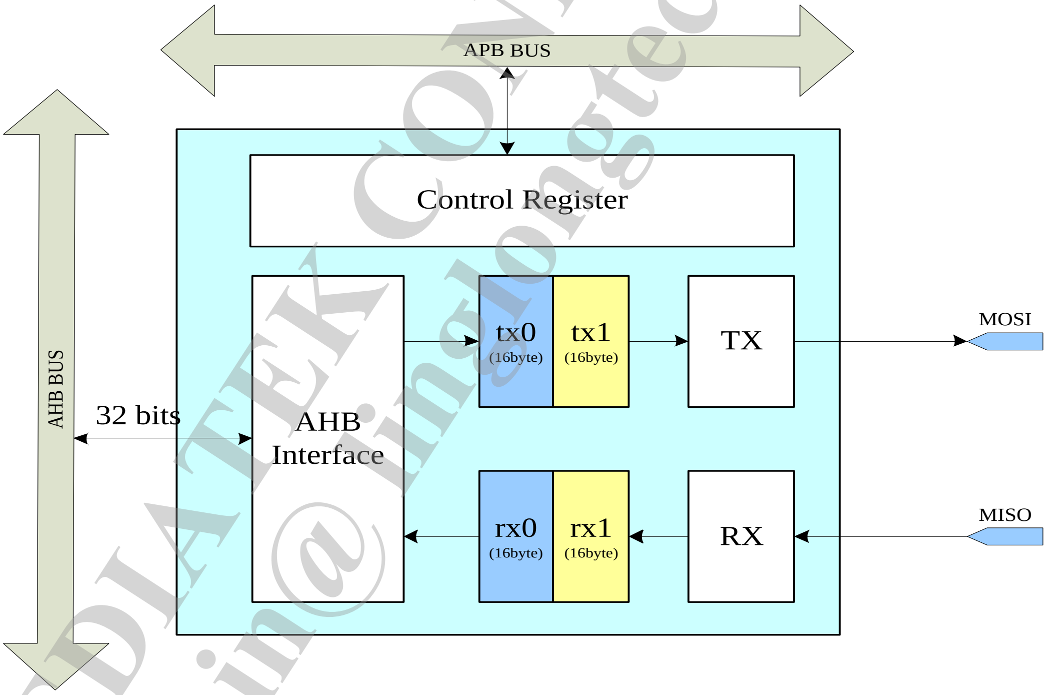


Figure-5-13. SPI Block Diagram

5.7.3 Pin Description

Table 5-4. SPI Controller Interface

Signal name	Type	Description
CS_N	O	Low active chip selection signal

Signal name	Type	Description
SCK	O	The (bit) serial clock
MOSI	O	Data signal from master output to slave input
MISO	I	Data signal from slave output to master input

5.7.4 Transmission Formats

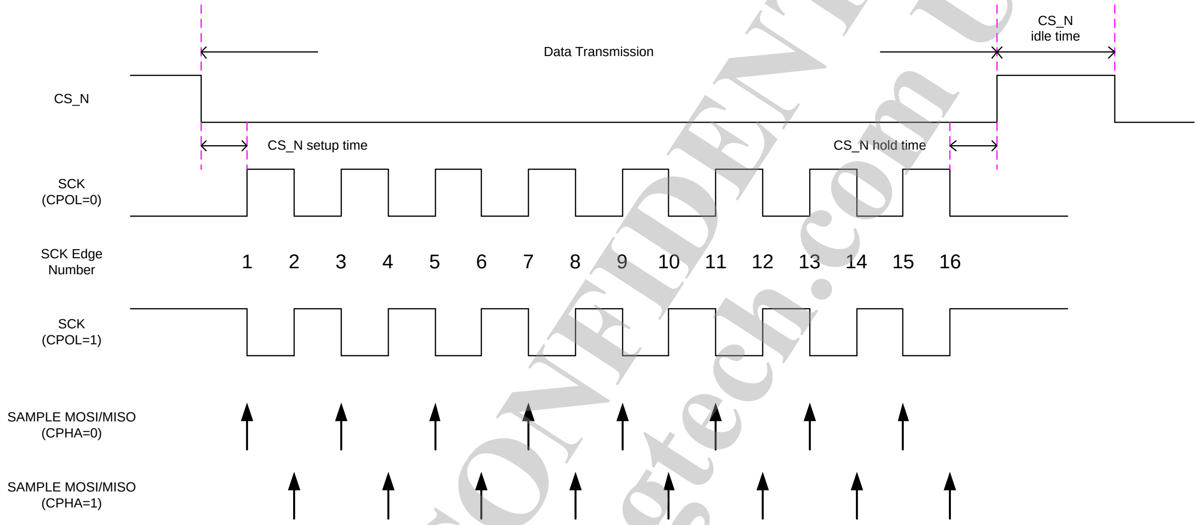


Figure 5-14. SPI Transmission Formats

The figure above 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. This figure is an example of both clock polarities (CPOL).

CPHA defines the legal timing to sample MOSI and MISO. Two different methods can be adopted.

5.7.5 Features

The features of the SPI controller (master) are:

- Configurable CS_N setup time, hold time and idle time
- Programmable SCK high time and low time
- Configurable transmitting and receiving bit order

- Two configurable modes for the source of the data to be transmitted. 1) In TX DMA mode, the SPI controller automatically fetches the transmitted data (to be put on the MOSI line) from memory; 2) 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. 1) In RX DMA mode, the SPI controller automatically stores the received data (from MISO line) to memory; 2) In RX FIFO mode, the received data keep being in RX FIFO of the SPI controller. 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 keep being 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 the figure below.
- 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 5-16 illustrates the waveform in this transmission format.

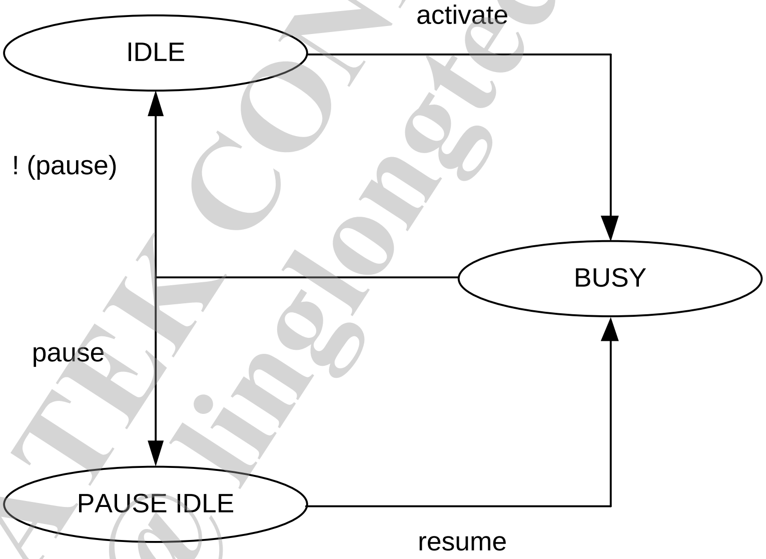


Figure 5-15. Operation Flow with or without PAUSE Mode

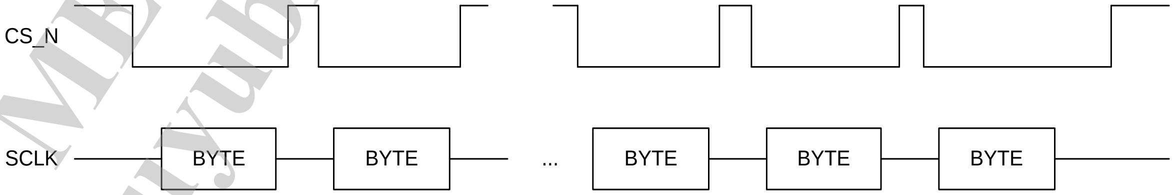


Figure 5-16. CS_N de-assert Mode

5.7.6 Register Definitions

For register details refer to chapter 3.7 in "MT8516A Application Processor Registers."

5.7.7 Programming Guide

Follow the steps below to perform SPI transmission:

1. Prepare the data in the memory with its start address to be the "source address".
2. Set up the timing and protocol for the SPI transmission (see for detailed setup parameters).
3. Fill the "destination address", which is the start address that you 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_DST respectively.
4. Set up the CMD_ACT (bit0 of register SPI_CMD) to start the transfer.
5. Get the data received from the buffer prepared starting from "destination address".

5.8 Memory Stick and SD Card Controller

5.8.1 Introduction

The MSDC (Memory Stick and SD card Controller) fully supports functions listed here. The default mode is accessing registers by AHB.

- SD memory card specification version 3.0
- SDIO card specification version 3.0
- MMC/eMMC 4.5

5.8.2 Features

Each MSDC module is responsible for attach the following features:

- 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 bytes FIFO buffers for transmit and receive
- Built-in CRC circuit
- Basic DMA mode, basic descriptor mode, and enhanced descriptor mode for SD/MMC
- Interrupt capabilities
- Does not support SPI mode for SD/MMC memory card
- Does not support suspend/resume for SD/MMC memory card
- Supports SD3.0 SDR104, data rate up to 208x4Mbps
- Supports SD3.0 DDR50, data rate up to 50x4x2Mbps(4-bit with clock dual edge)
- Supports e-MMC boot-up mode
- 256 programmable serial clock rates on SD/MMC bus from 100kHz to 208MHz
- Card detection capabilities

5.8.3 MSDC Block Diagram

The MSDC controller design includes two parts “msdc_top” and “MSDC pad macro”. Msdc_top is responsible for function implementation; MSDC pad macro is used for intermediate data transmission.

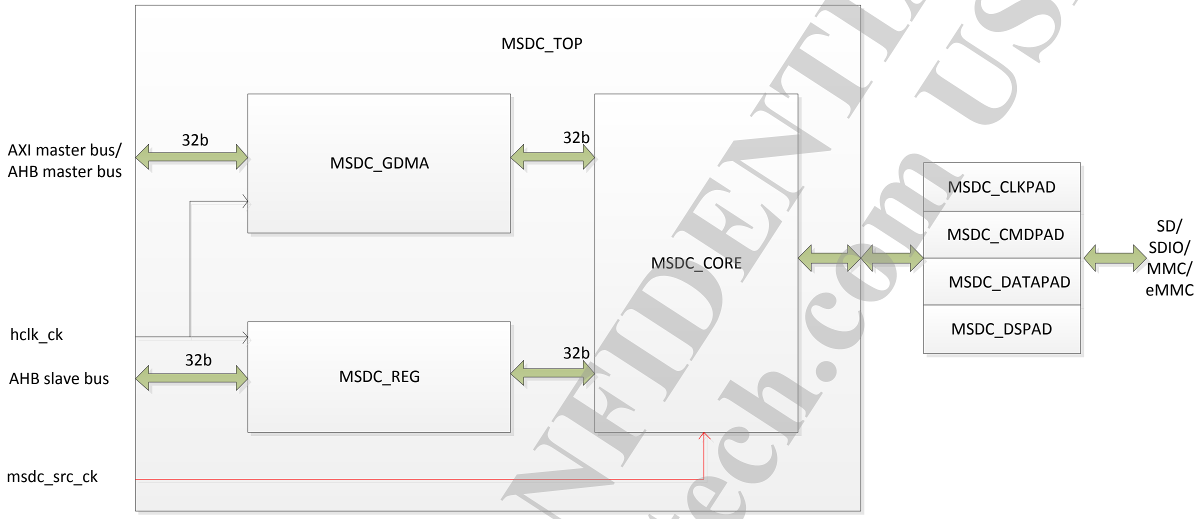


Figure 5-17. MSDC block diagram

Table 5-5. MSDC Functions and Address

MSDC List	Base address	Feature
MSDC0	0x11200000	EMMC4.5
MSDC1	0x11300000	SD3.0/SDIO3.0
MSDC2	0x11700000	SD3.0/SDIO3.0

There are three MSDC IPs in this SOC. Use of the registers are the same except that the base address needs to be changed to respective one.

5.8.4 Theory of Operations

MSDC is general memory device, which designed for mobile devices and aimed to offer the high throughput data transfers and small random data performance by outside devices considering power consumption and data security. The communication protocol between controller and device is based on an advanced 11 or 6 signal bus. Details are provided in the following table.

Table 5-6.Sharing of Pins for MSDC.

No.	Name	Type	MMC	SD	MS	MSPRO	Description
1	eMMC _CLK	O	CLK	CLK	SCLK	SCLK	Clock
2	eMMC DS	I	RCLK				
2	eMMC_DAT0	I/O/PP	DAT0	DAT0	SDIO	DAT0	Serial Data line bit 0
3	eMMC _DAT1	I/O/PP	DAT1	DAT1		DAT1	Serial Data line bit 1
4	eMMC _DAT2	I/O/PP	DAT2	DAT2		DAT2	Serial Data line bit 2
5	eMMC _DAT3	I/O/PP	DAT3	DAT3		DAT3	Serial Data line bit 3
6	eMMC _DAT4	I/O/PP	DAT4				Serial Data line bit 4
7	eMMC _DAT5	I/O/PP	DAT5				Serial Data line bit 5
8	eMMC _DAT6	I/O/PP	DAT6				Serial Data line bit 6
9	eMMC _DAT7	I/O/PP	DAT7				Serial Data line bit 7
10	eMMC _CMD	I/O/PP	CMD	CMD	BS	BS	Command / Bus State
11	SD_WP	I		WP			Write Protect
12	SD_INS	I	VSS2	VSS2	INS	INS	Card insertion

All I/O pads include both pull-up and pull-down resistors because they are shared by both the Memory Stick and SD/MMC Memory Card. Pull-down resistor for these pins can be used for power saving. All embedded pull-up and pull-down resistors can be disabled by programming the corresponding control registers if optimal pull-up or pull-down resistors are required on the system board. The pin SD_WP (Write Protection) is only valid when the controller is configured for SD/MMC Memory Card. It is used to detect the status of Write Protection Switch on SD/MMC Memory Card.

Communication over the SD bus is based on command and data bit streams that are initiated by a start bit and terminated by a stop bit.

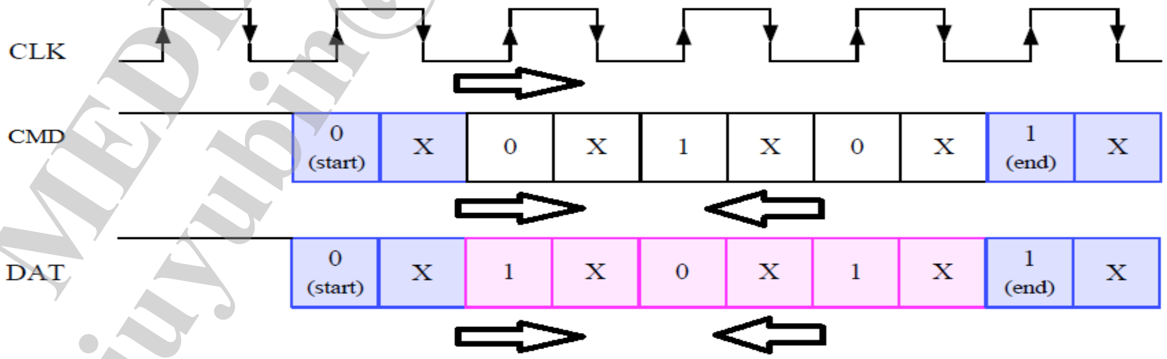


Figure 5-18.MSDC Transfer Waveform

5.8.5 Register Definitions

For register details refer to chapter 3.8 of “MT8516A Application Processor Registers.”

5.9 NAND Flash Interface (NFI)

5.9.1 Introduction

The NFI and ECC engine (in NFI mode) can automatically generate ECC syndrome bits when programming or reading the device. If you allow that it stores the syndrome bits in the spare area for each page, the HW_ECC mode can be used. Otherwise, you can prepare the data (may contains operating system information or ECC syndrome bits) for the spare area with other arrangement. In former cases, 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-bits errors within one sector.

5.9.2 Features

MT8516A provides NAND flash interface SLC/MLC/SPI NAND. The NAND flash interface supports the following features:

- Legacy NAND flash timing control
- Toggle NAND (v1.0) flash timing control
- ONFI NAND (v2.x) 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 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
- Supports 2-chip selection for NAND flash parts.
- Supports 8-bit TOGGLE/ONFI NAND I/O interface.
- CRC16
- Randomizer (TOSHIBA/SAMSUNG)

5.9.3 NFI Block Diagram

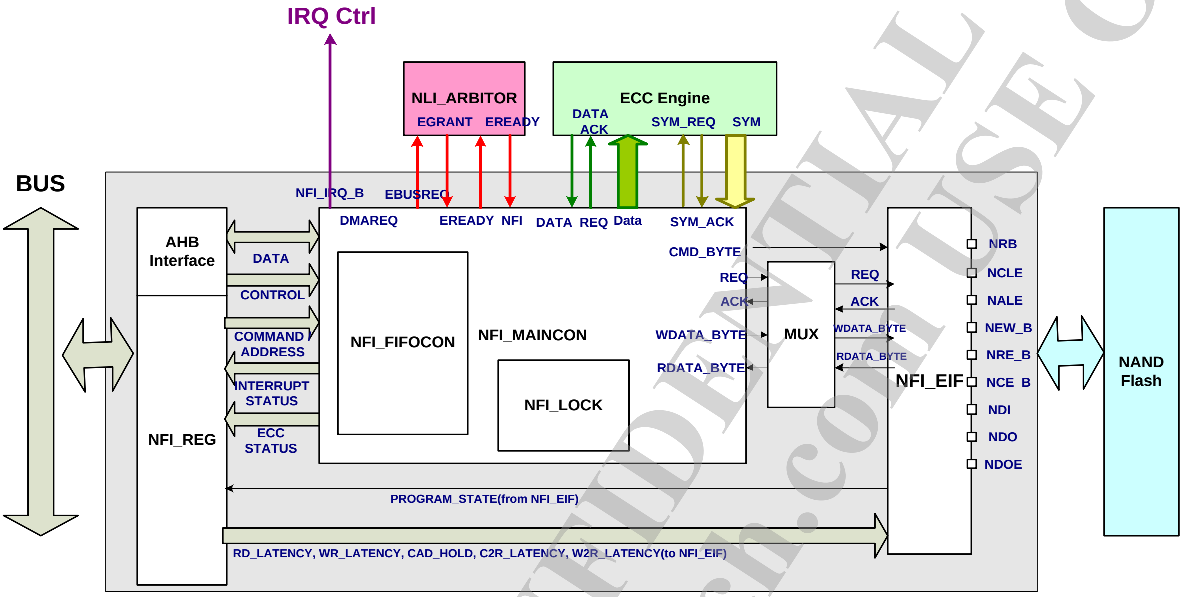


Figure 5-19. NFI Block Diagram

5.9.4 Register Definitions

For register details refer to chapter 3.9 in “MT8516A Application Processor Registers.”

5.10 Serial Flash Controller

5.10.1 Introduction

Serial flash controller is always a platform which can send different commands to flash to program or read. It has two different modes to read flash data: cpu direct access or flash dma. The system also can boot up from flash.

5.10.2 Features

- CPU access serial flash
- Check sum for serial flash read data
- System boot up from serial flash
- Supports 4byte address mode, compatible 3byte address mode.
- Supports 4bit output & 4bit I/O read mode, compatible single bit mode and dual bit mode
- Reads serial flash data through direct memory map, or dma path.

5.10.3 Block Diagram

The module has three buses:

- APB slave for serial flash controller register read/write.
- AXI slave for CPU direct access spi nor device through memory map
- AXI master for sflash dma moves data from spi nor to dram/sram.

flash_arb insures only one mode access spi nor device
sf_prefetch has a 32*32 sram to store prefetch data.
macro_sf_top has a delay-chain to adjust I/O bus skew.

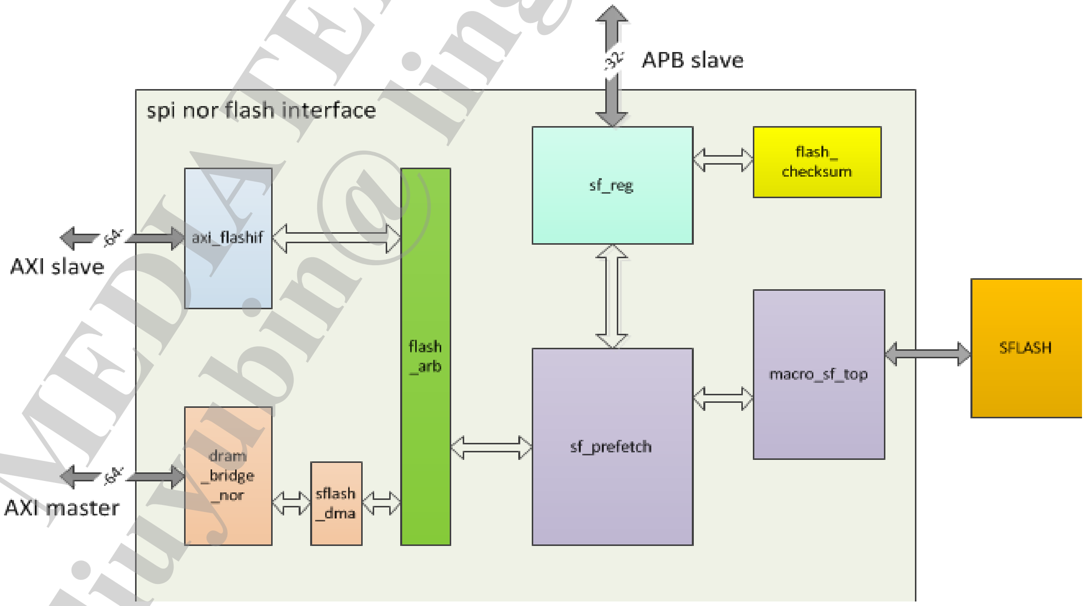


Figure 5-20.Flashif Block Diagram

5.10.4 Register Definitions

For detailed register information refer to chapter 3.10 in “MT8516A Application Processor Registers.”

5.10.5 Programming Guide

5.10.5.1 Program Serial Flash

The program command is used for programming the memory to be “0” . A Write Enable(WREN) command must execute the Write Enable Latch (WEL) bit before program process. The max program size one time of program command is a whole page size(256 byte for most device). There is allocated a 512byte prefetch buffer to handle an entire page program. In addition, it has program one byte one time using the controller registers. The program operation sequence is as below:

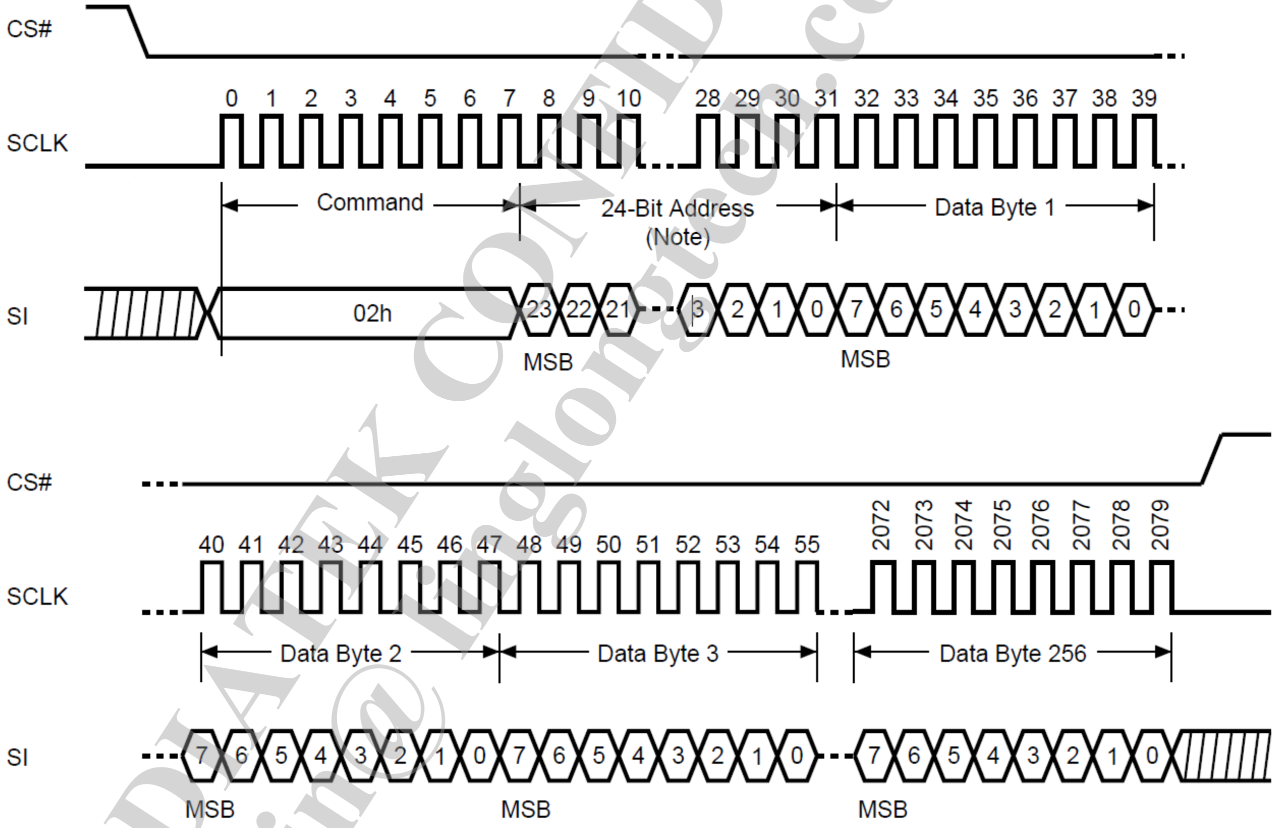


Figure 5-21. Program Operation Sequence

5.10.5.2 Entire Page Program Through Prefetch Buffer

- Write REG_SF_CFG2 0x1 to enable prefetch buffer for write.
- Write REG_SF_RADR2、REG_SF_RADR1 and REG_SF_RADR0 to set the program addrss
- Fill the Prefetch buffer by writing program data to REG_SF_PP_DW_DATA and loop 128 times .

- Write REG_SF_PAGECNT and REG_SF_PAGESIZE to set the page number and page size
- Write REG_SF_CMD bit4 1'b1 to trigger page program
- Write REG_SF_CMD bit1 1'b1 to send read flash status command
- Read REG_SF_RDSR bit0 to check page program done

5.10.5.3 Write Data To Serial Flash 1 Byte One Time

- Write program data to REG_SF_WDATA
- Write REG_SF_RADR2、REG_SF_RADR1 and REG_SF_RADR0 to set the program addrss
- Write REG_SF_CMD bit4 1'b1 to trigger page program
- Write REG_SF_CMD bit1 1'b1 to send read flash status command
- Read REG_SF_RDSR bit0 to check page program done

5.10.5.4 Read Flash

The controller read serial Nor flash via three ways: PIO read 、DMA read and Direct read mode. Standard read (SPI) operation sequence is as below:

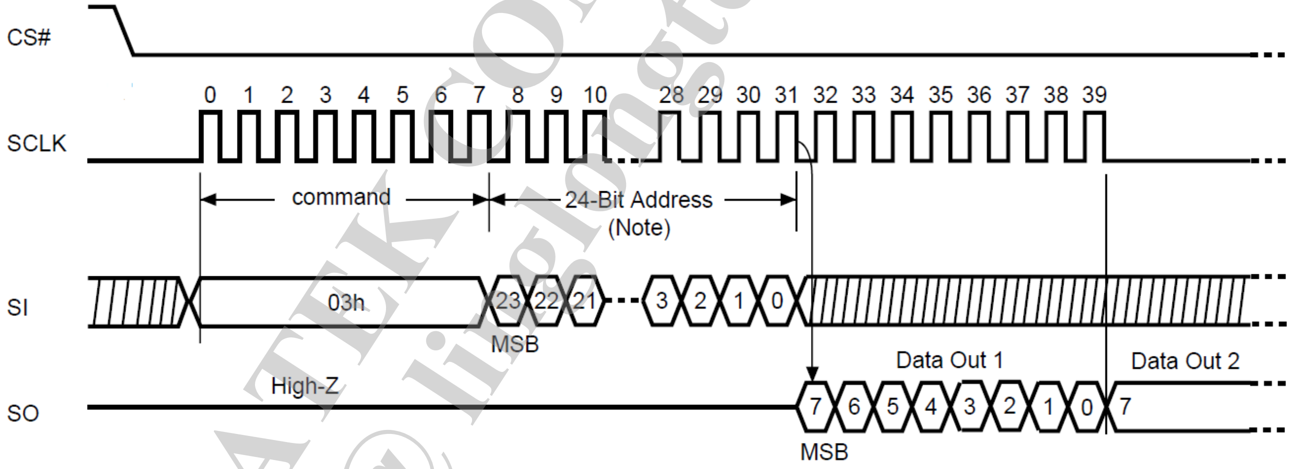


Figure 5-22. Read Operation Sequence

In 4byte address mode, the address cycles will be increased to 32 bit. Quad- IO and quad output read is also be supported, There will be 4 data pin at this case.

5.10.5.5 PIO Mode Read

The controller read the register REG_SF_RDATA to get the Nor flash data. Program flow is as below:

- Write REG_SF_RADR2、REG_SF_RADR1 and REG_SF_RADR0 to set the start read addrss, if at 4byte address mode, need write addr[31:24] to REG_SF_RADR3
- Write REG_SF_CMD bit7 and bit0 1'b1 to trigger read process
- Wait REG_SF_CMD bit0 tobe reset to 0

- Read REG_SF_RDATA to get the nor flash data

5.10.5.6 DMA Read

The controller support data from Nor flash to DRAM or SRAM via DMA mode. The DMA engine auto reads data from Nor flash device and write it to DRAM or SRAM, depend on the destination address manunal set. The source address destination start address and end address must be set before starting the DMA.

- Write REG_FDMA_FADR to set the DMA source address
- Write REG_FDMA_DADR to set the DMA destination start address
- Write REG_FDMA_END_DADR to set the DMA destination end address
- Write REG_FDMA_CTL bit0 1'b1 to trigger DMA start
- Wait REG_FDMA_CTL bit0 tobe 1'b0 to check DMA complete
- You can get the nor data from the destination space.

5.10.5.7 Direct Read

Direct read means data in Nor flash device could be directly read through address offset by CPU. For instance, the user would like to get the address of 0x100 data in Nor flash. User can capture data directly by read the contents of address offset 0x100 via direct read mode.

5.10.5.8 Enter 4bit Read Mode

The controller support 4bit SPI read mode to enhance the read performance. This is included the following two read sequence, the difference is whether the address is sent at 4bit mode:

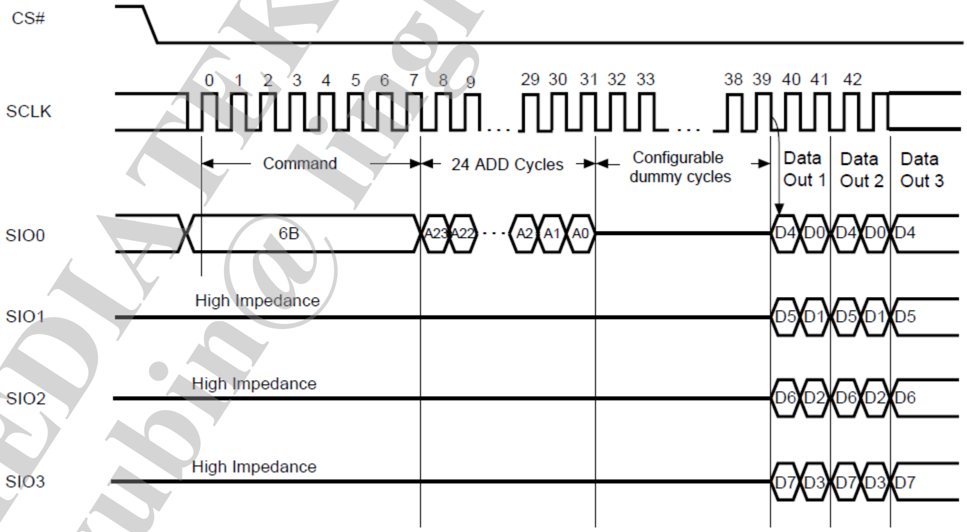


Figure 5-23. Quad Read Mode Sequence (Address Was Sent In Single Bit Mode)

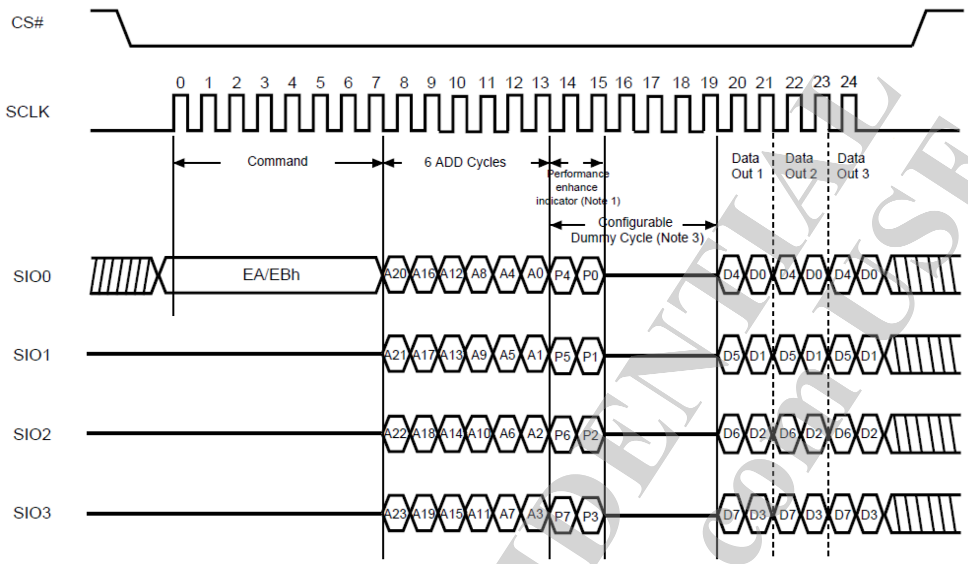


Figure 5-24. 4XIO Read Mode Sequence (Address Was Sent In 4bit Mode)

Enter the 4bit read mode, then the read operation will be at 4bit read mode. The program flow is as below:

- Write flash status register to enable SIO2&SIO3
- Write quad read or 4bit I/O read opration code to REG_SF_PRGDATA4
- Set add_quad and quad_read_en bit

5.10.5.9 Enter 4byte Address Mode

For most of Nor flash device, the default address mode is 3byte address mode. To access space larger than 16MB, you can set device to 4byte address mode

- Send EN4B command to set nor flash enter4 byte address mode
- Set 4byte_addr_en bit
- Wait REG_SF_DUAL bit4 tobe 1

5.10.5.10 Exit 4byte Address ModExit from 4byte address mode, please follow the steps below:

- Reset 4byte_addr_en bit
- Wait REG_SF_DUAL bit4 tobe 0
- Send EX4B command to set nor flash exit 4byte address mode

5.11 AUXADC

5.11.1 Introduction

The auxiliary ADC unit is used to identify the plugged peripheral and perform temperature measurement. There are 16 input channels allowing diverse applications, such as temperature measurement and light sensor.

Each channel only operates in the immediate mode. The time-trigger mode is now removed. In the immediate mode, the A/D converter 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 A/D converter 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 register AUXADC_DAT0, and the value for channel 1 is stored in register AUXADC_DAT1 and so on. If the AUTOSET(x) flag in the register AUXADC_CON0 is set, the auto-sampling function will be enabled in channel(x). The A/D converter samples the data for the channel(x) in which the corresponding data register is read. For example, in the case the AUTOSET0 flag is set. When the data register AUXADC_DAT0 is read, the A/D converter will sample the next value for channel 0 immediately.

If multiple channels are selected at the same time, the task will be performed sequentially on every selected channel from high to low channel. For example, if AUXADC_CON1 is set to 0x7f, i.e. all seven channels are selected, the state machine in the unit will start sampling from channel 6 to channel 0 and saves the values of each input channel in respective registers. The same process also applies to the timer-triggered mode.

The PUWAIT_EN bit in register AUXADC_CON3 is used to power up the analog port in advance, ensuring that the power is ramped up to the stable state before A/D converter starts the conversion. The analog part is automatically powered down after the conversion is completed.

Besides, there are several embedded temperature sensors. The module accepts signals from module of thermal controller to measure the temperature of the embedded sensors. The measurement result is able to be read in the command registers in the module of thermal controller.

5.11.2 Features

- Immediate analog-digital conversion
  In immediate mode, it supports auto-set option.
  In time-trigger mode, it supports auto-clear option.

- Background detection and interrupt
  Related command registers: AUXADC_DET_VOLT, AUXADC_PERIOD, AUXADC_DEBT, AUXADC_SEL

- Temperature measurement

5.11.3 AUXADC Block Diagram

Software controls the AUXADC through the APB bus. Once the hardware receives the command, it will trigger AUXADC channel sampling automatically. SW polls the status register or waits for interrupts from the CPU.

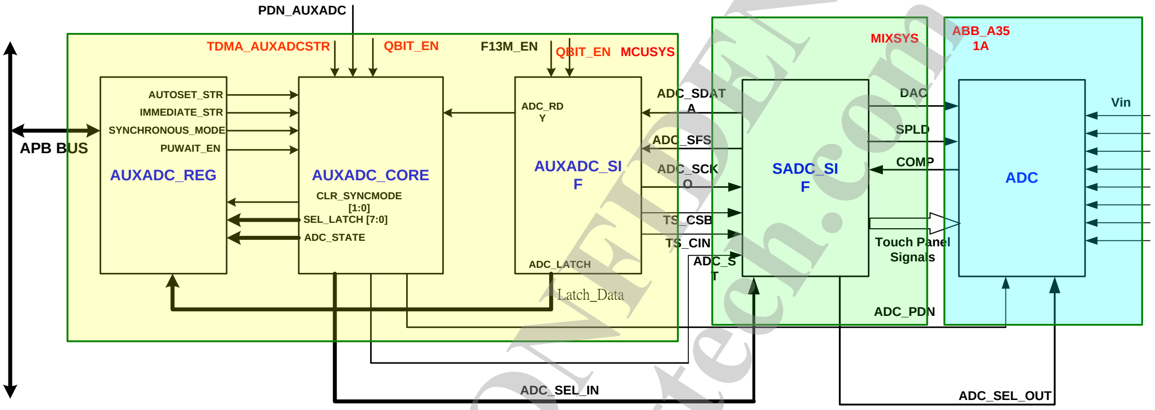


Figure 5-25. AUXADC Block Diagram

5.11.4 Theory of Operation

5.11.4.1 SAR ADC

Successive-approximation-register (SAR) ADC provides low power consumption, cost-effective and medium resolution. The AUXADC is SAR ADC architecture.

Here is an 8-bit conversion example. V_{REF} is the reference voltage of AUXADC.

The AUXADC implements a binary search algorithm. An initial register V_{DA} value compared to the input voltage V_{IN} is the mid-value between (2^8-1) and 0. 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. On the contrary, the MSB bit will be 0. Subsequently, bit 7 will be set to 1, and another comparison is done. Bit 6 to bit 0 will be executed as the previous action. Then, the 8-bit digital value will be available.

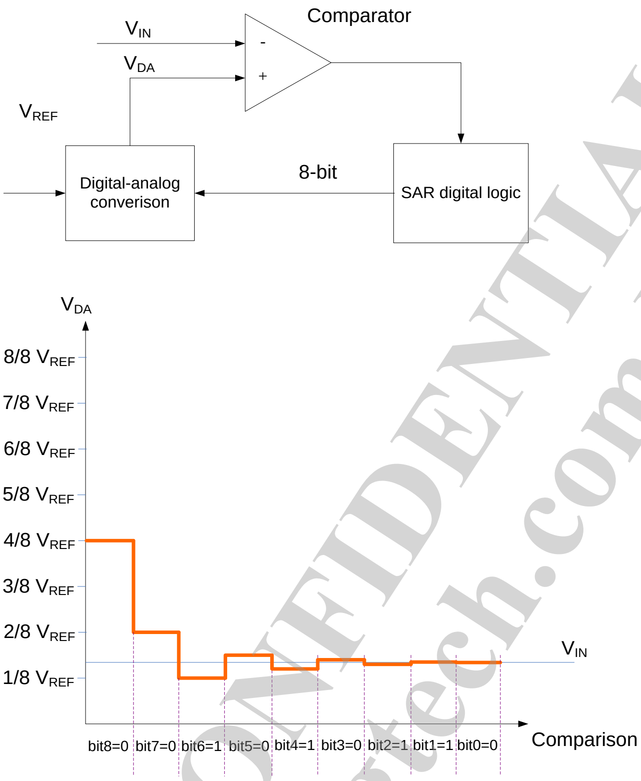


Figure 5-26. SAR ADC Architecture and Conversion

5.11.5 Design Partition

The table below shows the design partition.

Table 5-7. AUXADC Design Partition

Sub module name (hier1)	Description
AUXADC	Top module
AUXADC_REG	APB command registers
AUXADC_SIF	ADC serial interface with the module SADC_SIF
AUXADC_DEBUG	Debugging signal selection
AUXADC_MONITOR	Background detection and generate interrupt
AUXADC_CORE	AUXADC state machine and handle sampling sequence
SADC_SIF	Generate signals to analog part and transfer ADC result to the module AUXADC

5.11.6 Register Definitions

For register details refer to chapter 3.11 in “MT8516A Application Processor Registers.”

5.12 I2C/SCCB Controller

5.12.1 Introduction

I2C (Inter-IC)/SCCB (Serial Camera Control Bus) is a two-wire serial interface. The two signals are SCL and 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 specification.

5.12.2 Features

- I2C compliant master mode operation
- Adjustable clock speed for LS/FS mode operation
- Supports 7-bit/10-bit addressing
- Supports high-speed mode
- Supports slave clock extension
- START/STOP/REPEATED START condition
- 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
- Repeated start multiple transfer

5.12.3 Manual Transfer Mode

The controller offers manual mode.

When the manual mode is selected, in addition to the slave address register, the controller has a built-in 8-byte deep FIFO which allows 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.

5.12.4 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. Here are the transfer formats supported through different software configurations:

Wording convention note

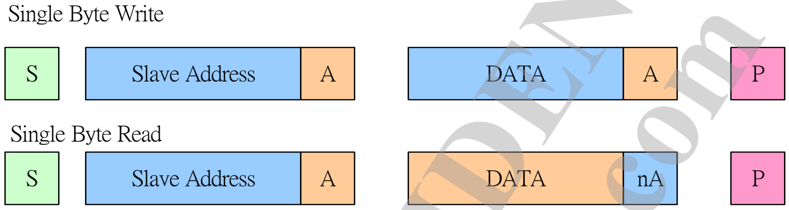
- Transfer = Anything encapsulated within a Start and Stop or Repeated Start.
- Transfer length = Number of bytes within the transfer
- Transaction = This is the top unit. Everything combined equals 1 transaction.

- Transaction length = Number of transfers to be conducted.

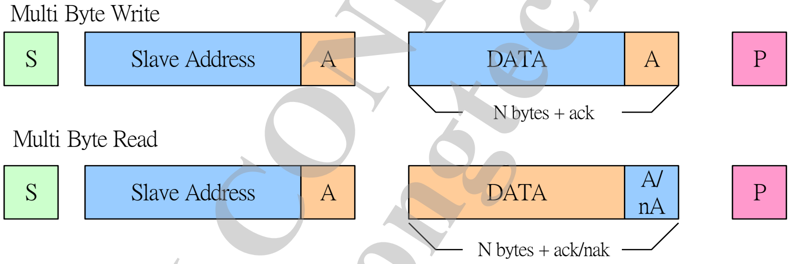
🟦 Master to slave dir

🟧 Slave to master dir

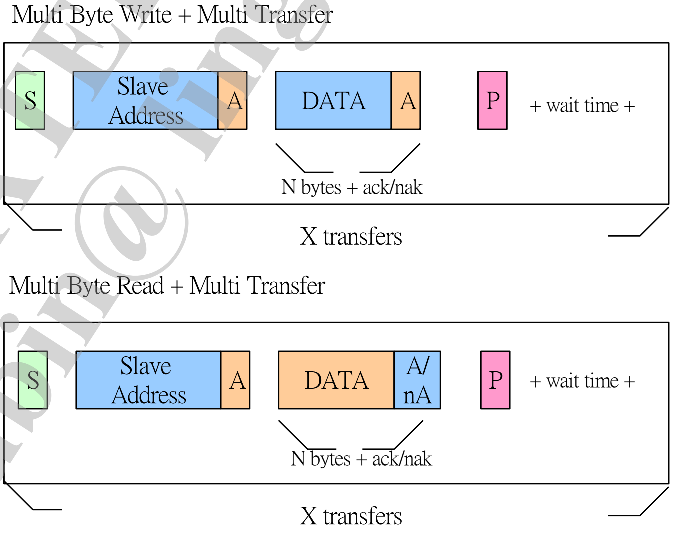
Single byte access



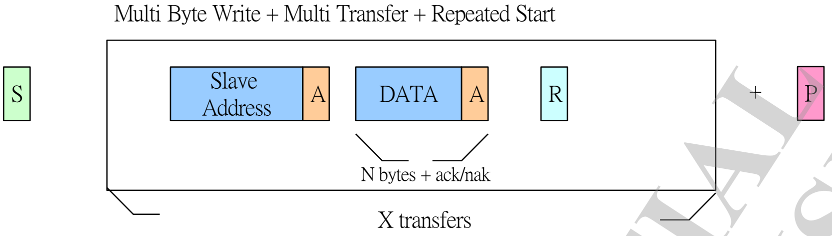
Multi byte access

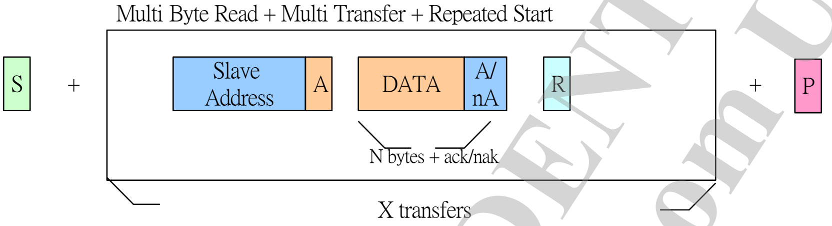


Multi-byte transfer + multi-transfer (same direction)



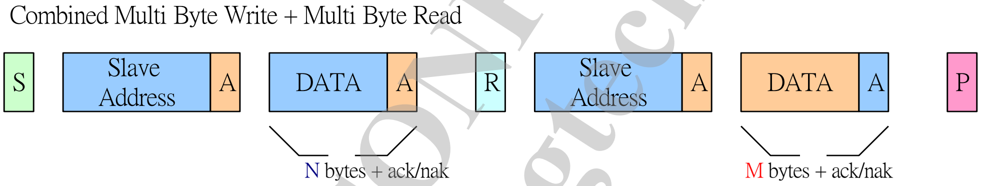
Multi-byte transfer + multi-transfer w RS (same direction)



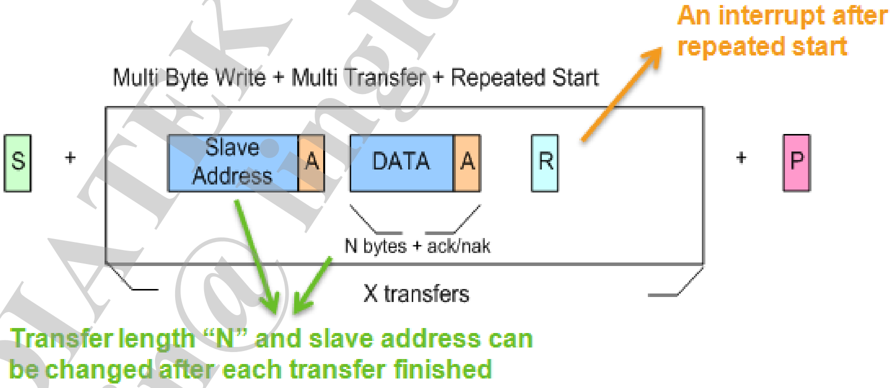


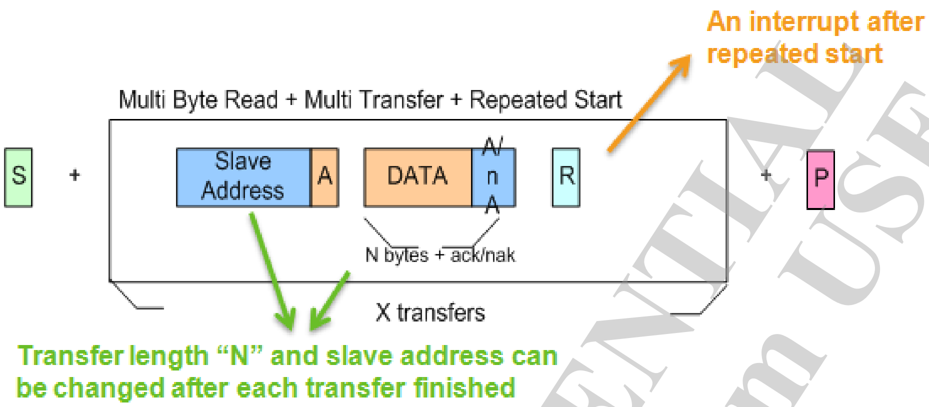
Combined write/read with Repeated Start (direction change)

Note: Only supports write and then read sequence. Read and then write is not supported.



Repeated start multiple transfer (write/read)


An interrupt after repeated start
Multi Byte Write + Multi Transfer + Repeated Start
S
+
Slave Address
A
DATA
A
R
+
P
N bytes + ack/nak
X transfers
Transfer length "N" and slave address can be changed after each transfer finished


Multi Byte Read + Multi Transfer + Repeated Start
S
+
Slave Address
A
DATA
A/n A
R
+
P
N bytes + ack/nak
X transfers
An interrupt after repeated start
Transfer length "N" and slave address can be changed after each transfer finished

5.12.5 I2C Block Diagram

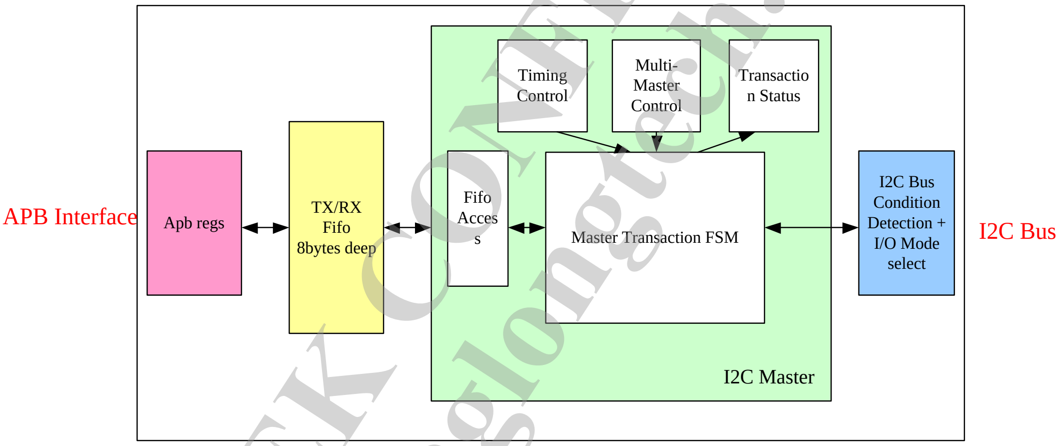


Figure 5-27. I2C Block Diagram

5.12.6 Register Definitions

I2C number	Base address	Feature
I2C0	0x11009000	Supports DMA
I2C1	0x1100A000	Supports DMA
I2C2	0x1100B000	Supports DMA

There are three I2C IPs in this SOC. The usage of the registers is the same except that the base address must be changed to respective one.

For register details refer to chapter 3.12 in "MT8516A Application Processor Registers."

5.13 Pulse-Width Modulation (PWM)

5.13.1 Introduction

Three generic pulse-width modulators are implemented to generate pulse sequences with programmable frequency and duration for LCD backlight, charging or other purposes. Before enabling 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 (refer to figure below).

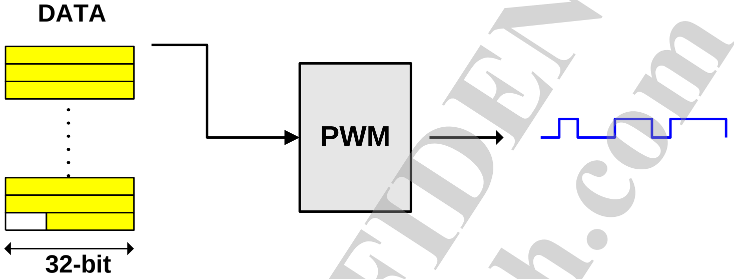


Figure 5-28. PWM Generation Procedure

5.13.2 Features

- Old mode, FIFO mode
- Periodical memory and random mode
- Sequential output mode and 3DLCM mode

5.13.3 PWM Block Diagram

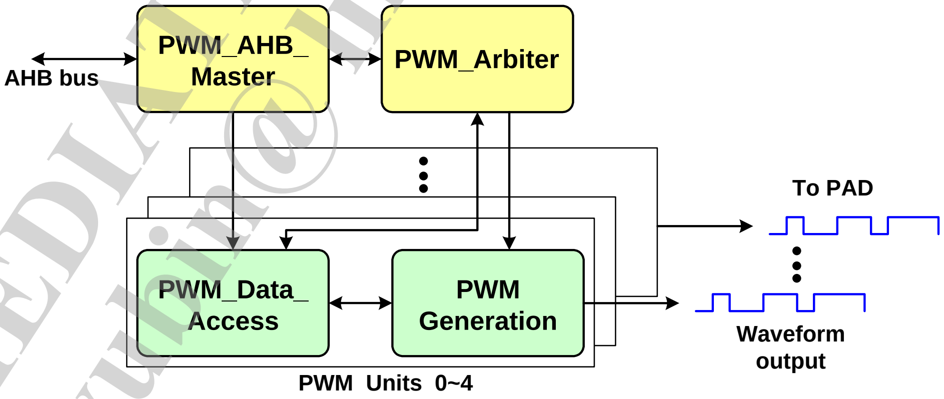


Figure 5-29. PWM Block Diagram

5.13.4 Register Definitions

For register details refer to chapter 3.13 in “MT8516A Application Processor Registers.”

5.14 System Timer

5.14.1 Introduction

The 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 passing to APMCU, SCP, and other micro-processors to provide unify system timestamp between OSs (Android Linux, RTOS …etc).

5.14.2 Features

The system timer includes following functions:

- A 64-bit, always-on up-counter (this counter is default enabled ticking with 13MHz clock period)
- Clock divider to allow timer ticking with 26MHz/13MHz/6.5MHz clock period
- HW counter increment offset compensation when switching to 32KHz 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)

5.14.3 Block Diagram

Below is the sys_timer block diagram.

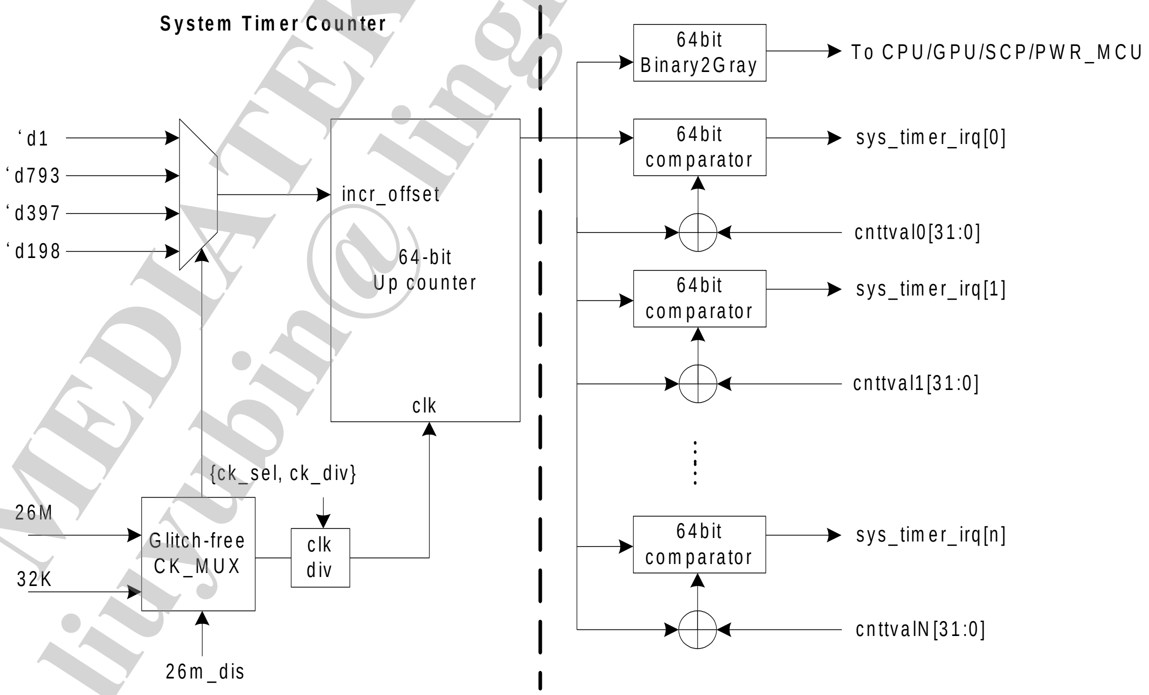


Figure 5-30. sys_timer Block Diagram

5.14.4 Programming Guide

The sys_timer consists of one 64-bit up counter, glitch-free clock mux, clock divider, and multiple 64-bit comparators. The 64-bit up counter is default enabled and will start ticking with 13MHz clock period after reset released. The timer tick can be programmed as 26MHz, 13MHz, or 6.5MHz, and can be switched to 32KHz by power manager when 26MHz clock source is unavailable.

In 32KHz mode, the counter increment offset values changes 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, 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 covert the counter value back at receiving side. Except exporting 64-bit counter value to different sub-system, the sys_timer also provides multiple comparators which allow programmer to setup 32-bit counter timeout values that can trigger interrupts after timeout.

When programming, write a 32-bit offset value into the CNTTVAL[n] register, the 32-bit offset value is added to current 64-bit counter as expected timeout value. The behavior of timer is shown in the figure below. Reading CNTTVAL[n] represents the difference between expected timeout value and current 64-bit counter value. Therefore, the CNTTVAL[n] can be seen as a 32-bit down-counter (with 64-bit up-counter counting) which triggers sys_timer_irq[n] when CNTTVAL reaches zero.

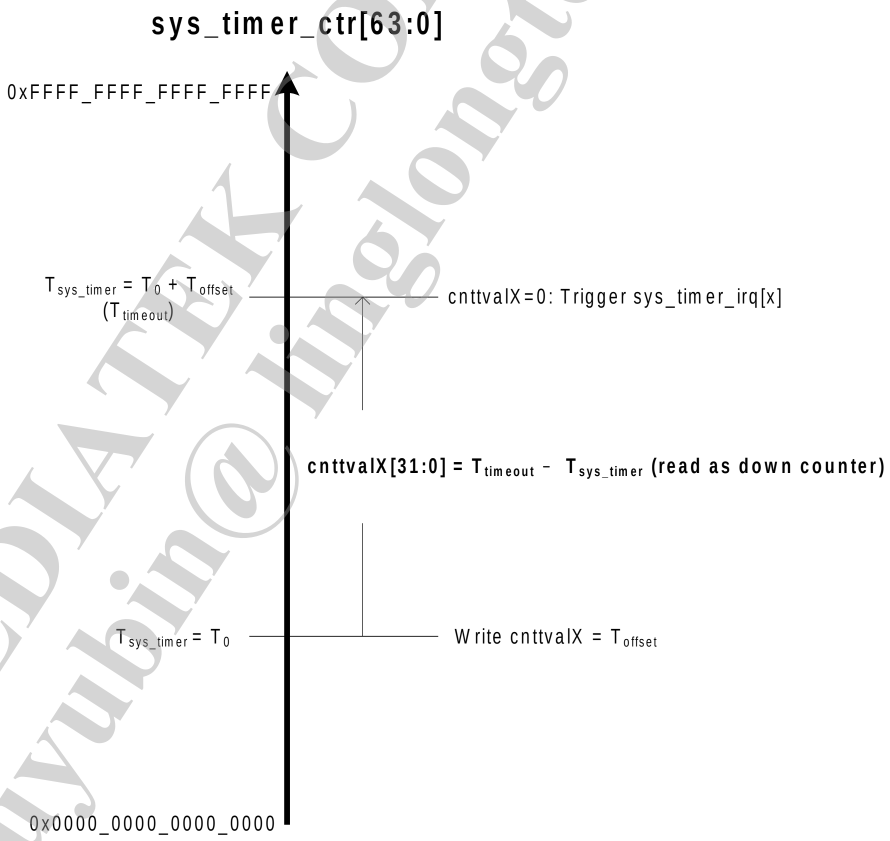

sys_timer_ctr[63:0]
0xFFFF_FFFF_FFFF_FFFF
T_{sys_timer} = T_0 + T_{offset}
($T_{timeout}$)
cnttvalX=0: Trigger sys_timer_irq[x]
cnttvalX[31:0] = $T_{timeout}$ - T_{sys_timer} (read as down counter)
T_{sys_timer} = T_0
Write cnttvalX = T_{offset}
0x0000_0000_0000_0000

Figure 5-31. Behavior of sys_timer Counter Timeout Value

The security access control is controlled by three registers:

- CNTWACR (write access control)
- CNTRACR (read access control)
- CNTACR_LOCK (program lock bit)

The three registers can only be programmed via secure access, which means they can only be configured in 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 filed definition is described in register definition table). CNTACR_LOCK is a one-time program register which can lock the CNTWACR and CNTRACR after configuration is fixed.

5.14.5 Register Definitions

For register details refer to chapter 3.14 in “MT8516A Application Processor Registers.”

5.15 General-Purpose Timer (GPT)

5.15.1 Introduction

The GPT includes five 32-bit timers and one 64-bit timer. Each timer has four operation modes, which are ONE-SHOT, REPEAT, KEEP-GO and FREERUN, and can operate on one of the two clock sources, RTC clock (32.768kHz) and system clock (13MHz).

5.15.2 Features

The four operation modes for GPT are ONE-SHOT, REPEAT, KEEP-GO and FREERUN. See Table 5-8 for the functions of each mode.

Table 5-8 GPT Operation Mode

Mode	Auto Stop	Interrupt	Increases when EN=1 and ...	When COUNTn equals COMPAREn	Example: Compare is set to 2 **Bold means interrupt*
ONE-SHOT	Yes	Yes	Stops when COUNTn equals to COMPAREn	EN is reset to 0.	0,1,**2**,2,2,2,2,2,2,2,2,2,...
REPEAT	No	Yes		Count is reset to 0.	0,1,**2**,0,1,**2**,0,1,**2**,0,1,**2**...
KEEP-GO	No	Yes	Reset to 0 when overflow		0,1,**2**,3,4,5,6,7,8,9,10,...
FREERUN	No	No	Reset to 0 when overflow		0,1,2,3,4,5,6,7,8,9,10,...

Each timer can be programmed to select the clock source, RTC clock (32.76kHz) or system clock (13MHz). 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, 2, 3, 4 to 13 and coarse-granulated as 16, 32 and 64.

5.15.3 GPT Block Diagram

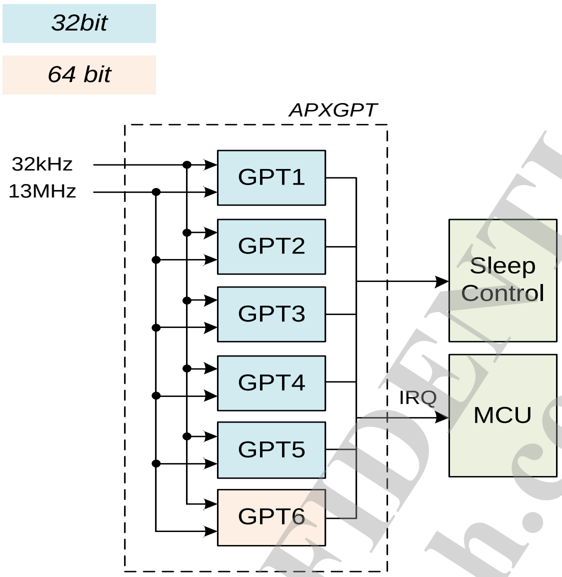


Figure 5-32. APXGPT Block Diagram

5.15.4 Register Definitions

For register details refer to chapter 3.15 in the “MT8516A Table Application Processor Registers.”

5.16 Thermal Controller

5.16.1 Introduction

In the audio platform, thermal management is very fundamental. The thermal management controls the platform computing performance to achieve the requirement and maintain the IC within the temperature constraints. Operation in temperature over for a long time will have a risk of damage for IC reliability.

In MT8516A , 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 the software effort of temperature monitoring, the thermal controller will generate interrupts informing the CPU of abnormal condition.

5.16.2 Features

- Support up to three thermal sensors
- Periodic temperature measurement
- Temperature monitoring
- Different type of low pass filter for thermal sensor readings

5.16.3 Thermal Controller Block Diagram

There are microprocessors in the MT8516A and their max frequency is over G Hz and the transistor count is also large. The microprocessors consume a high percentage of whole chip power consumption. In addition to microprocessor power consumption, EMI is also a source of power consumption because it provides high DRAM data bandwidth to other modules in MT8516A .

The MT8516A Thermal controller is implemented for software to operate with a pre-defined temperature range so as to avoid function failure and reliability issues. According to the temperature measurement, the system performance can be adjusted. And system design for power dissipation can be also monitored.

The hottest location in the MT8516A may be different in different scenarios. When the thermal controller informs the software of an abnormal condition, the following power reduction action should be an efficient and low latency.

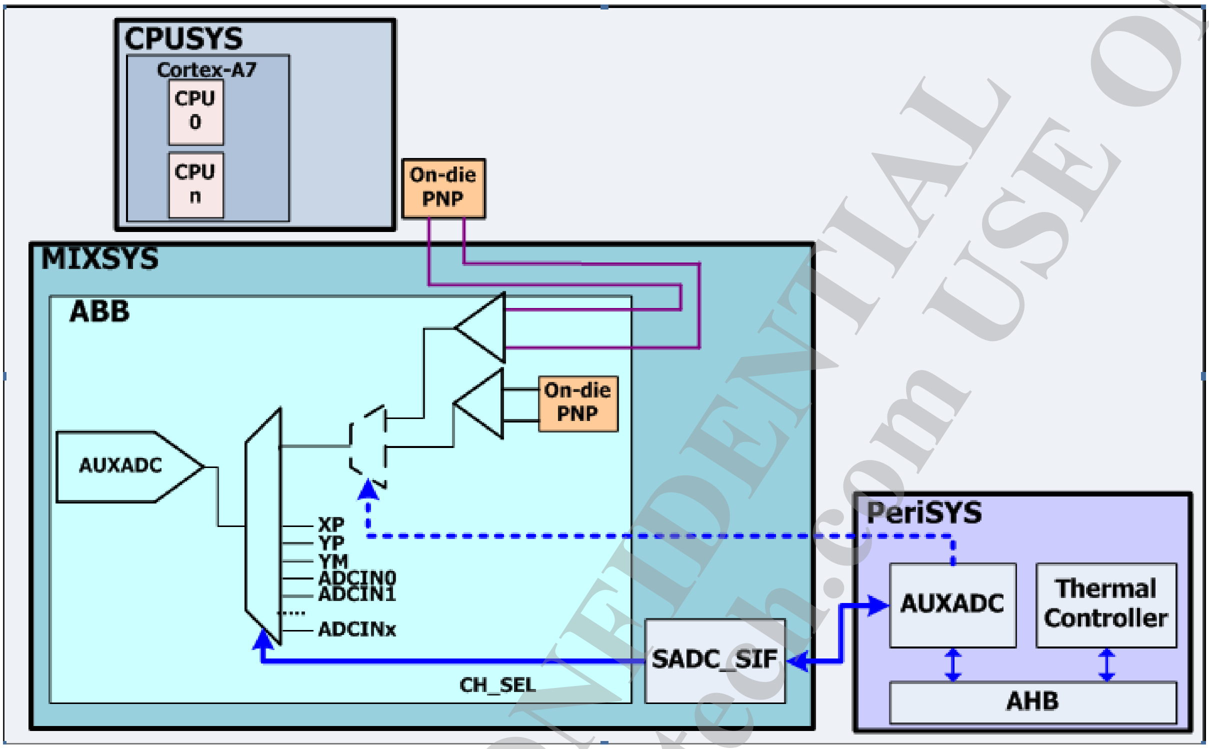


Figure 5-33. Implementation of CMOS Temperature Sensor

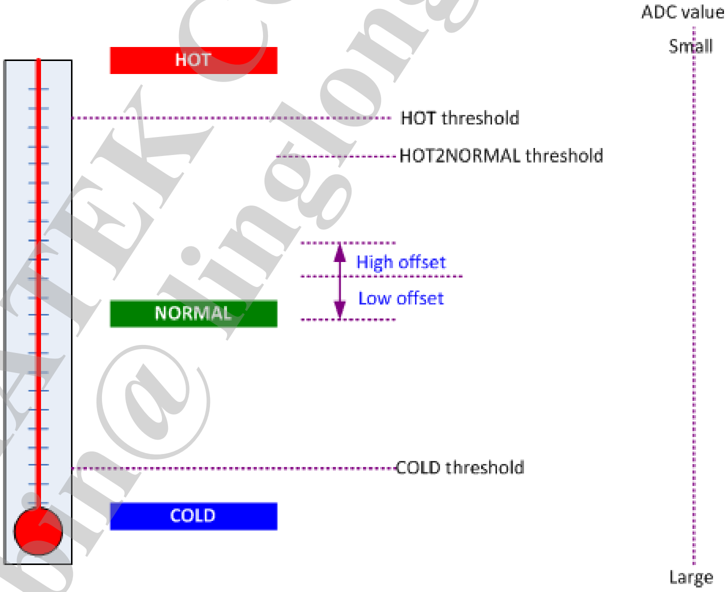


Figure 5-34. System Temperature Measurement Block Diagram

5.16.4 Register Definitions

See Chapter 3.16 of "MT8516A Application Processor Registers."

5.16.5 Programming Guide

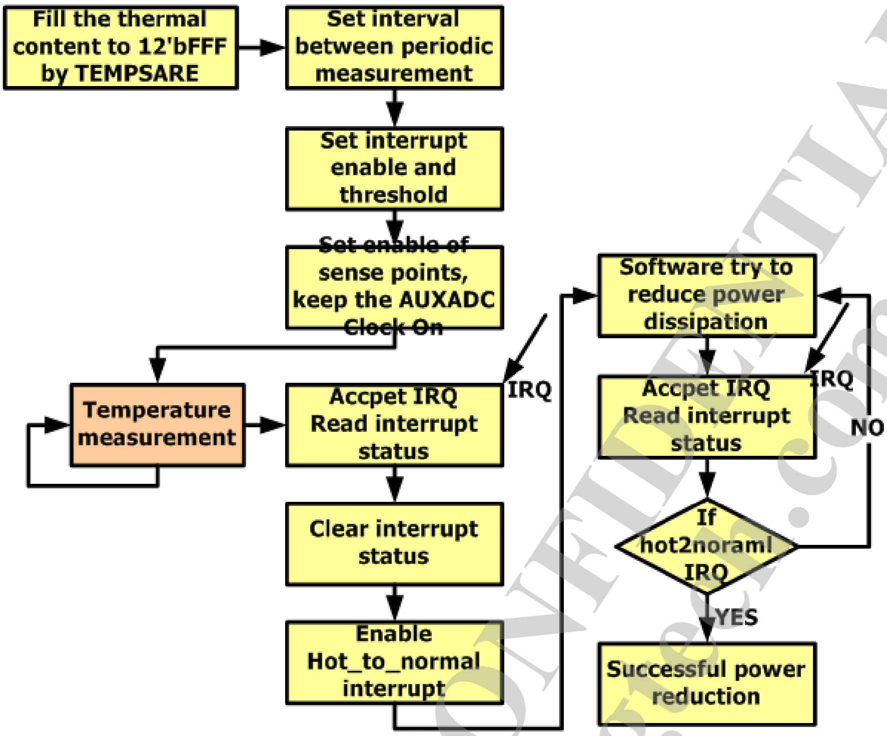

Fill the thermal content to 12'bFFF by TEMPSARE
Set interval between periodic measurement
Set interrupt enable and threshold
Set enable of sense points, keep the AUXADC Clock On
Temperature measurement
Accpet IRQ Read interrupt status
IRQ
Clear interrupt status
Enable Hot_to_normal interrupt
Software try to reduce power dissipation
Accpet IRQ Read interrupt status
IRQ
NO
If hot2noraml IRQ
YES
Successful power reduction

Figure 5-35. Programming Flow

Fill the thermal content to 12'bFFF by access TEMPSPARE
vWriteREG(TEMPMONCTL1, 'h0);
vWriteREG(TEMPMONCTL2, 'h0);
vWriteREG(TEMPAHBPOLL, 'h0); // polling interval to check if temp sense is ready
vWriteREG(TEMPAHBTO, 'hFF); // Exceed this polling time, IRQ would be inserted
vWriteREG(TEMPSPARE0, 'h1FFF); // fill the TEMPSPARE0 register as 'h1FFF
vWriteREG(TEMPPNPMUXADDR, 32'hTS_CON1); // The adxadc mux address to select to Thermal channel, and please reference mixsys.doc
vWriteREG(TEMPADCENADDR, 32'TEMPSPARE1); // The adxadc enable address to trigger Thermal senser, please reference auxadc.doc
vWriteREG(TEMPADCVALIDADDR, 32'hTEMPSPARE1); // The adxadc status address to check if Thermal senser reading is valid, please reference auxadc.doc
vWriteREG(TEMPADCVOLTADDR, 32'hTEMPSPARE0); // The adxadc temperature address for the value read back from temp senser, please reference auxadc.doc
vWriteREG(TEMPRDCTRL, 'h0); // use TEMPSPARE0 as valid address
vWriteREG(TEMPADCVALIDMASK, 'h2c); // set adxadc valid polarity to 0
vWriteREG(TEMPMONCTL0, 'h0F); // enable all sense points include the debug one
Wait until the content of TEMPIMMD are filled by 'hFFF

Set interval between periodic temperature measurement, if the MODULE clock is 66MHz
vWriteREG(TEMPMONCTL1, 'h3FF); // counting unit is 1024*15.15ns=15.5 us

```
vWriteREG(TEMPMONCTL2, '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); // Exceed this polling time, IRQ would be inserted
vWriteREG(TEMPADCPNP0, 32'h0); // The adxadc PNP data to select to Thermal Sensor, and please reference mixsys.doc
vWriteREG(TEMPADCPNP1, 32'h1); // The adxadc PNP data to select to Thermal Sensor, and please reference mixsys.doc
vWriteREG(TEMPADCPNP2, 32'h2); // The adxadc PNP data to select to Thermal Sensor, and please reference mixsys.doc
vWriteREG(TEMPADCEN, 32'h800); // The adxadc enable data(CH-11) to trigger Thermal senser, please reference auxadc.doc
vWriteREG(TEMPADCMUX, 32'h800); // The adxadc enable data (CH-11) to clear trigger Thermal senser, please reference auxadc.doc
vWriteREG(TEMPPNPMUXADDR, 32'hTS_CON1); // The adxadc mux address to select to Thermal channel, and please reference mixsys.doc
vWriteREG(TEMPADCENADDR, 32'hAUXADC_CON1_SET); // The adxadc enable address to trigger Thermal senser, please reference auxadc.doc
vWriteREG(TEMPADCMUXADDR, 32'hAUXADC_CON1_CLR); // The adxadc enable address to trigger Thermal senser, please reference auxadc.doc
vWriteREG(TEMPADCVALIDADDR, 32'hAUXADC_CON3); // The adxadc status address to check if Thermal senser reading is valid, please reference auxadc.doc
vWriteREG(TEMPADCVOLTADDR, 32'hAUXADC_DAT11); // The adxadc temperature address for the value read back from temp senser, please reference auxadc.doc
vWriteREG(TEMPRDCTRL, 'h0); // use AUXADC_DAT11 as valid address
vWriteREG(TEMPADCVALIDMASK, 'h2c); // set adxadc valid polarity to 0
vWriteREG(TEMPWRITECTRL, 'h3); // Enable Write transaction type
```

Set monitoring threshold and SPM wake up event

```
vWriteREG(TEMPHTHRE, 'hxxx); // set hot threshold
vWriteREG(TEMPCTHRE, 'hxxx); // set cold threshold
vWriteREG(TEMPCTHRE, 'hxxx); // set hot to normal threshold
vWriteREG(TEMPPROTCTL, 'h20xxx); // set hot to wakeup event control
vWriteREG(TEMPPROTTC, 'hxxx); // set hot to HOT wakeup event
vWriteREG(TEMPMONINT, 'h8000001F); // enable interrupt
```

Set sense points enable

```
vWriteREG(TEMPMONCTL0, 'h07); // enable all three sense points
```

Accept IRQ

```
vReadREG(TEMPMONINTSTS); // read interrupt and clear interrupt status
```

Read temperature readings (optional)

```
vReadREG(TEMPMSR0); // read temperature reading of sense point 0
vReadREG(TEMPMSR1); // read tempe
```

rature reading of sense point 1
vReadREG(TEMPMSR2); // read temperature reading of sense point 1

Release pause of periodic temperature measurement
vWriteREG(TEMPMSRCTL1, vReadREG(TEMPMSRCTL1) & 0xFFFE);

5.16.6 Immediate Temperature Measurement

After each immediate done, software must disable immediate mode.

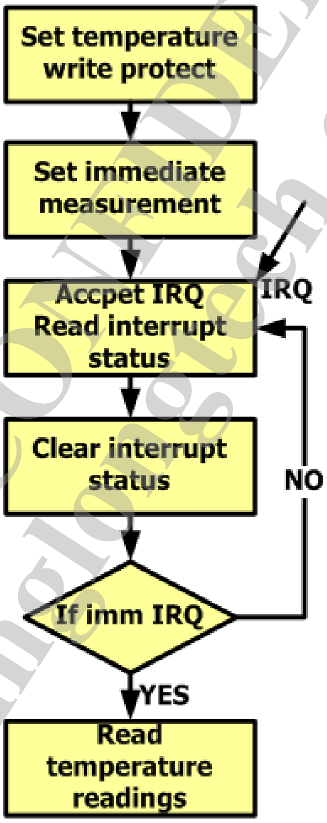

Set temperature write protect
Set immediate measurement
IRQ
Accpet IRQ Read interrupt status
Clear interrupt status
NO
If imm IRQ
YES
Read temperature readings

Figure 5-36. Immediate Measurement Programming Flow

5.16.7 Hardware Interrupt

The Interrupt condition of high and low temperature monitoring is shown below.
Software will accept interrupts while the following three conditions occur. Software can determine which temperature sensor will be monitored. Once the condition in any one of the three temperature sensors occurs, the interrupt will be issued.
The state machine is shown in Figure 5-38 below.

- Cold interrupt: When the temperature decreases to be lower than the cold threshold form the normal temperature range. It means that when the state of NORMAL transferred into the state of COLD.
- Hot interrupt: When the temperature increases to be higher than the hot threshold from the temperature below the hot threshold.
- The state of VERY_HOT indicates that the temperature is higher than the hot threshold. The state of HOT1 indicates that the temperature is higher than the hot_to_normal threshold. NORMAL state cannot transfer into VERY_HOT directly.

Hot to normal temperature interrupt: When HOT2 state transferred into NORMAL state.

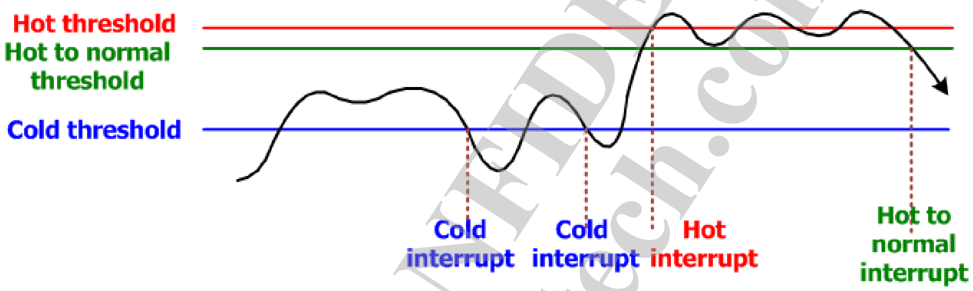


Figure 5-37. Interrupt Condition of High/Low Temperature Monitoring

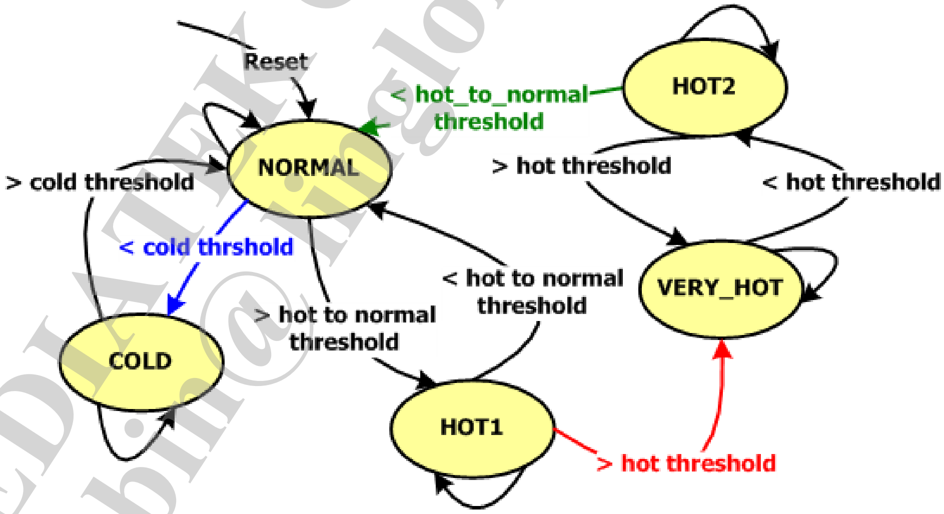


Figure 5-38. Finite State Machine of High/Low Temperature Monitoring

In Figure 5-39 when software immediate measurement is enabled, the state will maintain the current state until software disable immediate mode. It is shown as the "*" mark.

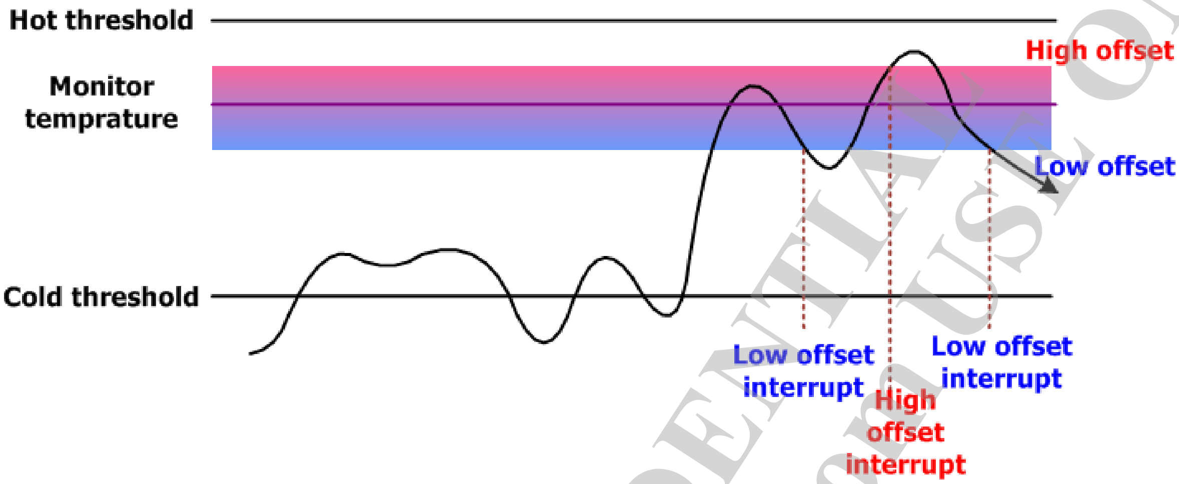


Figure 5-39. Interrupt Condition of High/Low Offset Monitoring

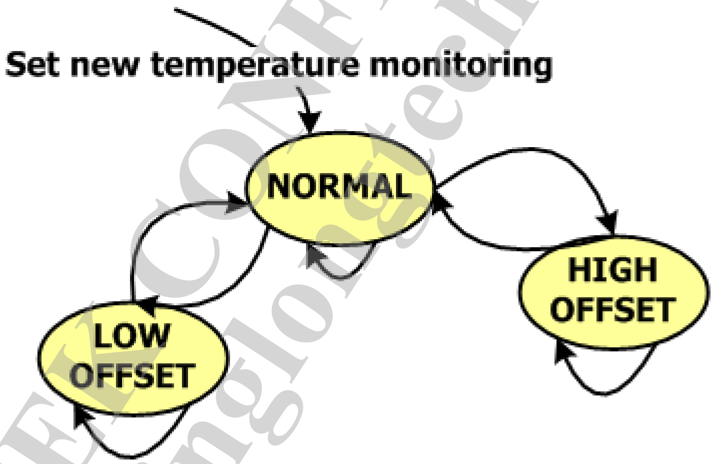


Figure 5-40. Finite State Machine of High/Low Offset Monitoring

5.17 Infrared-Receiver (IRRX)

The IRRX module can receive the Infra-Red signal and can support NEC protocol, RC5 protocol, RC6 protocol.

5.17.1 Introduction

This IR receiver can decode various IR transmission protocols. They could be divided into two groups. One is pulse-width coding such as NEC IR transmission protocol; the other is bi-phase coding, for example, RC5, RC6, RCMM. Figure 5-41 is an example for pulse-width coding. We can decode the signal by the length of pulse width. Figure 5-42 is an example for bi-phase coding. We can decode the signal by a constant period sampling pulse.

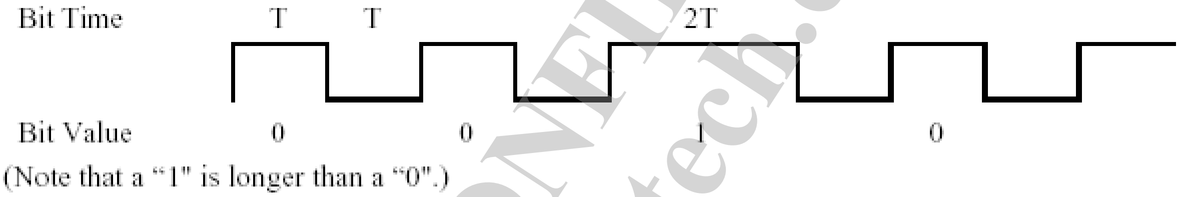


Figure 5-41. **Pulse-width Coding**

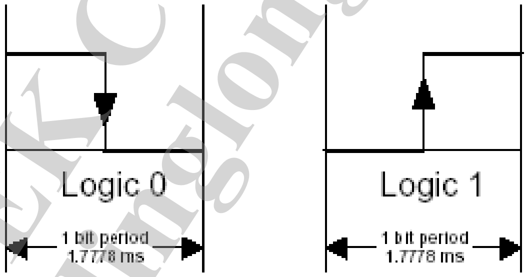


Figure 5-42. **Bi-phase Coding**

5.17.2 IRRX Block Diagram

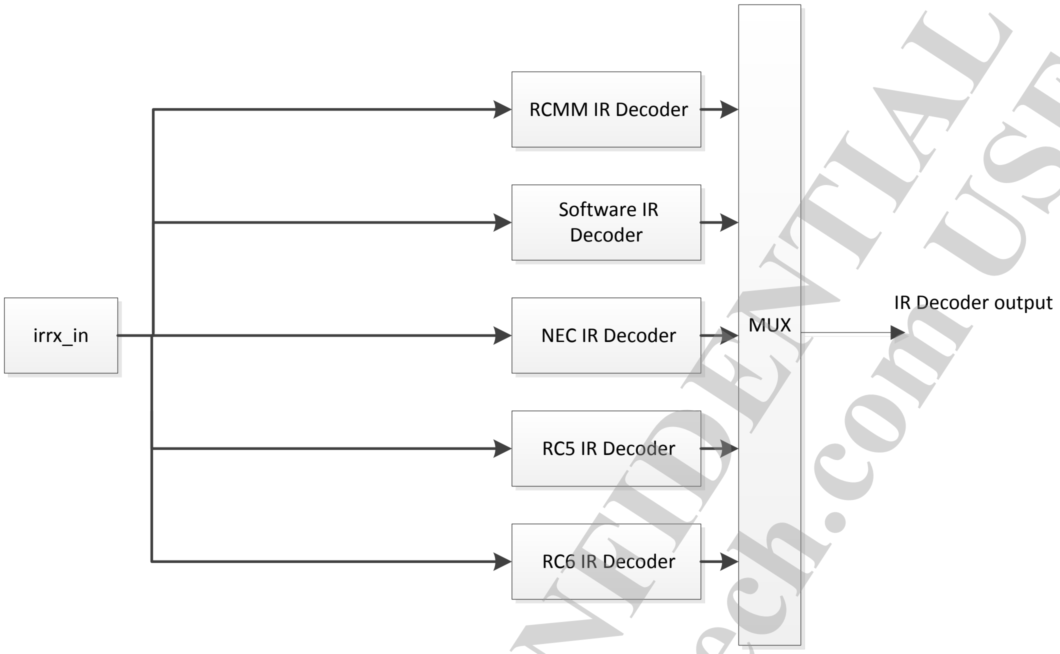


Figure 5-43. Infrared-Receiver Block Diagram

5.17.3 Register Definitions

For register details refer to chapter 3.17 in “MT8516A Application Processor Registers.”

5.18 Ethernet NIC

5.18.1 Introduction

IP201, a 10/100 MAC controller, is compliant with 802.3 standards. It supports power management with Energy Efficient Ethernet and Wake-on-LAN specification. Flow control is provided for half-duplex and full-duplex mode. For packet transmission and reception, it supports IPv4/UDP/TCP checksum offload and VLAN tag insertion.

5.18.2 Features

- Dynamically configurable to support 10/100M with MII/RMII
- EEE (Energy Efficient Ethernet) MII signaling according to the IEEE802.3az specification
- CRC-32 checking with optional forwarding of the FCS field to the user application
- CRC-32 generation and append on transmit or forwarding of user application provided FCS
- Optional MAC address comparison on receive and overwrite on transmit with programmable promiscuous mode operation
- Optional multicast address filtering with 512-bin hash code lookup table on receive
- Operational Ethernet Pause Frame generation from FIFO congestion thresholds
- Optional Ethernet Pause Frame (802.3 Annex 31A) termination providing fully automated flow control without any user application overhead
- Optional Magic packet detection
- Support for VLAN tagged frames according to IEEE 802.1Q specification in both transmit and receive
- Support TX/RX IPv4/UDP/TCP checksum offload
- Clause 22 and Clause 45 MDIO master interface for PHY device configuration and management

5.18.3 Block Diagram

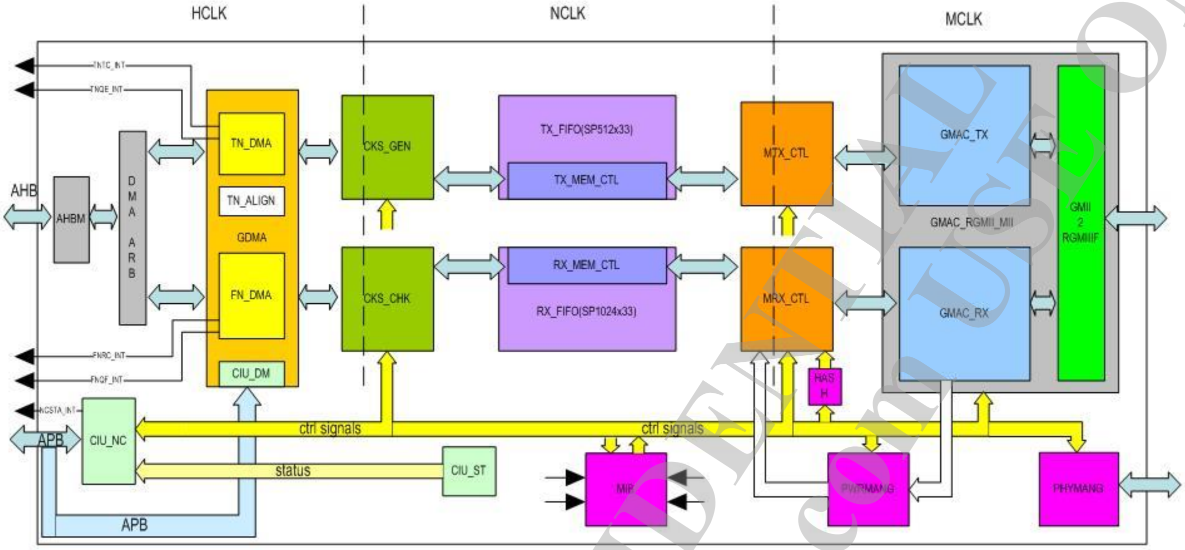


Figure 5-44. Ethernet MAC Top Block Diagram

5.18.4 Theory of Operations

5.18.4.1 MAC Receive

5.18.4.1.1 Overview

The MAC receive engine performs the following tasks:

- Check Frame Framing
- Remove Frame preamble and Frame SFD field
- Terminate Pause Frames
- Support 16 (15 of them are optional) configurable DA to check received DA. If not match, the packet will be dropped.
- Drop processing of oversize frame and short frame.
- Calculate and verify CRC-32
- Write received Frames in the Core receive FIFO
- IP and TCP/UDP checksum
- DMA to write packet data from receive FIFO to external memory.
- DMA interface transfer to AHB

5.18.4.1.2 Preamble Processing

MAC Core checks for the start frame delimiter (SFD) byte. Before SFD, 0~7 bytes preamble is acceptable.

The following shows cases of no preamble and odd preamble.

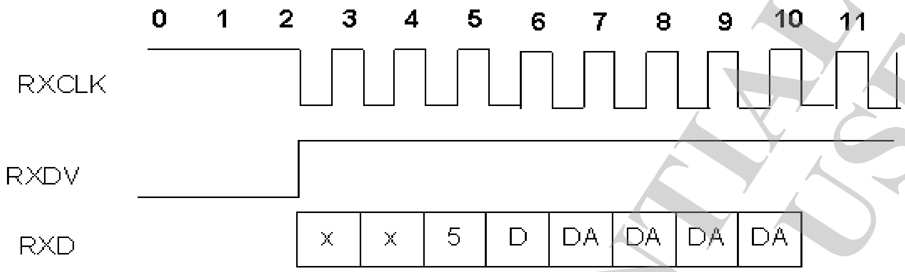


Figure 5-45. Waveform in No Preamble Case

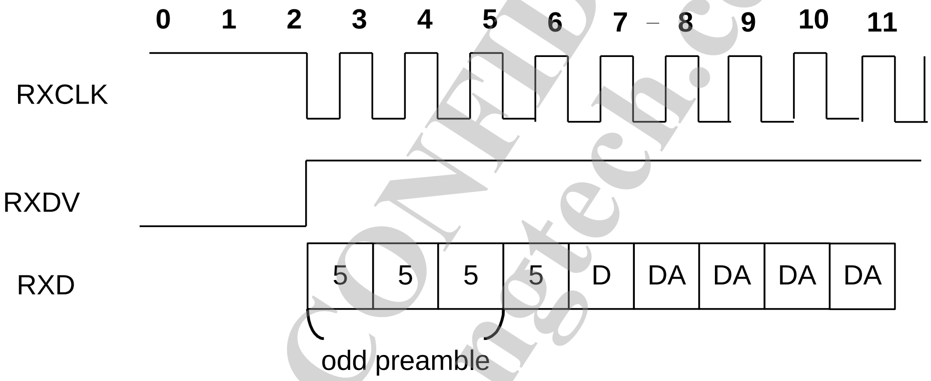


Figure 5-46. Waveform in Odd Preamble Case

5.18.4.1.3 Frame Length / Type Verification

The NIC does not check the correction of length field. An internal counter is used to calculate frame length. If calculated length < 64 byte, then drop as a runt packet. If calculated length > 1518 (or 1522, 1536, by configuration), then assert oversize indication in frame information and this packet will be optionally dropped in DMA.

Control and VLAN frames (Frame Length / Type field 0x8808 and 0x8100 respectively) are processed by the Core as described in the two following sections.

5.18.4.1.4 VLAN frame Processing

The NIC supports 802.1q tag-based VLAN ingress check, and it can support up to 4 VLANs, set in registers, where these VLAN IDs can be any in 4K VLAN space. Internally, the controller uses 4 bits of "My VLAN ID Control Register" to enable VLAN ingress check for each pre-defined VLAN ID.

When at least one of the pre-defined VLAN ID is enabled, RX MAC will compare the pre-defined VLAN ID with the tagged VID of the received packet. If one of them is matched, the packet will be received, otherwise, it will be dropped, and the relevant MIB counter will be increased by 1 accordingly. Please note that VLAN ingress check has no effect on non VALN tagged packet. When a received packet is VLAN-tagged, the tag can be stripped from the packet or retained with the packet. No matter VLAN tag is stripped or not, the VLAN tag information will be stamped at RX Descriptor.

5.18.4.1.5 Pause Frame Processing

Pause frame are not transferred to the receive FIFO.
A pause frame is valid only, if all the following conditions are valid:

- Length / Type is set to 0x8808
- The Opcode immediately following the Length / Type field is 0x0001
- The frame MAC destination address is either the configured unicast address (Registers MAC_ADDR_0 and MAC_ADDR_1) or the control frame multicast address 01-80-c2-00-00-01
- The frame has a valid CRC
- The frame has a length of 64 octets

Check if the received packet is a pause frame and generate signal to pause TX (pause_tx) when receiving pause on frame.

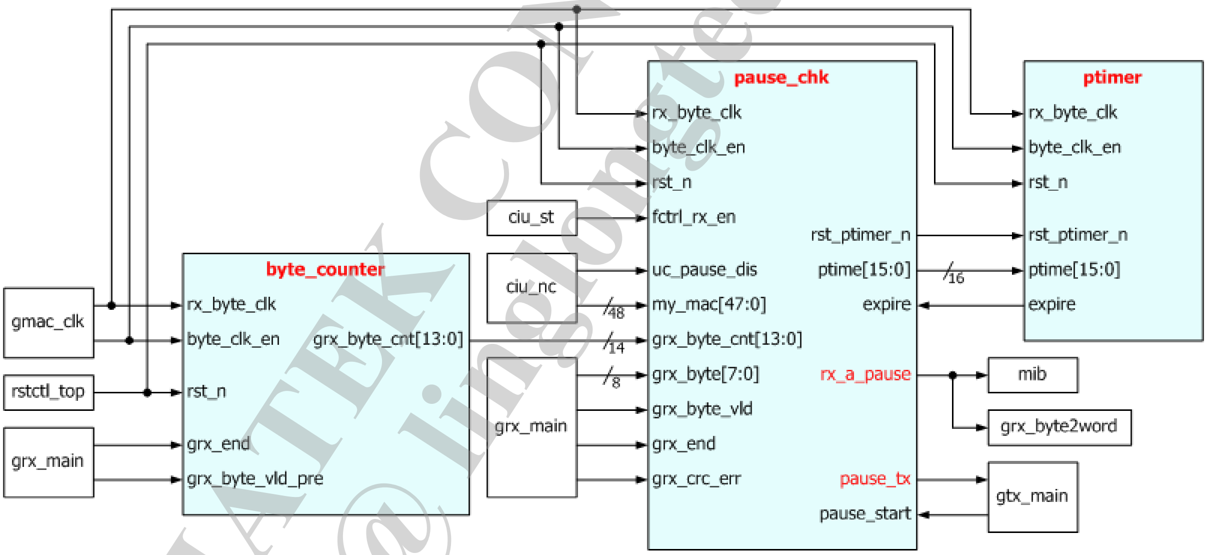


Figure 5-47. pause_chk Block Diagram

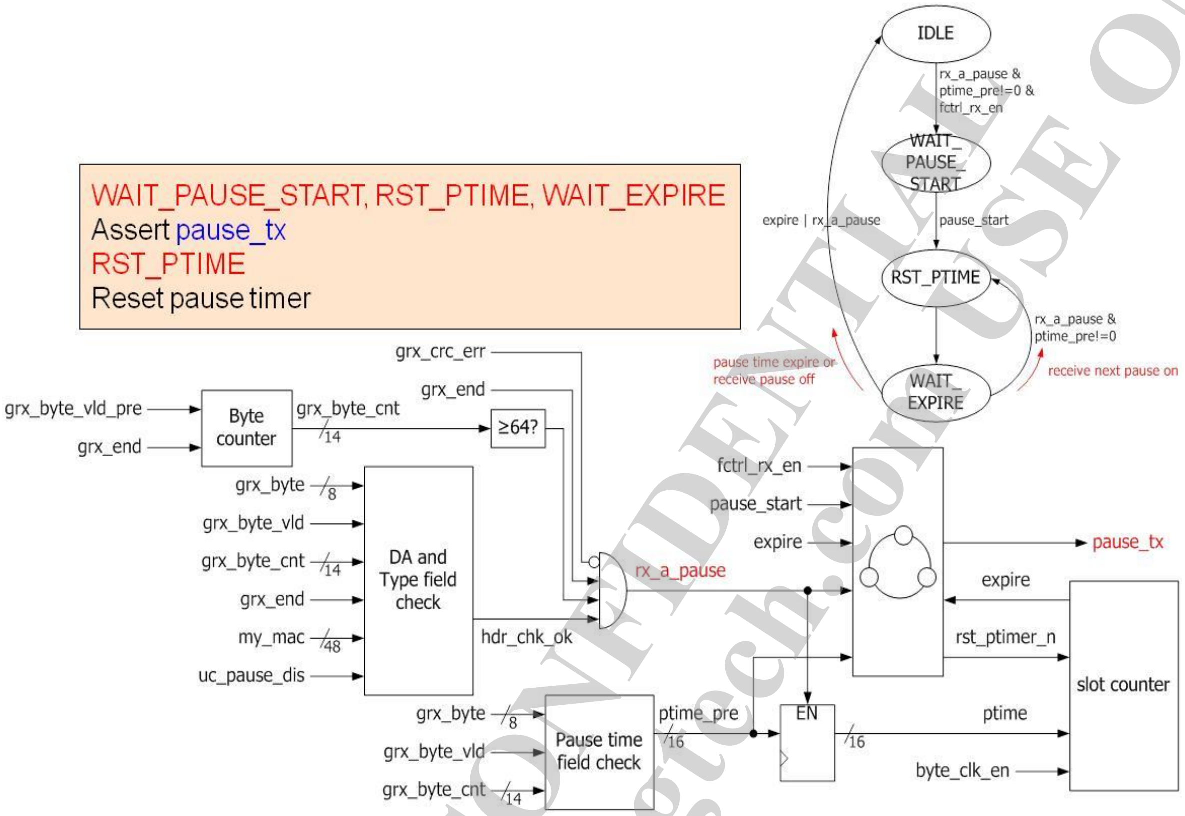


Figure 5-48. pause_chk Architecture

5.18.4.1.6 CRC Check

The CRC-32 field is always checked in the received side . The CRC polynomial, as specified in the 802.3 Standard, is:
FCS(X) = $X_{32} + X_{26} + X_{23} + X_{22} + X_{16} + X_{12} + X_{11} + X_{10} + X_{8} + X_{7} + X_{5} + X_{4} + X_{2} + X_{1} + 1$

The 32 bits of the CRC value are placed in the FCS field so that the X31 term is the right-most bit of the first octet. The CRC bits are thus received in the following order: X31, X30,..., X1, X0.

If a CRC-32 error is detected, the frame is marked invalid and the frame status bit 1 indicating a CRC error is set to "1"

Following is grx_main module for packet reception and CRC error check.

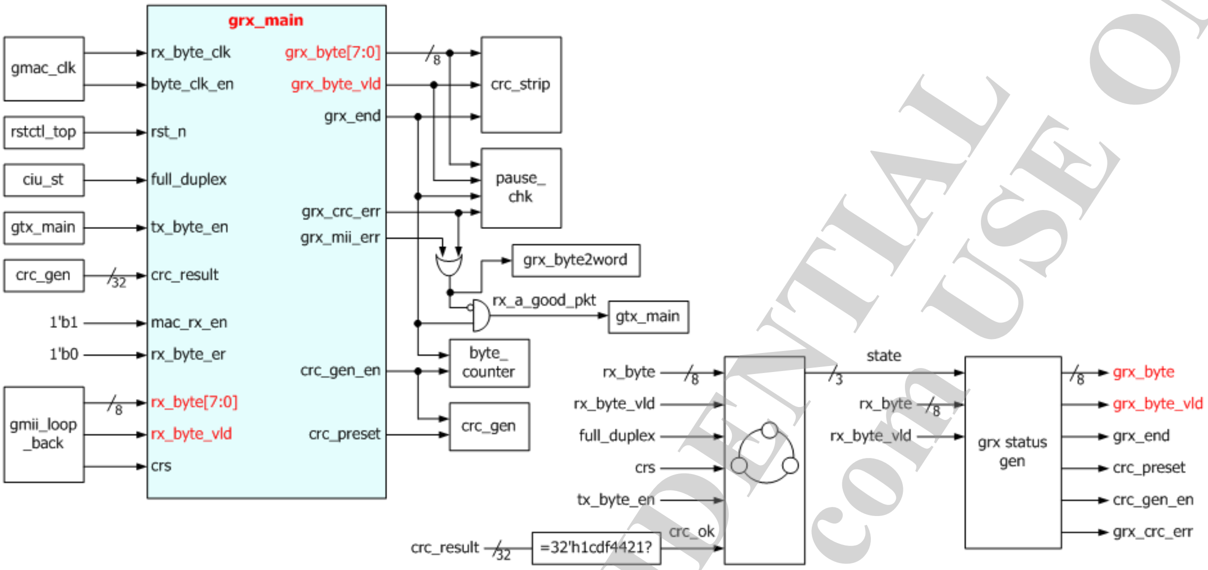


Figure 5-49. grx_main Block Diagram

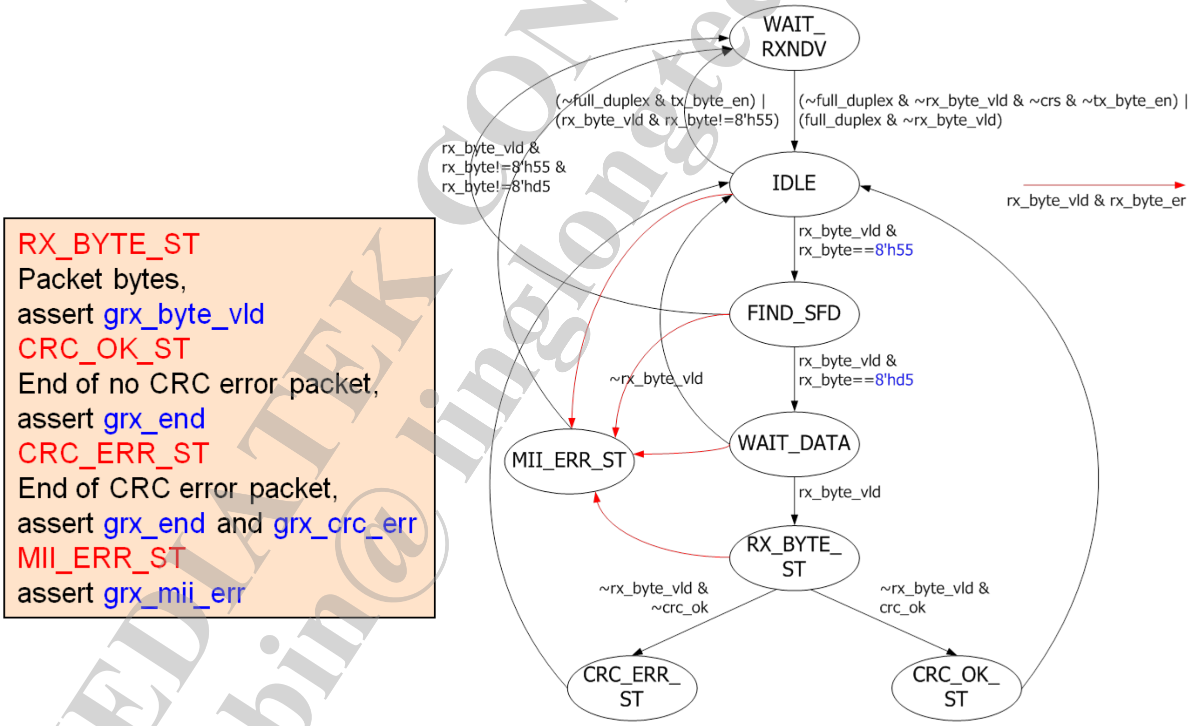


Figure 5-50. grx_main State Machine

5.18.4.1.7 Frame Padding

In receive, the MAC does not remove the padding octets even if the Payload lengths is less than 46 Bytes (42 Bytes for VLAN tagged frames)

5.18.4.1.8 Frame Truncation

Since NIC does not do length field checking, that function of frame truncation is not implemented.

5.18.4.1.9 Hash Table

A Hash table of 512-set is implemented for Multicast MAC Address Filter function. It operates as follows.

- First, on receiving a multicast MAC address frame, it calculates 8/9bit Hash Value from –
- Next, it compares the calculated Hash Value with the SelectBit_n(n = 0 ~ 511) references of the Multicast MAC Address Hash Filter Table.
- Last, it decides whether to forward the frame to memory or to deny it.

If a Hash Value of Rx frame MAC address becomes n and the SelectBit_n is 0, the Rx frame is denied. Otherwise if the SelectBit_n is 1, the Rx frame is forward to memory.

The Hash Value results is aggregated form CRC32 calculation from when CRC32 is used in the following generator polynomial to degenerate the destination MAC address (48 bits) .
Generator Polynomial: $= x^{32}+ x^{26}+ x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^{8}+ x^{7}+ x^{5}+ x^{4}+ x^{2}+ x^{1}+ 1$

The 9-bit hash value can be generated from
{crc[15],crc[0],crc[1],crc[2],crc[3],crc[4],crc[5],crc[6],crc[7]} or { DA[40],DA[7:0]}

5.18.4.1.10 Magic Packet Detection

Wake-on-LAN ("WOL") is implemented using a specially designed packet called a magic packet. The magic packet is a broadcast frame containing anywhere within its payload 6 bytes of all 255 (FF FF FF FF FF FF in hexadecimal), followed by sixteen repetitions of the target computer's 48-bit MAC address, for a total of 102 bytes.

When Wake-on-LAN is enabled, TX MAC will be powered down and RX MAC will only scan Magic Packet and not forward any packet to system memory. After detecting the Magic Packet, MAC asserts WOL interrupt to CPU and wake-up CPU accordingly.

5.18.4.2 MAC Transmit

5.18.4.2.1 Overview

Ethernet Frame transmission starts when the Transmit FIFO holds enough data. Once a transfer has started, the transmit engine performs the following tasks:

- Convert word to byte
- Generate Preamble and SFD field before Frame transmission
- When in Link Pause Mode, generate Pause frames if the Receive FIFO reports a congestion or if the pause generation pin back_pressure on gtx_main is asserted
- When in Link Pause Mode, suspend Ethernet Frame transfer (XOFF) if a non zero Pause

Quanta is received from the MAC receive path (optional)

- Calculate and replace CRC-32 to the transmitted frame (optional)
- Send Frame with correct Inter Packet Gap (IPG)

1G/100M/10M mode default stop TX when link status of PHY is deassert.

Following figure show the TX transmit flow chart

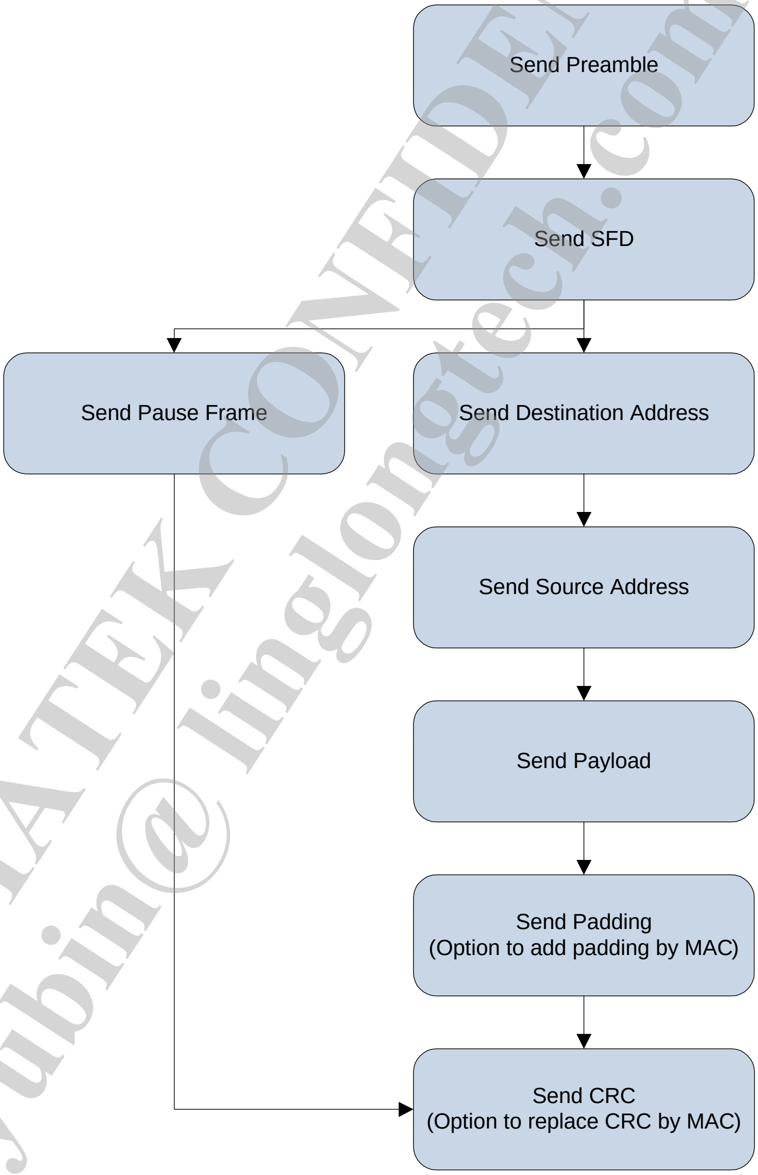


Figure 5-51. TX Transmit Flow Chart

5.18.4.2.2 Frame Payload Padding

The IEEE specification defines a minimum frame length of 64 bytes. If the frame sent to the MAC from the user application has a size smaller than 64 bytes, the MAC automatically (optional, controlled by TX_AUTO_PAD register) inserts padding bytes so that frames transmitted to the Ethernet link do not violate the Ethernet minimum frame length specification (60 byte frame data plus 4 byte FCS). The content of the pad is unspecified. If padding is already provided by the user application, the MAC does no further action.

5.18.4.2.3 Frame Truncation

It is the responsibility of the application to ensure frames with at most 1536 octets are written into the transmit FIFO. If too long frames are provided they will not be truncated by the NIC. If TX FIFO is underflow, the current frame is truncated and appended with wrong CRC. The remaining part of this frame in the FIFO is discarded and the FIFO read pointer jumps to the start address of the next packet.

5.18.4.2.4 CRC Calculation

The CRC-32 field is generated and replacing the last 4 bytes of a Frame. The CRC polynomial, as specified in the 802.3 Standard, is:

$$FCS(X) = X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+1$$

The 32 bits of the CRC value are placed in the FCS field so that the X31 term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: X31, X30,..., X1, X0.

5.18.4.2.5 Half Duplex Operation

The NIC supports Half Duplex in 10/100 Mbps mode. In half duplex mode, the carrier_detect module in gmac_tx will detect if the link is busy before it starts transmitting. If the link is currently busy, the carrier sense will be asserted and the TX will wait until the link is available.

5.18.4.2.6 Transmit Deferral

To account for the clock frequency variation caused by the circuit tolerance, the NIC makes an unconditional decision to transmit at some point in the deferral process when waiting out an interframe gap, regardless of whether it hears another station after that point. This is depicted in the figure below. The defer_ckt in gmac_tx checks for carrier sense during the first 64 bit times. If carrier sense is asserted during this period, it will hold the TX from start sending the next frame.

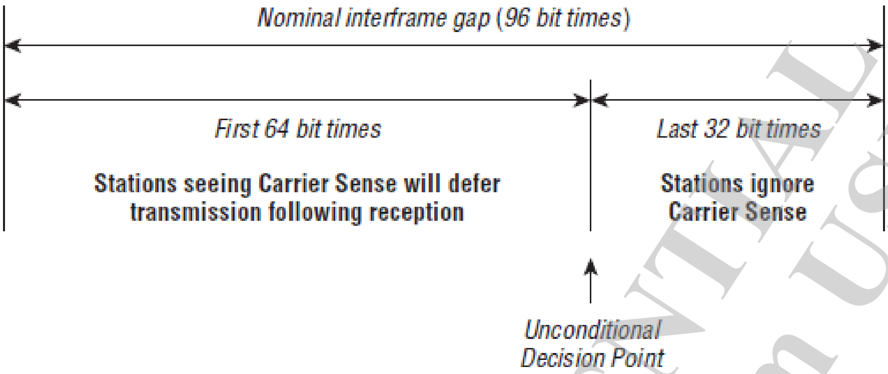

Nominal interframe gap (96 bit times)
First 64 bit times
Stations seeing Carrier Sense will defer transmission following reception
Last 32 bit times
Stations ignore Carrier Sense
Unconditional Decision Point

Figure 5-52. Interframe Gap

5.18.4.2.7 Collision Handling

If collision occurs when TX is sending frames, the TX will enter the JAM state and send JAM code (8'h55) for 4 cycles. The NIC will try to retransmit this frame for 16 times before this frame is discarded due to too many retry attempts. The backoff time between each retry is defined by a random variable r. The definition for r on the nth transmit attempt is as below.

$$0 \le r < 2^k$$

$$\text{, where } k = MIN(n,\ 10)$$

5.18.4.2.8 Inter-Frame Gap

5.18.4.2.9 IPG Biasing

The default value of IPG is 96 bit times and can be configured by application using the IPG field in the MAC configuration register. In the IPG field, IPG[4:2] represents the 2/3 of the IPG and the IPG[1:0] represent the 1/3 of the IPG. The number of their values are set to be number of byte_clock-1, where one byte clock is 2 cycles for MII.

Table 5-9. IPG Field Value

IPG (4:0)	IPG in bit time
111_11	96
110_11	88
101_11	80
100_11	72

011_11	64
010_11	56
001_11	48
001_10	40
001_01	32

Configuring the IPG leading to a slight increase in bandwidth can be used to compensate for oscillator differences in certain applications. However, it is a non-standard behavior, and causes the sending device to exceed the nominal bandwidth.

The IPG corresponding to different IPG field value in MAC configuration register is shown here.

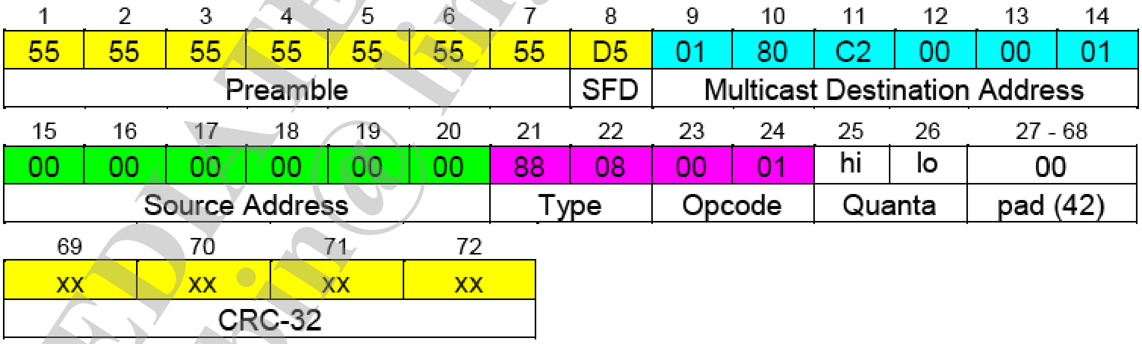
5.18.4.3 Flow Control

5.18.4.3.1 Overview

The MAC supports flow control: Link Pause Flow Control in full-duplex mode and Backpressure Flow Control in half-duplex mode.

5.18.4.3.2 Link Pause Flow Control Frames in Full-Duplex

Link Pause Flow Control: This is the standard IEEE 802.3 defined pause frame to allow pausing the remote device connected to the link (link local pause). Upon reception (if enabled by configuration) the transmitter is paused for the time received in the pause frame.
The IEEE 802.3 defined pause frame has the following format:



1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					

15	16	17	18	19	20	21	22	23	24	25	26	27 - 68
00	00	00	00	00	00	88	08	00	01	hi	lo	00
Source Address						Type		Opcode		Quanta		pad (42)

69	70	71	72
xx	xx	xx	xx
CRC-32			

Figure 5-53. Format of Pause Frame

There is no Payload Length field found within a Pause Frame and a Pause Frame is always padded with 42 bytes (0x00).
If a pause frame with a pause value greater than zero (XOFF Condition) is received, the MAC stops transmitting data as soon as the current Frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512 bit times.

If a pause frame with a pause value of zero (XON Condition) is received, the transmitter is allowed to send data immediately.

5.18.4.3.3 Backpressure Flow Control in Half-Duplex

When the threshold in RX FIFO is reached, RX asserts the send_pause signal to trigger TX backpressure flow control. Both the Force Carrier Sense and Force Collision are supported by NIC. The default backpressure method is Force Collision, which sends 8'h55 for 8 cycles. If Force Carrier Sense is chosen over Force Collision, it sends out 72 bytes carrier.

5.18.4.3.4 Transmit Pause Operation

The sd_pause module inide the NIC detects the send_pause signal from RX fifo if the amount of packets inside RX fifo is larger than the threshold value send_pause_th. The state machine inside the sd_pause for TX send_pause is shown as below. Note that the thresuold value send_pause_th can be adjusted by software.

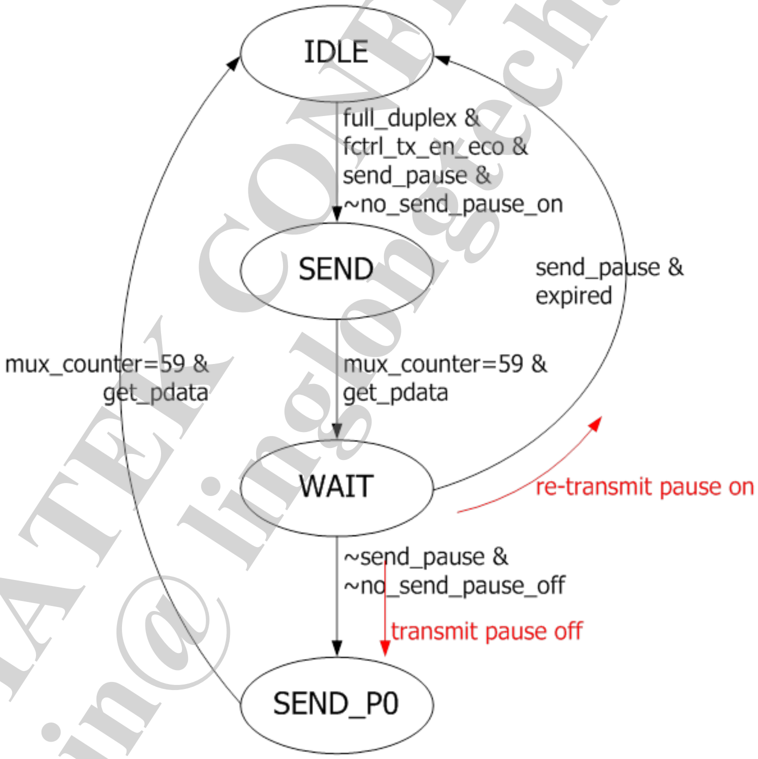


Figure 5-54. sd_pause State Machine

The PAUSE data is formed in the sd_pause using the register **pdata**. When the sate of the FSM in sd_pause is SEND or SEND_P0, sd_pause triggers the pdata_vld to gtx_main. When pdata_vld is asserted, TX will start sending PAUSE frame after all the data in current packet are sent. The transmission of pause frame is affected by no_send_pause_on and no_send_pause_off.

After the completion of a frame, the gtx_main inside the NIC samples the pdata_vld input and determines if, depending on the current mode, a Link Pause control frame should be immediately scheduled. The following cases can exist:

- A pause is not in progress and the XOFF bit is set, so the current timer value is 0. In this case a new Pause/PFC frame should be sent with the programmed quanta value.
- A pause is already in progress for this priority, but the XOFF bit is now cleared, a new Pause/PFC frame with QUANTA = 0 needs to be sent.
- A pause is already in progress for that priority and the XOFF bit is still set, but the quanta timer is between its max value and the threshold value, no new pause update is needed, so send the next application frame.
- A pause is already in progress for this priority and the XOFF bit is still set, but the timer threshold has been reached. A refresh PFC/Pause control frame should be scheduled with the programmed quanta timer value.
- A pause is not in progress and the XOFF bit is cleared, no pause is needed, so send the next application frame.

5.18.4.3.5 Receive Pause Operation

The check of whether a Link Pause frame is received is carried out in the pause_chk module. The state machine of pause_chk is shown as below.

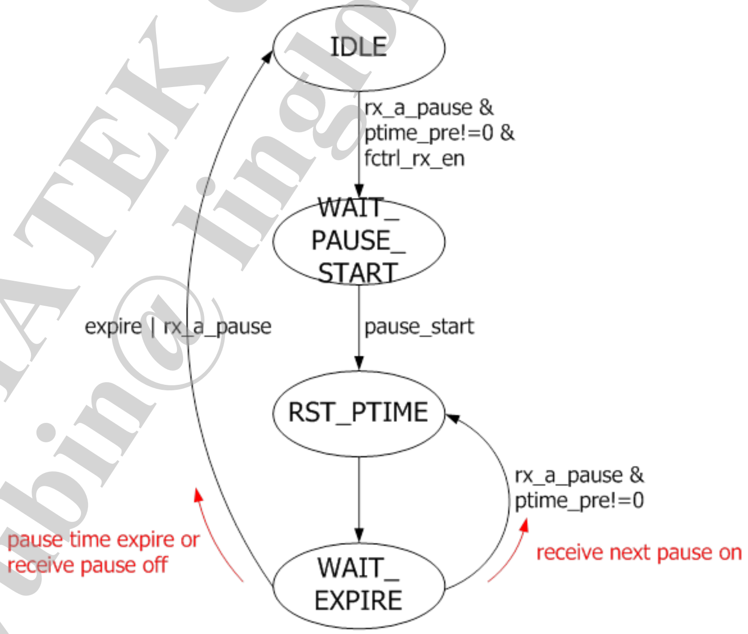

IDLE
rx_a_pause & ptime_pre!=0 & fctrl_rx_en
WAIT_ PAUSE_ START
pause_start
expire | rx_a_pause
RST_PTIME
rx_a_pause & ptime_pre!=0
pause time expire or receive pause off
receive next pause on
WAIT_ EXPIRE

Figure 5-55. pause_chk State Machine

The architecture of the pause_chk is shown here.

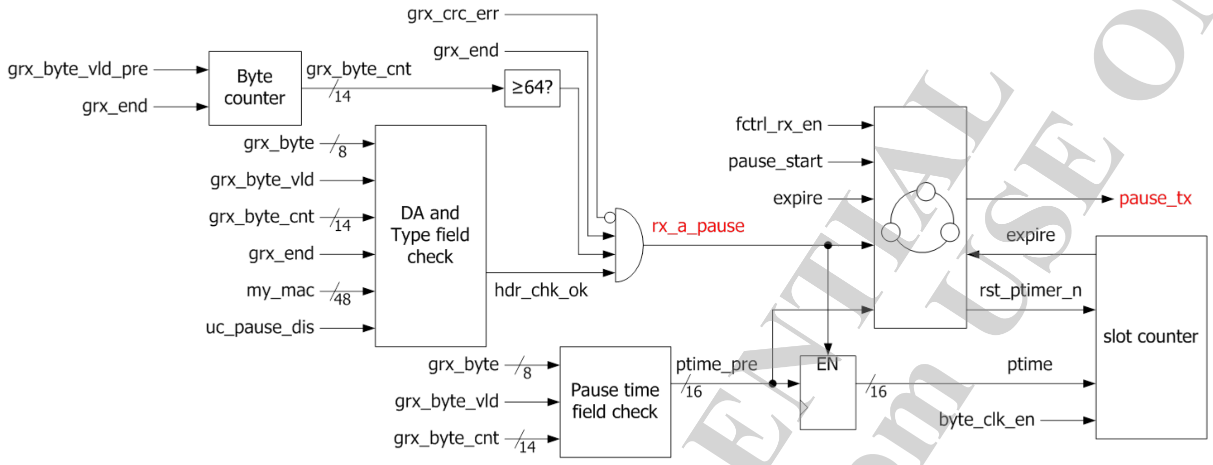


Figure 5-56. pause_chk Architecture

When a Link Pause frame is received the quanta is extracted and loaded into an internal timer to pause the transmitter. A pause_tx signal is sent to the transmitter. The transmitter continues to complete any ongoing frame transmission and then enters a pause state where it does not read any user frames from the transmit FIFO (optinal). MAC layer pass the pause frame to application layer and indicate the frame is the pause frame.

When the transmitter has reached its pause state, the timer starts to decrement. When the timer reaches 0 the transmitter resumes to normal transmission of frames (optional).

5.18.4.4 Energy Efficient Ethernet Interface

The IEEE 802.3az defines procedures to implement energy efficient Ethernet (EEE). It allows end stations to exchange a so-called low power idle (LPI) sequence to indicate the link is not used and may be allowed to power down.

The MAC provides mechanisms to allow this signaling of low power idle indications. It is then the responsibility of the application and management to use the information and implement power modes as necessary.

The following functions are available for use by the application.

Table 5-10. LPI Sequence Transmission and Reception Functions for EEE

Function	Description
Indicate low power idle to the remote	MAC to transmit the low-power idle (LPI) sequence. Occurs when the configuration bit LPI_MODE_EN is set, and the TX is idle for a period. If LPI_MODE_EN is set, the LPI sleep timer will start to count whenever TX is idle. When the LPI sleep timer reaches the LPI sleep threshold set by the register LPI_SLEEP_TH, the TX will start to transmit LPI sequence to the remote.

Function	Description
Indicate reception of low power idle indication from remote.	MAC receives low-power idle (LPI) sequence. When the LPI_MODE_EN is set and the LPI sequence is detected in the RS layer, the RX sends a interrupt rx_pcode_int to the CPU.

Including tn_dma IDLE, TX FIFO empty,... etc. Therefore, the LPI sleep timer does not start counting immediately when MII interface becomes IDLE.
An example of the LPI behavior is shown below:

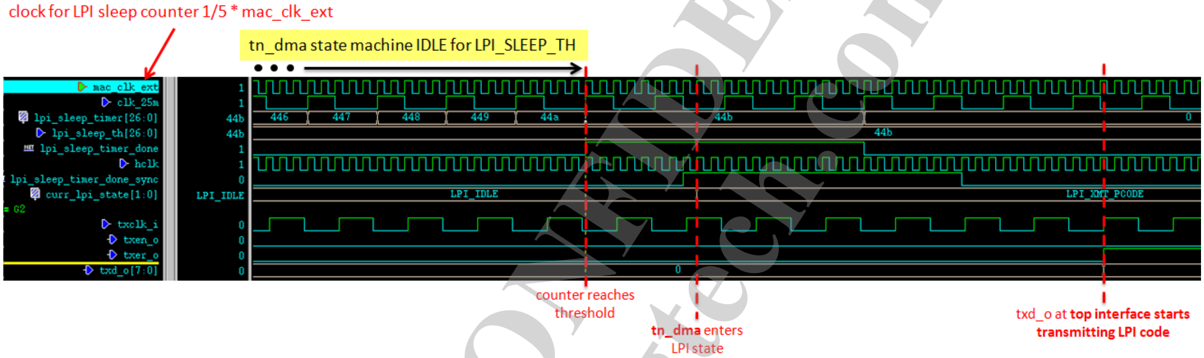


Figure 5-57. Waveform of TX in LPI Mode

5.18.4.4.1 LPI Signaling with MII

When operating in MII, the LPI sequence is indicated by asserting the error signal (txer_o, rxer_i) while the data valid signal (txen_o, rxdv_i) is low and the data bus presents the value 1 as illustrated below.

The PHY then encodes the MII Low Power signaling with the corresponding Low Power Idle sequences. On receive the PHY detects the LPI sequences and presents it to the MAC using the same MII signaling.

Note that the NIC does not support LPI sequence when it is operating in 10M mode.

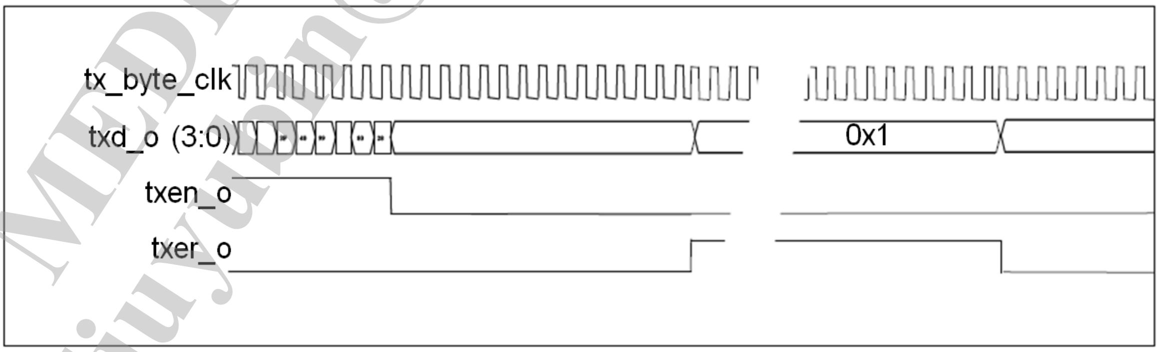


Figure 5-58. Waveform of MII Interface in LPI Mode

5.18.4.4.2 TX Force LPI Mode

For verification purpose, the NIC supports force LPI mode that can be enabled or disabled by software. The behavior of force LPI mode is shown as below:

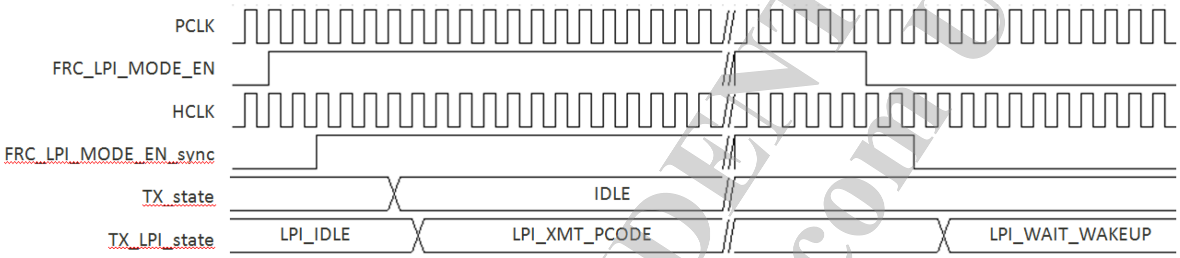


Figure 5-59. Waveform in Force LPI Mode

Notice that, just like entering normal LPI mode, the NIC wait until the TX enters IDLE state before it start transmitting LPI code. In another word, if FRC_LPI_MODE_EN is asserted while TX is transmitting a packet, the NIC wait until the TX finishes transmission of the current packet before it enters LPI mode.

As long as the FRC_LPI_MODE_EN is asserted, the NIC will stay in LPI mode. When FRC_LPI_MODE_EN is deserted, the NIC transit to LPI_WAIT_WAKEUP state and wait for PHY to wakeup.

5.18.4.5 DMA Operation

The DMA controller forwards packets between host memory and embedded packet memory within NIC. It implements sophisticated descriptor ring architecture, and support multiple segments for a TX/RX packet to comply modern zero-copy socket driver architecture. The following diagram shows the architecture. It also supports READ-Alignment and WRITE-Alignment for both transmit path and receive path respectively. The READ-Alignment feature enhances the DMA performance in cache-line oriented accessing, by terminating transmit DMA cycles on a cache line boundary and start the next transaction on a cache-line aligned address.

The WRITE-Alignment feature allows a packet to be stored at 2-bytes address offset from a cache line boundary at host memory. This feature meets 2-byte offset requirement in protocol stack for high-level commercial RTOS (like VxWorks) or Linux OS, and achieves zero-copy from Ethernet driver to TCP/IP protocol stack software to enhance the packet transferring efficiency.

Below shows the descriptor ring architecture.

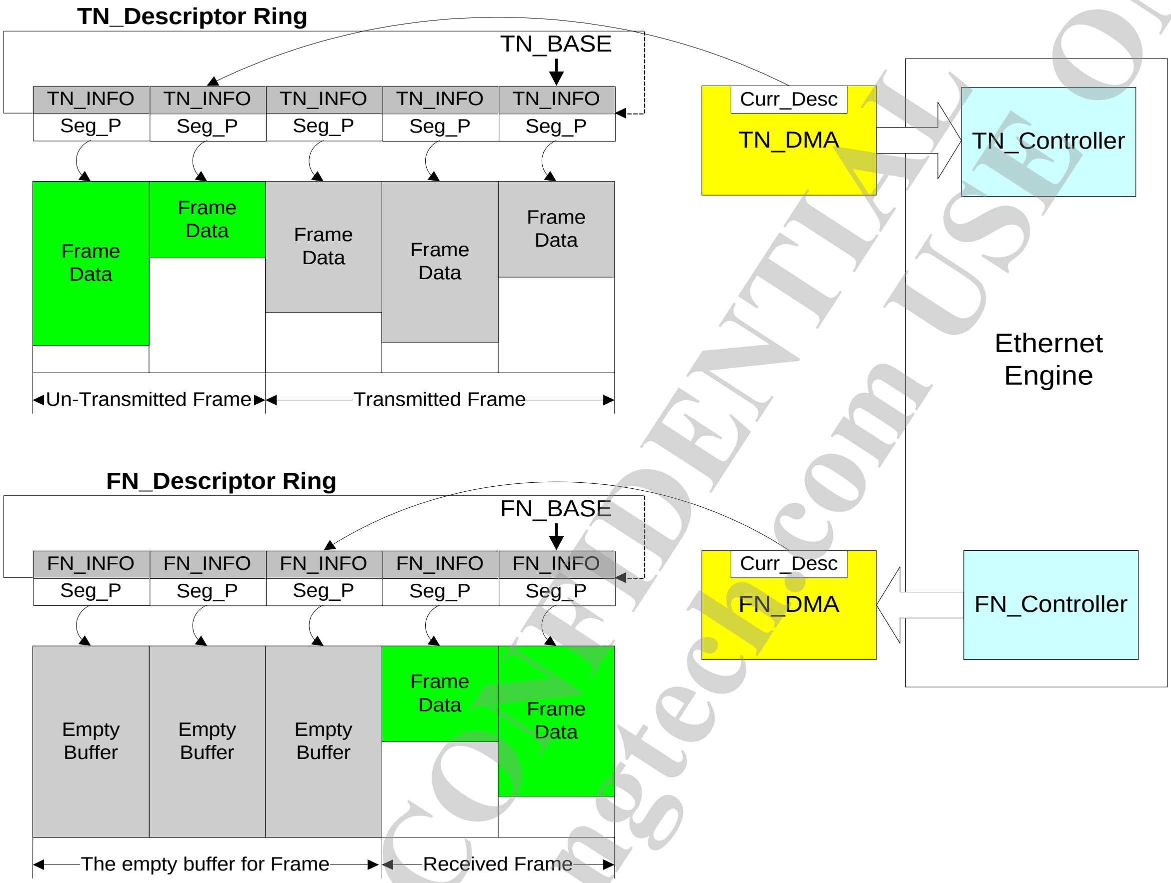


Figure 5-60. Descriptor Ring Architecture

The detailed TN and FN descriptor formats are shown below.

Table 5-11. TN Descriptor Format description

Offset	Bit	Symbol	Descriptions
0	31	COWN	CPU Ownership: This bit, when set, indicates that the descriptor owned by CPU. When cleared, it indicates that the descriptor own by the DMA. The DMA sets this bit when the relative segment data is transmitted and return it to the CPU.
0	30	EOR	End of descriptor ring: This bit, when set, indicates that this is the last descriptor in the descriptor ring. When DMA's internal transmit pointer reaches here, the pointer will return to the first descriptor (TX_DES_BASE, reg. 0x110) of the descriptor ring
0	29	FS	First Segment descriptor: This bit, when set, indicates that this is the first descriptor of a TX packet, and that this descriptor is pointing to the first segment of the packet.
0	28	LS	Last Segment descriptor: This bit, when set, indicates that this is the last

Offset	Bit	Symbol	Descriptions
			descriptor of a TX packet, and that this descriptor is pointing to the last segment of the packet
0	27	INT	Interrupt: When set, DMA will generate an interrupt (txtc_int) after sending out this packet (not this segment only).
0	26	INSV	Insert VLAN Tag in the following word.
0	25	ICO	Enable IP checksum generation offload
0	24	UCO	Enable UDP checksum generation offload
0	23	TCO	Enable TCP checksum generation offload
0	22	VTG	VLAN Tag
0	21	ICOE	IP checksum offload error. TX will write info0 back with this bit set to 1 if ICO is set to 1 and the transmitting frame is fragmented.
0	20	UCOE	UDP checksum offload error. TX will write info0 back with this bit set to 1 if UCO is set to 1 and the transmitting frame is fragmented.
0	19	TCOE	TCP checksum offload error. TX will write info0 back with this bit set to 1 if TCO is set to 1 and the transmitting frame is fragmented.
0	18	LSO	Enable LSO function
	17	INCID	Whether to increment the LSO packet ID or not. 0: The LSO output packets will have the same ID. 1: The LSO output packets will have increasing IDs.
0	16		Reserved
0	15:0	SDL	Segment Data length: indicate the segment length of this current descriptor in bytes. The minimum SDL value is 64.
4	31:0	SDP	Segment data pointer: point to the starting address of this transmitted data segment. The pointer is allowed to be only byte alignment.
8	31:16	EPID	VLAN Tag EPID
8	15:13	PRI	VLAN Tag Priority
8	12	CFI	VLAN Tag CFI (Canonical Format Indicator)
8	11:0	VID	VLAN Tag VID
12	31:21	MSS	The MSS value (in Bytes) used in LSO function. This value is the maximum TCP payload length of the LSO segmented packets.
12	20:0		The LSO total length in Bytes. This field indicates the total length of the packet before segmentation.

Table 5-12. FN Descriptor Format Description

Offset	Bit	Symbol	Descriptions
0	31:0	SDP	Segment data pointer: point to the starting address of this received data segment. The pointer must be 4-word cache line alignment or offset 2 bytes from the cache line boundary.
4	31	COWN	CPU Ownership: This bit, when set, indicates that the descriptor owned by the CPU. When cleared, it indicates that the descriptor own by the DMA. The DMA sets this bit when the relative segment data is

Offset	Bit	Symbol	Descriptions
			received.
4	30	EOR	End of descriptor ring: This bit, when set, indicates that this is the last descriptor in the descriptor ring. When DMA's internal receive pointer reaches here, the pointer will return to the first descriptor (RX_Des_BASE) of the descriptor ring
4	29	FS	First Segment descriptor: This bit, when set, indicates that this is the first descriptor of a RX packet, and that this descriptor is pointing to the first segment of the packet. CPU should reset this bit when it allocates this descriptor.
4	28	LS	Last Segment descriptor: This bit, when set, indicates that this is the last descriptor of a RX packet, and that this descriptor is pointing to the last segment of the packet CPU should reset this bit when it allocates this descriptor
4	27		Reserved
4	26	PROT[2]	Refer the descriptions after this table
4	25	OSIZE	The Received Packet is oversize.
4	24	CRCE	The Received Packet is CRC Error Note: Only when ACPT_CRC_ERR = 1 of MAC Configuration Register, CRC error packet will be received to CPU.
4	23	RMC	The Received Packet DMAC is Reserved Multicast Address
4	22	HHIT	The Received Packet DMAC is hit in hash table
4	21	MYMAC	The Received Packet DMAC is My_MAC
4	20	VTED	VLAN Tagged in the following word.
4	19:18	Prot[1:0]	Refer the descriptions after this table
4	17	IPF	IP checksum check fail. This bit is meaningful only when Prot != 2'b11. Note: Only when Acpt_CKS_Err = 1 of MAC Configuration Register, checksum error packet will be received to CPU.
4	16	L4F	Layer-4 checksum fail (TCP or UDP over IP). This bit is meaningful only when Prot=2'b01(UDP) or 2'b10(TCP) Note: Only when Acpt_CKS_Err = 1 of MAC Configuration Register, checksum error packet will be received to CPU.
4	15:0	SDL/WPL	Segment Data Length/Whole Packet Length: indicates the length of this received segment in bytes when FS=0, or the length of this received packet when FS=1. CPU should set SDL to the allocated segment buffer length (in bytes) when it allocates the descriptor. DMA will modify this field to the actual data length it fills for the non-first segment (FS=0) or the whole packet length for the first segment (FS=1).
8	31:16	EPID	VLAN Tag EPID
8	15:13	PRI	VLAN Tag Priority
8	12	CFI	VLAN Tag CFI (Canonical Format Indicator)
8	11:0	VID	VLAN Tag VID
12	31:0		Reserved

PROT[2:0] (protocol) :

3'b000: (IPV4H5 & Fragment) or (IPV4H5NF & not TCP & not UDP) (can do IP checksum)
3'b001: IPV4H5NF & UDP (can do IP/UDP checksum)
3'b010: IPV4H5NF & TCP (can do IP/TCP checksum)
3'b100: (IPV6 & Fragment) or (IPV6NF & not TCP & not UDP)
3'b101: IPV6NF & UDP (can do IP/UDP checksum)
3'b110: IPV6NF & TCP (can do IP/TCP checksum)
3'b111: Others (no any checksum offload is done)

5.18.4.6 IPv4/UDP/TCP Checksum offload

The NIC performs checksum update in the mtx_data.

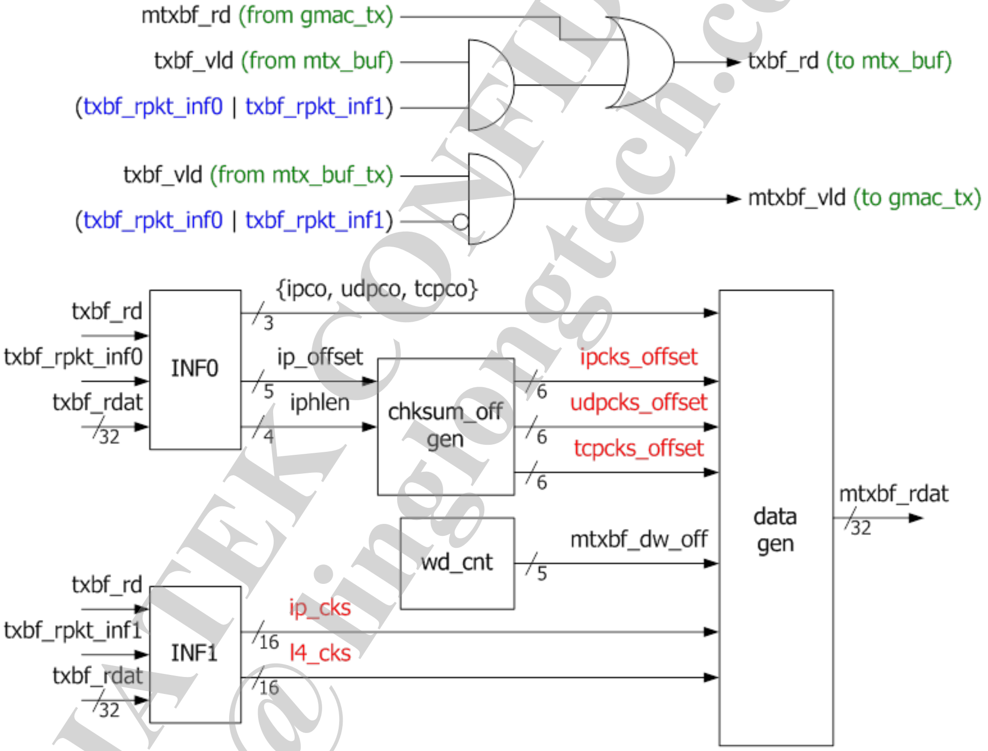
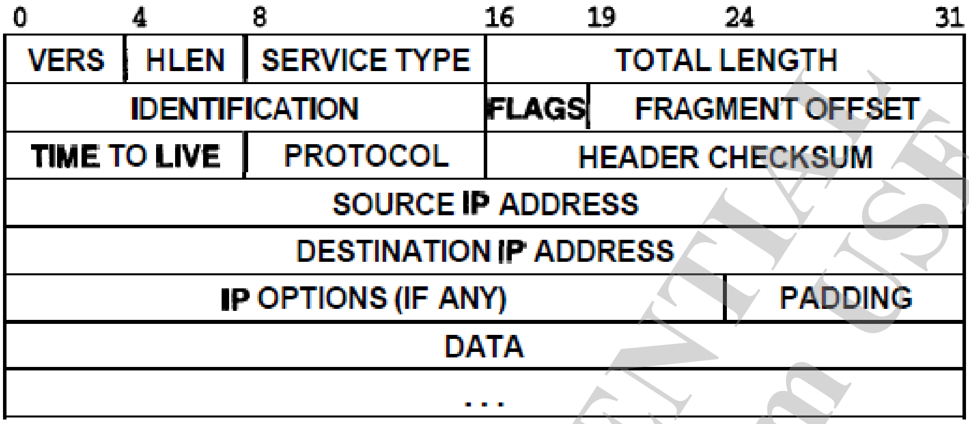


Figure 5-61. mtx_data Block Diagram

5.18.4.6.1 IPv4 Checksum Offload

The format of IPv4 datagram header is shown here:



<table>
<tr><th>0</th><th>4</th><th>8</th><th>16</th><th>19</th><th>24 31</th></tr>
<tr><td>VERS</td><td>HLEN</td><td>SERVICE TYPE</td><td colspan="3">TOTAL LENGTH</td></tr>
<tr><td colspan="3">IDENTIFICATION</td><td>FLAGS</td><td colspan="2">FRAGMENT OFFSET</td></tr>
<tr><td colspan="2">TIME TO LIVE</td><td>PROTOCOL</td><td colspan="3">HEADER CHECKSUM</td></tr>
<tr><td colspan="6">SOURCE IP ADDRESS</td></tr>
<tr><td colspan="6">DESTINATION IP ADDRESS</td></tr>
<tr><td colspan="5">IP OPTIONS (IF ANY)</td><td>PADDING</td></tr>
<tr><td colspan="6">DATA</td></tr>
<tr><td colspan="6">. . .</td></tr>
</table>

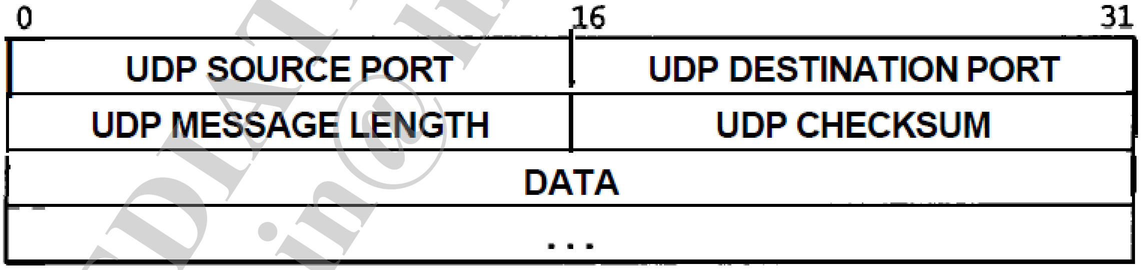
Figure 5-62. Format of IPv4 Header

The first 4-bit field in a datagram (VERS) contains the version of the IP protocol that was used to create the datagram. The header length field (HLEN), also 4 bits, gives the datagram header length measured in 32-bit words. The TOTAL LENGTH field gives the length of the IP datagram measured in octets, including octets in the header and data.

Field ***HEADER CHECKSUM*** in the IPv4 header ensures integrity of header values. The IP checksum is formed by treating the header as a sequence of 16-bit integers (in network byte order), adding them together using one's complement arithmetic, and then taking the one's complement of the result. For purposes of computing the checksum, field ***HEADER CHECKSUM*** is assumed to contain zero. It is important to note that the checksum only applies to values in the IP header and not to the data.

5.18.4.6.2 UDP Checksum Offload

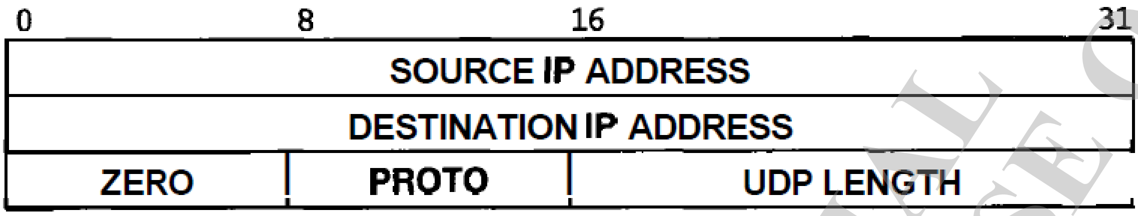
The format of UDP datagram is shown below:



<table>
<tr><th>0</th><th>16 31</th></tr>
<tr><td>UDP SOURCE PORT</td><td>UDP DESTINATION PORT</td></tr>
<tr><td>UDP MESSAGE LENGTH</td><td>UDP CHECKSUM</td></tr>
<tr><td colspan="2">DATA</td></tr>
<tr><td colspan="2">. . .</td></tr>
</table>

Figure 5-63. Format of UDP Datagram

The UDP checksum covers more information than is present in the UDP datagram alone. To compute the checksum, UDP prepends a ***pseudo-header*** to the UDP datagram, appends an octet of zeros to pad the datagram to an exact multiple of 16 bits, and computes the checksum over the entire object. The format of a UDP pseudo-header is shown as below, where the PROTO field is 17 for UDP and the UDP Length field is the length of the UDP datagram.



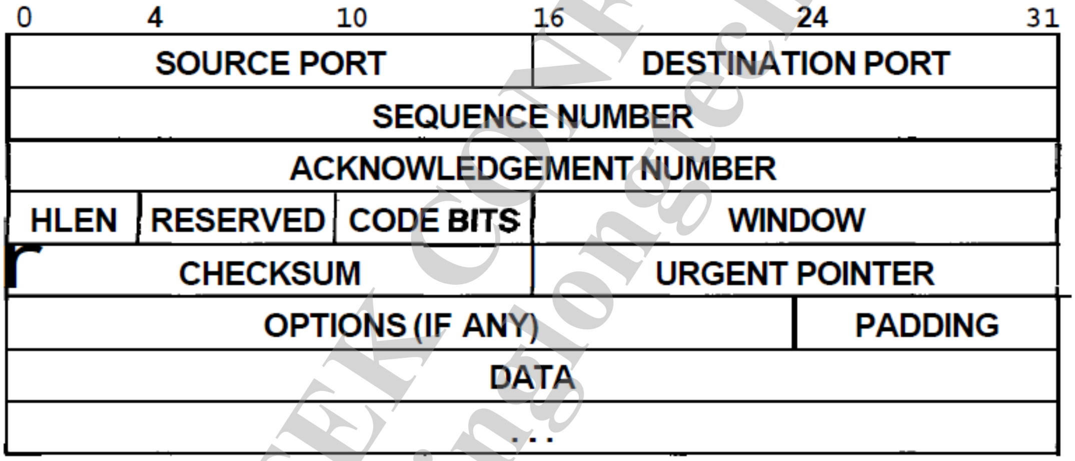
0	8	16	31
SOURCE IP ADDRESS			
DESTINATION IP ADDRESS			
ZERO	PROTO	UDP LENGTH	

Figure 5-64. format of UDP pseudo-header

The octet used for padding and the pseudo-header are ***not*** transmitted with the UDP datagram, nor are they included in the length. To compute a checksum, the software first stores zero in the ***CHECKSUM*** field, then accumulates a 16-bit one's complement sum of the entire object, including the pseudo-header, UDP header, and user data.

5.18.4.6.3 TCP Checksum Offload

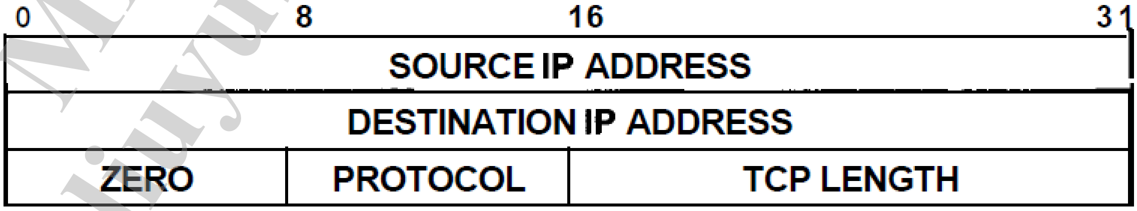
The format of a TCP segment is shown below. The HLEN field indicates the length of the segment in 32-bit multiples.



0	4	10	16	24	31
SOURCE PORT			DESTINATION PORT		
SEQUENCE NUMBER					
ACKNOWLEDGEMENT NUMBER					
HLEN	RESERVED	CODE BITS	WINDOW		
CHECKSUM			URGENT POINTER		
OPTIONS (IF ANY)				PADDING	
DATA					
. . .					

Figure 5-65. Format of TCP Segment

The ***CHECKSUM*** field in the TCP header contains a 16-bit integer checksum used to verify the integrity of the data as well as the TCP header. To compute the checksum, TCP software on the sending machine prepends a pseudo header to the segment, appends enough zero bits to make the segment a multiple of 16 bits, and computes the 16-bit checksum over the entire result. The format of a TCP pseudo-header is shown as below, where the PROTO field is 6 for TCP and the TCP length field specifies the total length of the TCP segment including the TCP header.



0	8	16	31
SOURCE IP ADDRESS			
DESTINATION IP ADDRESS			
ZERO	PROTOCOL	TCP LENGTH	

Figure 5-66. Format of TCP Header

TCP does not count the pseudo header or padding in the segment length, nor does it transmit them. Also, it assumes the checksum field itself is zero for purposes of the checksum computation. As with other checksums, TCP uses 16-bit arithmetic and takes the one's complement of the one's complement sum. At the receiving site, TCP software performs the same computation to verify that the segment arrived intact.

5.18.4.7 MDIO Master Interface

MDIO is a bidirectional signal between the PHY and the station management entity (STA.) It is used to transfer control information and status between the PHY and the STA. Control information is driven by the STA synchronously with respect to MDC and is sampled synchronously by the PHY. Status information is driven by the PHY synchronously with respect to MDC and is sampled synchronously by the STA. A single STA, through a single MDIO interface, can access up to 32 consisting of up to 32 MDIO Managable Devices (MMDs) as shown in the figure below. The MDIO interface can support up to a maximum of 65 536 registers in each MMD.

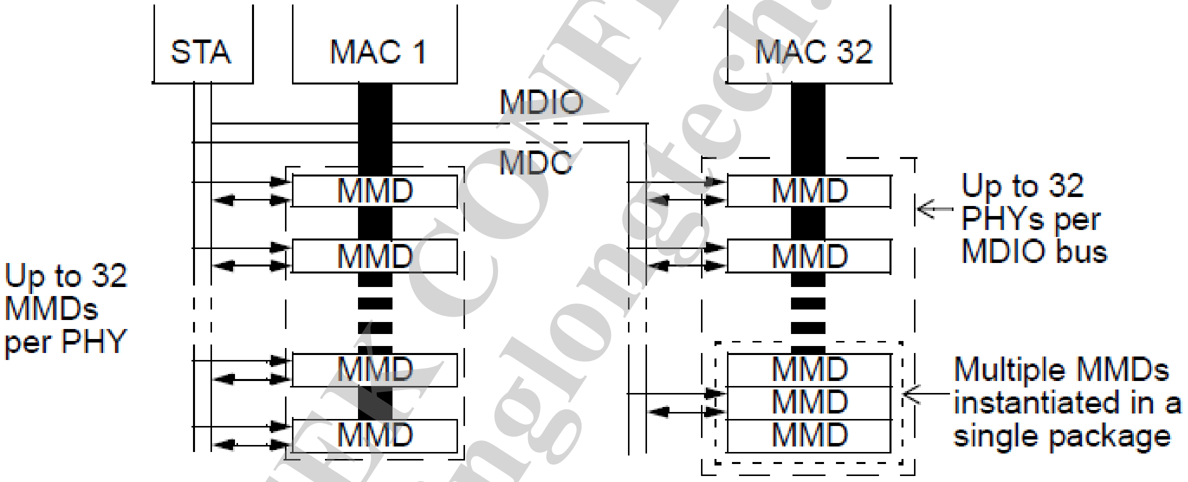

STA
MAC 1
MAC 32
MDIO
MDC
MMD
MMD
MMD
MMD
MMD
MMD
MMD
MMD
MMD
Up to 32 MMDs per PHY
Up to 32 PHYs per MDIO bus
Multiple MMDs instantiated in a single package

Figure 5-67. MAC and MMD Devices

5.18.4.7.1 MDIO Interface Registers

The MDC/MDIO interface of NIC_TOP and its architecture are illustrated as follows:

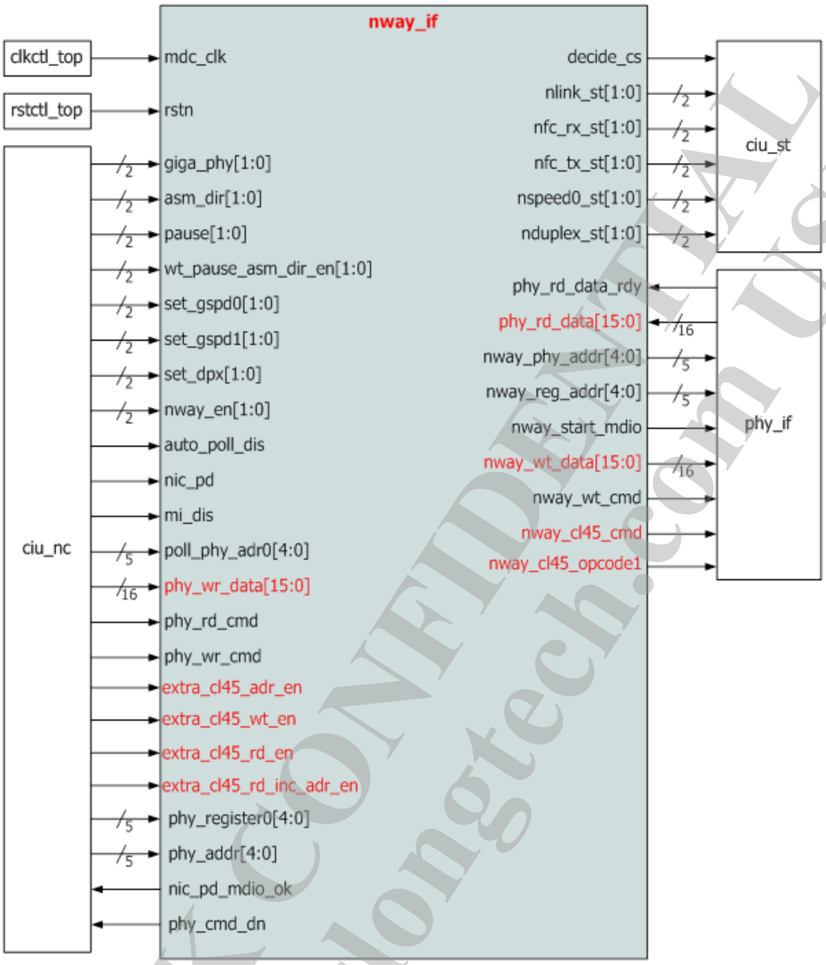


Figure 5-68. MDIO Control Module Interface

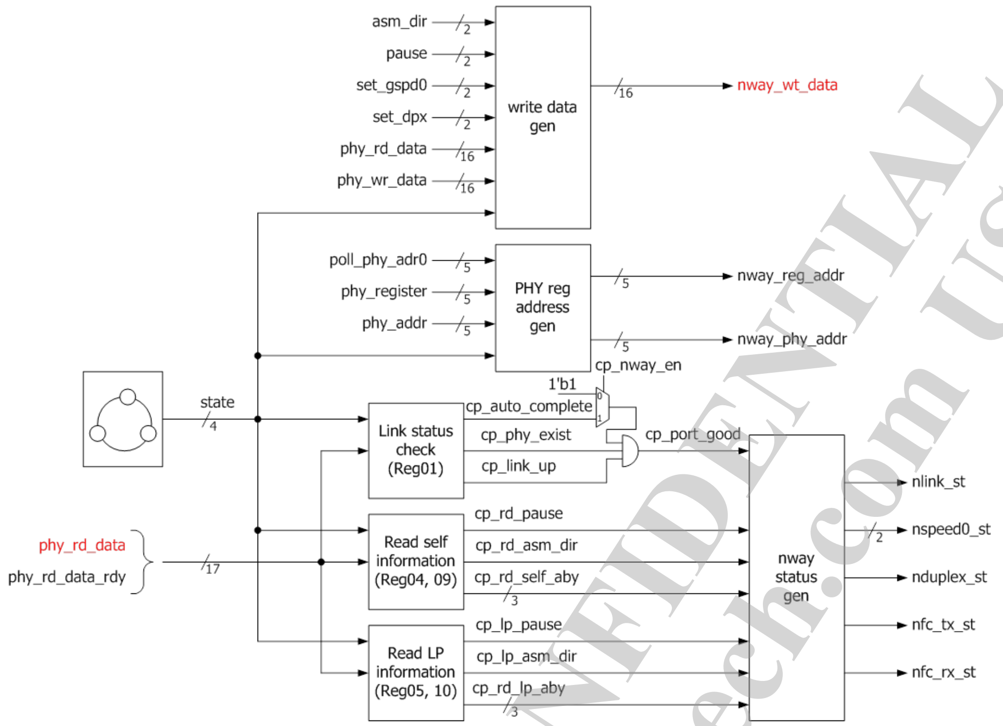


Figure 5-69. MDIO control module architecture

Configurations and data from CPU are sent to the PHY through the nic_cu and phymang, using the PHY control register. Similarly, data and status from PHY are also transfer back to the CPU.
For example, the CPU can enable PHY auto polling by setting AUTO_POLL_DIS to 0 and providing the PHY address used for auto polling. The PHY will return the polling results such as Linkup/Link down, Speed, Half duplex/Full duplex and Flow control capability. If the CPU set AN_EN to 1 to enable auto negotiation, the CPU will send the data specifying desired operation mode to PHY. The data will then be transferred to the other end of the link.

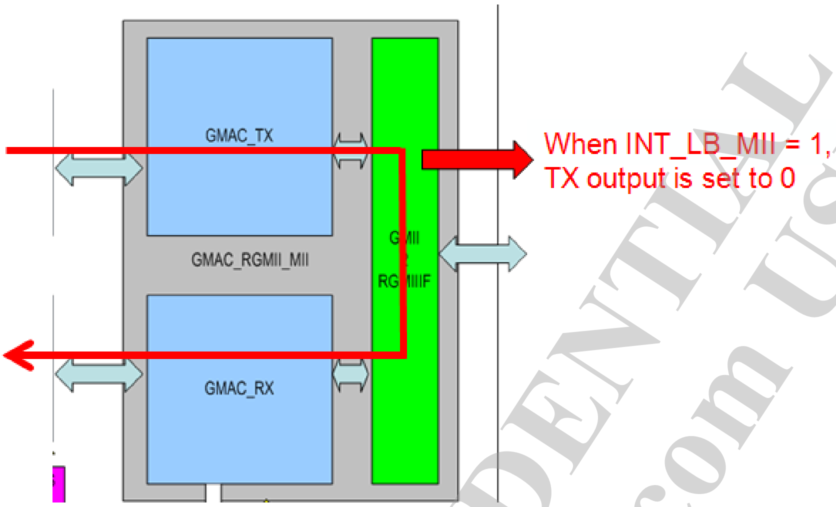
5.18.4.8 Loopback

5.18.4.8.1 Overview

The loopback mode can be enabled by setting the INT_LB_MII register or EXT_LB_MII register.

5.18.4.8.2 MII TX-to-RX Loopback

The TX-to-RX Loopback is carried out when INT_LB_MII is set. The purpose of this test is to test the DMA and TMAC/RMAC of NIC. However the MII interface is not tested.

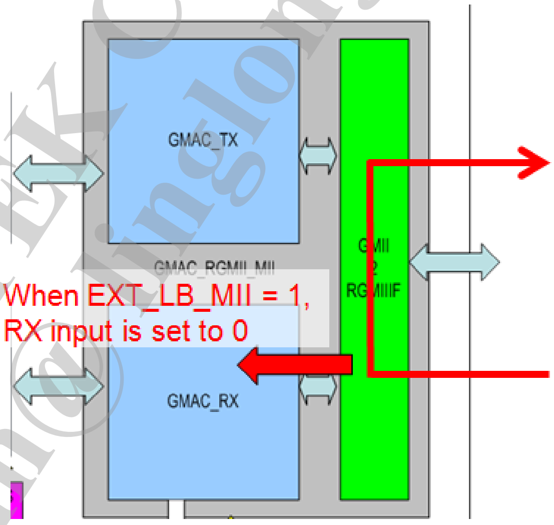


When INT_LB_MII = 1,
TX output is set to 0

Figure 5-70. TX-to-RX Loopback Data Path

5.18.4.8.3 MII RX-to-TX Loopback

The RX-to-TX Loopback is carried out when EXT_LB_MII is set. The purpose of this test is to test whether the internal NIC can correctly TX/RX packet to/from external PHY at 10/100Mbps



When EXT_LB_MII = 1,
RX input is set to 0

Figure 5-71. RX-to-TX Loopback Data Path

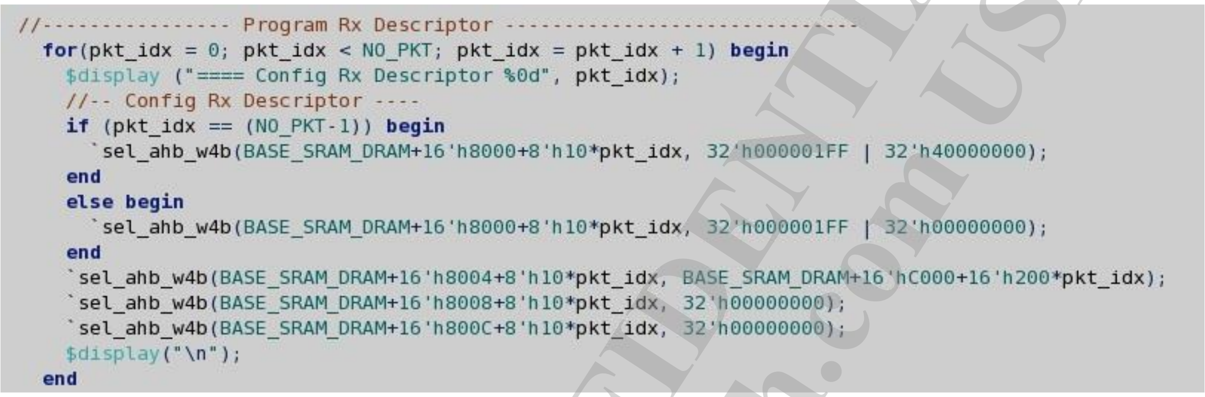
5.18.5 Register Definitions

For register details refer to chapter 3.18 in MT8516A Application Processor Registers."

5.18.6 Programming Guide

5.18.6.1 RX Descriptor Initialization

Here use a testbench of rtl localsim for example. Section 1.1.4.5 illustrate format and field descriptions of rx descriptor.

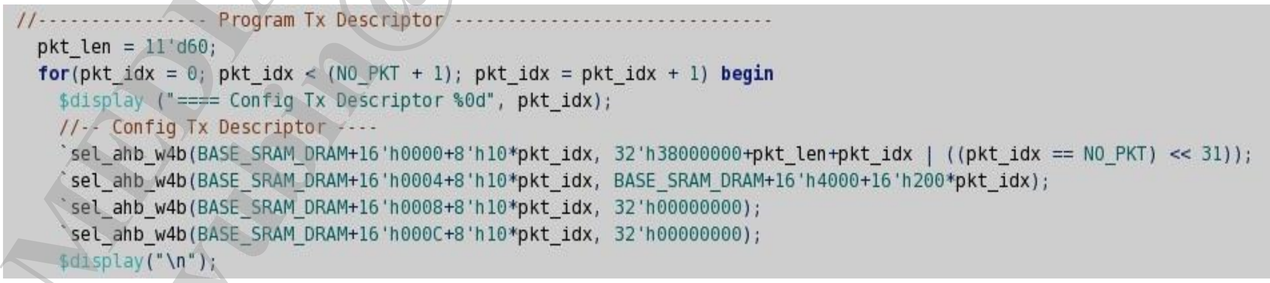


```
//---------------- Program Rx Descriptor --------------------------------
  for(pkt_idx = 0; pkt_idx < NO_PKT; pkt_idx = pkt_idx + 1) begin
    $display ("==== Config Rx Descriptor %0d", pkt_idx);
    //-- Config Rx Descriptor ----
    if (pkt_idx == (NO_PKT-1)) begin
      `sel_ahb_w4b(BASE_SRAM_DRAM+16'h8000+8'h10*pkt_idx, 32'h000001FF | 32'h40000000);
    end
    else begin
      `sel_ahb_w4b(BASE_SRAM_DRAM+16'h8000+8'h10*pkt_idx, 32'h000001FF | 32'h00000000);
    end
    `sel_ahb_w4b(BASE_SRAM_DRAM+16'h8004+8'h10*pkt_idx, BASE_SRAM_DRAM+16'hC000+16'h200*pkt_idx);
    `sel_ahb_w4b(BASE_SRAM_DRAM+16'h8008+8'h10*pkt_idx, 32'h00000000);
    `sel_ahb_w4b(BASE_SRAM_DRAM+16'h800C+8'h10*pkt_idx, 32'h00000000);
    $display("\n");
  end
```

NO_PKT: number of descriptors in rx descriptor ring.
sel_ahb_w4b(address, wdata): task of write 4-byte data via AHB
BASE_SRAM_DRAM+16'h8000: address for start of rx descriptor ring.
8'h10*pkt_idx: space of one descriptor is 16-byte
32'h000001FF | 32'h40000000: set EOR in end of descriptor ring and received segment data length(512-byte).
32'h000001FF | 32'h00000000: only set segment data length in non-end of descriptor ring.
BASE_SRAM_DRAM+16'hC000+16'h200*pkt_idx: Segment data pointer(SDP), point to the starting address of this received data segment.

5.18.6.2 TX Descriptor Initialization

Here use a testbench of rtl localsim for example. This example only configure some basic fields in tx descriptor, and all of one packet data is in one segment . Others like vlan informations and options of checksum offload, or case of packet in multiple (<=3) segments can also be configured for verification. Section 1.1.4.5 illustrate format and field descriptions of tx descriptor.



```
//---------------- Program Tx Descriptor ------------------------------
  pkt_len = 11'd60;
  for(pkt_idx = 0; pkt_idx < (NO_PKT + 1); pkt_idx = pkt_idx + 1) begin
    $display ("==== Config Tx Descriptor %0d", pkt_idx);
    //-- Config Tx Descriptor ----
    `sel_ahb_w4b(BASE_SRAM_DRAM+16'h0000+8'h10*pkt_idx, 32'h38000000+pkt_len+pkt_idx | ((pkt_idx == NO_PKT) << 31));
    `sel_ahb_w4b(BASE_SRAM_DRAM+16'h0004+8'h10*pkt_idx, BASE_SRAM_DRAM+16'h4000+16'h200*pkt_idx);
    `sel_ahb_w4b(BASE_SRAM_DRAM+16'h0008+8'h10*pkt_idx, 32'h00000000);
    `sel_ahb_w4b(BASE_SRAM_DRAM+16'h000C+8'h10*pkt_idx, 32'h00000000);
    $display("\n");
```

NO_PKT: number of descriptors in tx descriptor ring.
sel_ahb_w4b(address, wdata): task of write 4-byte data via AHB
BASE_SRAM_DRAM+16'h0000: address for start of tx descriptor ring.
8'h10*pkt_idx: space of one descriptor is 16-byte

32’h38000000+pkt_len+pkt_idx | (pkt_idx==NO_PKT1) <<31 :
Set FS=1, LS=1, INT=1, and transmitted segment data length(increment by pkt_idx). After initializing NO_PKT descriptors, set own bit in next descriptor to be ‘1’ (cpu own) to terminate tx simulation.
BASE_SRAM_DRAM+16’h4000+16’h200*pkt_idx : Segment data pointer(SDP), point to the starting address of this transmitted data segment.

5.18.6.3 Operated flow of RX descriptors

Step1: Initialize rx descriptor ring as section
Step2: Well configure NIC and set RX_EN, then start to receive packets.
Step3: CPU receive interrupt(FNRC_INT) from NIC when rx_dma write done a packet to memory. Then read descriptor (address of descriptor is increased by interrupt) to check own bit and FS/LS for received segment, and read packet information in descriptor (refer Table 5-12)
Step4: clear own bit and other fields in descriptor, assign new segment data pointer for further receiving packet.

5.18.6.4 Operated flow of TX descriptors

Step1: Initialize tx descriptor ring as per section 5.18.4.5

Step2: Correctly configure NIC and set TX_EN, then start to transmit packets.

Step3: CPU receive interrupt(TNTC_INT) from NIC when tx_dma read done a packet from memory. Then read descriptor (address of descriptor is increased by interrupt) to check own bit and FS/LS for transmitted segments (up to 3 segments for NIC design) ,

Step4: clear own bit and other fields in descriptor, assign new fields information and new segment data pointer for further transmitting packet which is ready in memory.

5.18.6.5 Hash Table Initialization

Step1: Set ht_bist_en =1 ,

Step2: Read ht_bist_done. If ht_bist_done=1 then go to next step.

Step3: select mode of hash table (0: 512-set, 1: 256-set)

Step4: configure hash table control register
hash_acc_cmd = 1
hash_bit_data
hash_bit_address
cmd_start = 1

Step5: read hash table control register and check cmd_start. If cmd_start = 0(write finish, clear by HW), then return to step4 for next bit data and address.
For Example of mode1,
MAC Address[47:0] = 0xBF579597BCEB. The CRC 32 Result is 0xFE96B8B6, and its result perform to reverse every bits. And the reverse bits of CRC 32 (rcrc) Result is 0x01694749.
The 8-bit hash value can be generated from

{rcrc[31],rcrc[30],rcrc[29],rcrc[28],rcrc[27],rcrc[2],rcrc[1],rcrc[0]}
As a result, Hash Value can be calculated to 0x01 by above bit combination.
Configure hash table as hash_bit_address = 0x01 and hash_bit_data = 1'b1, then NIC should accept destination MAC Address[47:0] = 0xBF579597BCEB.

5.18.6.6 TX Transmit LPI Sequence

Step 1: Set LPI_MODE_EN = 1 to enable LPI mode and set the LPI sleep threshold in LPI_SLEEP_TH.
Step 2: Let TX stay in IDLE state for a period longer than the LPI_SLEEP_TH by setting TX_EN=0 in TX_DMA Control Register.
Step 3: TX should starts sending LPI sequence.
Step 4: Set TX_EN=1 to wakeup TX.

5.18.6.7 RX Receive LPI Sequence

Step 1: Set LPI_MODE_EN = 1 to enable LPI mode.
Step 2: Send LPI sequence to the NIC. The NIC should assert the RX_PCODE_INT in the Interrupt Status Register.

5.18.6.8 WOL Mode

Step1: set WOL_PD = 1 to enter power down mode
Step2: when enter power down mode, HW start to detect magic packet. If magic packet is detected, assert interrupt.
Step3: CPU receive interrupt(MAGIC_PKT_REC), then wake up system.

5.18.6.9 References

- IEEE 802.3 CL4
- IEEE 802.3 CL22
- IEEE 802.3 CL35
- IEEE 802.3 CL45
- IEEE 802.3ae, IEEE 802.3 CL46
- IEEE 802.3ba, IEEE 802.3 CL81
- IEEE 802.3az
- IEEE 802.3 Annex 31A, IEEE 802.3Qbb
- RFC 2863, 3635, 2819, 3273

5.18.6.10 Register Definitions

For register details refer to chapter 3.18 in “MT8516A Application Processor Registers.”

Multimedia

6 Audio and Speech

6.1.1 Introduction

The audio system provides the audio data exchange ability. The interfaces are list as follows:

- Master/Slave I2S input interface with SRC x 1
- Master I2S output x 2
- Master I2S input x 1
- PCM/I2S merged interface for MTK connectivity IC x 1
- DIR(SPDIF-Input) x1
- SPDIF-Output x1
- Master TDM TX x1
- Master TDM RX x1

6.1.2 Features

The audio system is responsible for generating the following clock signals:

-
- **Audio playing**
  - Supports 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, and 48kHz sampling rate output
  - Supports playing stereo data
- **Audio recording**
  - Supports 8, 16, 32, 48kHz sampling rate recording
  - Supports stereo recording
- **Speech**
  - Supports dual MIC
  - Supports 8/16kHz sampling rate recording
  - Supports side tone filter
- **I2S**
  - Supports master/slave input mode
  - Supports master output mode
  - Supports 16/24-bit stereo data
  - Supports 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 88.2, 96, 176.4, and 192kHz sampling rate in master mode
  - Supports EIAJ/I2S format
  - Supports I2S input/output with the same sampling rate at the same time
  - Supports MCLK frequency range is 1.024~49.152MHz

- **PCM/I2S merged interface**
  - 4-pin interface for concurrently supporting I2S and PCM
  - PCM supports 8k/16k Hz sampling rate
  - I2S supports 32, 44.1, and 48 kHz sampling rate

- **DIR**
- supports SPDIF input decode
- supports 32, 44.1, 48, 88.2, and 96KHz sample rate

- **SPDIF-Out**
- supports SPDIF output encode
- supports 32, 44.1, 48, 88.2, and 96KHz sample rate

- **TDM TX**
- supports Time Division Multiplexer I2S output (master mode only)
- supports eight, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 88.2, 96, and 192KHz sample rate
- supports channel number up to 2/4/8 in configuration by 1/2/4 data pins (corresponding to 2/4/8 channels)
- dedicated pin for TDM TX (not share clock pins with TDM RX)
- pin share with DAC I2S Out ,only can use I2S1 or TDM TX at the same time, see Figure 1-1.

- **TDM RX**

- supports Time Division Multiplexer input
- supports 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 88.2, 96, and 192KHz sample rate
- supports channel number up to 2/4/8 in 1 serial data pin
- dedicated pin for TDM RX (not share clock pins with TDM TX)

- **Audio CODEC**
- Support 2-ch internal downlink (playback) audio CODEC, sample rate 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, and 48KHz
- support 2-ch internal uplink (record) audio CODEC, sample rate 8, 16, 32, 48KHz
- support 2-ch digital MIC uplink (record) audio CODEC, sample rate 8, 16, and 32KHz, sharing input pin with internal uplink record path

- The accessory detector (ACCDET) supports 2 types of external accessories, which are microphone and hook-switch mode
- 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 CPU

6.1.3 Audio System Block Diagram

The diagram below shows the flexibility on the interconnection between audio interfaces.

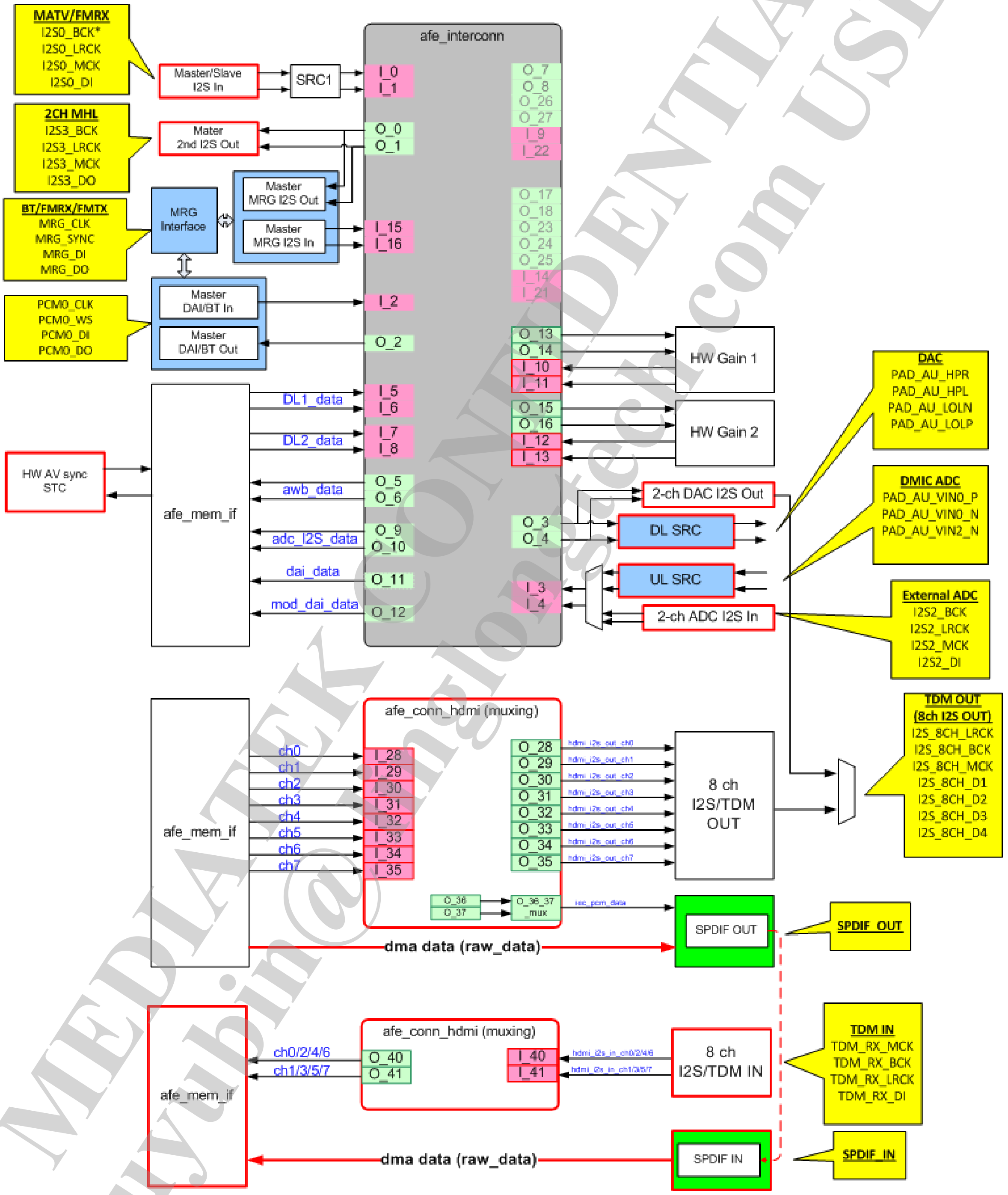
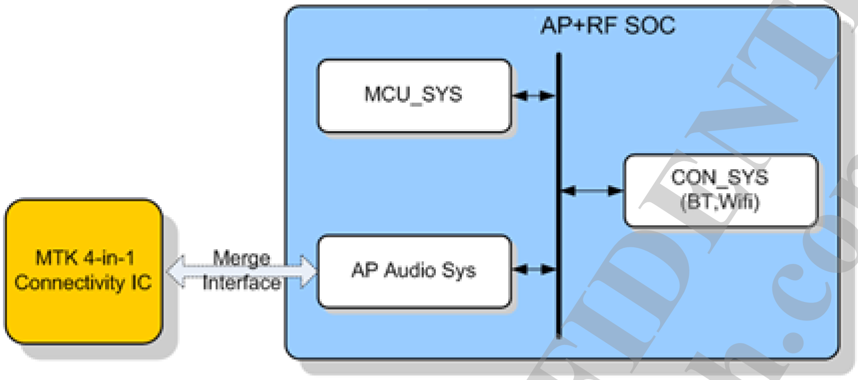


Figure 6-1. Audio System Block Diagram

6.1.4 Theory of Operations

The Audio system architecture, shown below, can connect to the MTK 4-in-1 (Wi-Fi/BT/FM/GPS) connectivity IC by merge interface, and the CPU of MCU_SYS can access Audio Sys memory and R/W Audio register to configure settings for different applications.



.

Figure 6-2. Audio System Overview

6.1.4.1 Time Division Multiplexed (TDM) Interface Overview

The TDM Interface is a digital multiplexing technique for combining several low-rate digital channels into one high-rate one. The MT8516A supports TDM IN/OUT Interface which provides a serial output of eight channels of audio data with sample rates up to 192 kHz within a single data stream.

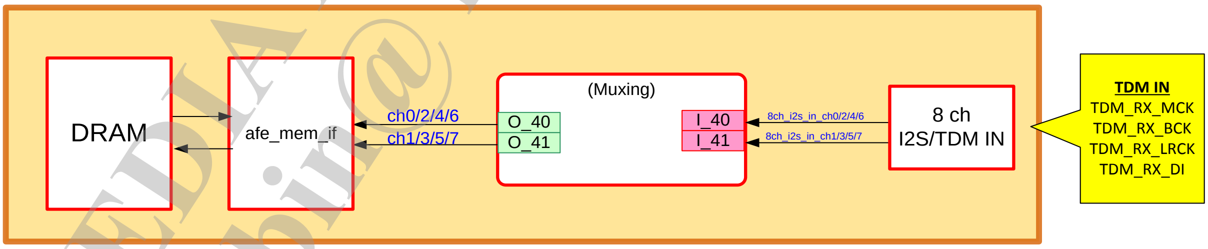


Figure 6-3. TDM IN Interface Overview

6.1.4.2 TDM IN Interface Specification

- Master output mode only.
- Programmable LRCK width, 1 BCK ~ N-1 BCK. Inverse or not.
- Channel width: 16 or 32 BCK cycles.

- Formats: I2S & EIAJ modes.
- Supports 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 88.2, 96, 192KHz sample rate.
- Dedicated pin for TDM input (not share clock pins with TDM OUT)
- Supports channel number up to 2/4/8 in 1 serial data pin.

6.1.4.3 TDM IN Data Path

In MT8516A , TDM IN interface has 4-pin (IN_DATA, OUT_BCK, OUT_LRCK, OUT_MCK) signal and can receive 8/4/2 channel data by 1-bit data pin, then transfer to paralle data(16/24 -bit) to SRAM/DRAM. The clk source comes from PLL pass through divider for TDM. TDM IN interface supports 2 different data formats. They are I2S and EIAJ format.

IN_DATA is received most significant bit (MSB) first, all data is valid on the rising edge of OUT_BCK. The IN_DATA MSB is transmitted early, but is guaranteed valid for a specified time after OUT_BCK rises. All other bits are transmitted on the falling edge of OUT_BCK.

OUT_LRCK identifies the start of a new frame and is equal to the sample rate, Fs. In I2S format, OUT_LRCK is sampled as valid on the rising OUT_BCK edge preceding the most significant bit of the first data sample and must be held valid for at least 1 OUT_BCK period. In EIAJ format, OUT_LRCK is delay 1cycle of OUT_BCK between I2S format.

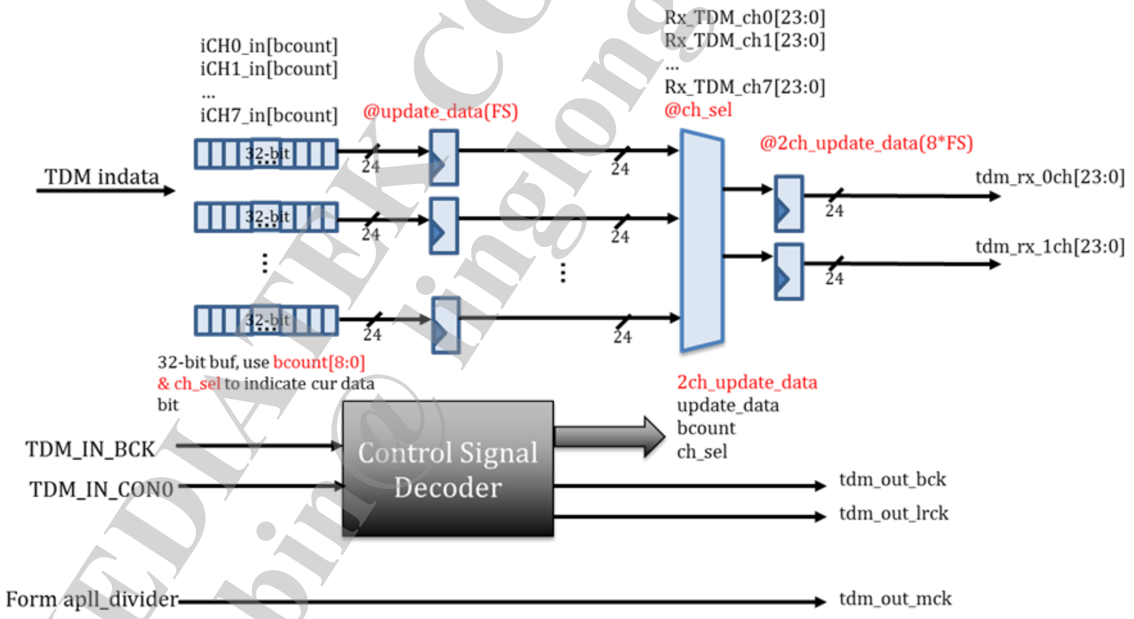


Figure 6-4. TDM IN Interface Data Path

The.Data Path of TDM IN interface shows data path which receives single data to buffer and arrangement to parallel data. When buffer is filling, send parallel data to memory. User should configure sample rate, channel width, LRCK width, format by seeking control register. In MT8516A supports 8-channel format to receive data, but only saves 6 channels into SRAM/DRAM by setting control reg. It can improve memory usage for software part.

After the audio system receives the data transferred from TDM, it will send the data to DRAM which is used as a ping-pong buffer. See figure below. The interrupt will be sent when the DMA system(afe_memif_if) writes half of buffer. After the CPU receives the interrupt, the written data in the buffer will be read out. The Steps are shown here:

1. After audio hardware receives data from TDM-in module, it starts to write received data to DRAM buffer A.

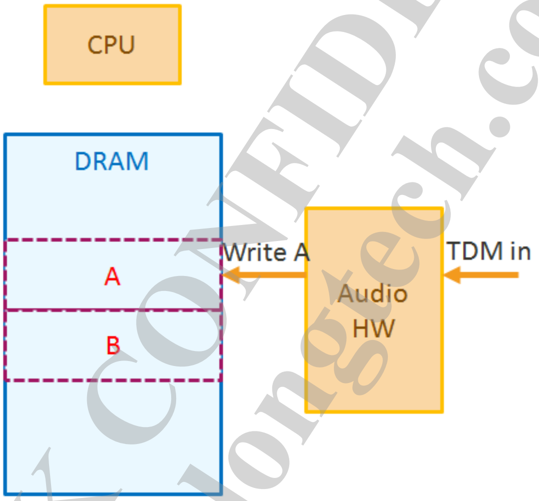


Figure 6-5. Step 1 for afe_memif_if Mechanism

2. While audio hardware starts to write DRAM buffer B, it sends an interrupt to CPU.

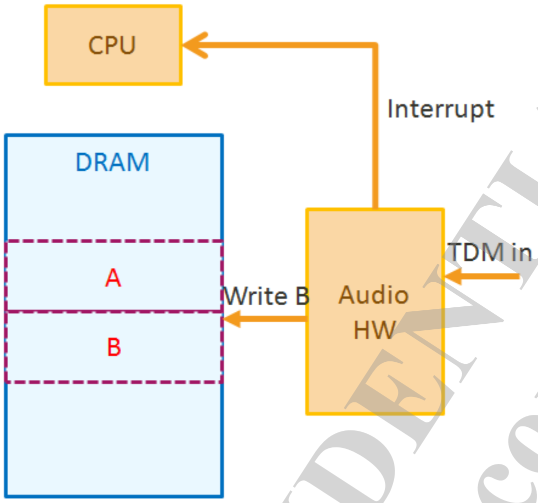


Figure 6-6. Step 2 for afe_memif_if Mechanism

3. CPU received the interrupt and then starts to read buffer A

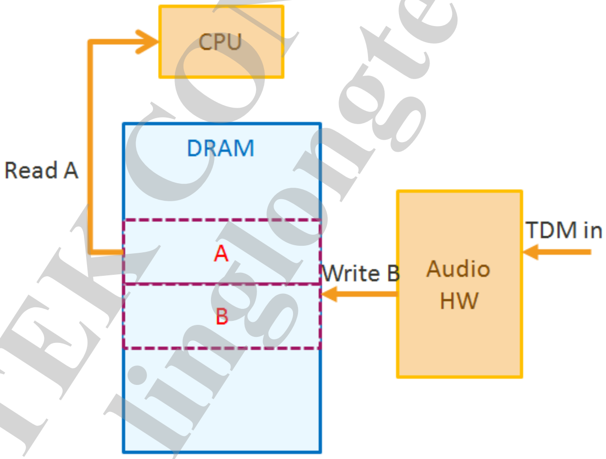


Figure 6-7. Step 3 for afe_memif_if Mechanism

4. After finished writing buffer B, audio hardware will turn back to write buffer A. In the meanwhile audio hardware also sends and interrupt to CPU starting read buffer B.

Afterward, repeat steps 1~4. This mechanism assures the data can be read continuously.

6.1.4.4 Digital Audio Interface Supported Formats

OUT_LRCK(FSYNC) : a frame synchronization
OUT_BCK : serial clock
IN_DATA : the serial data line

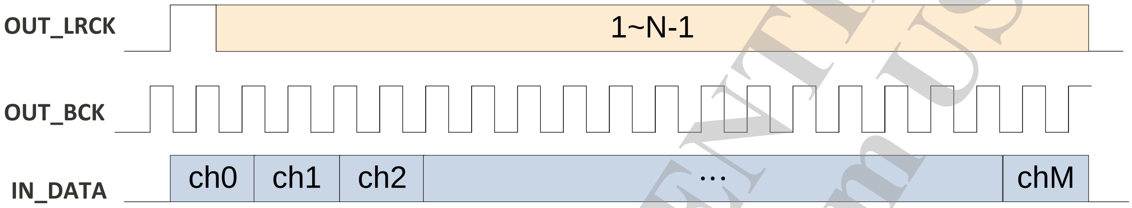


Figure 6-8. TDM IN Interface Signal

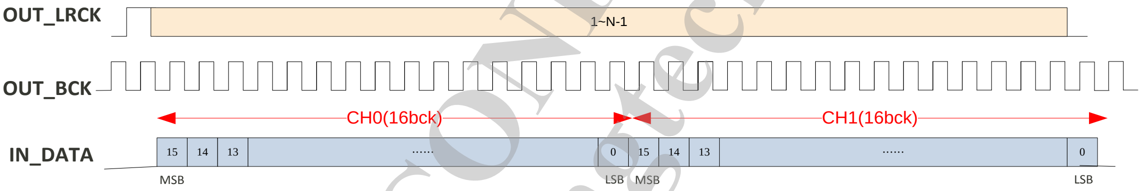


Figure 6-9. Wave Form of TDM IN Interface (2 ch, 16 bck, 16 bit, I2S format)

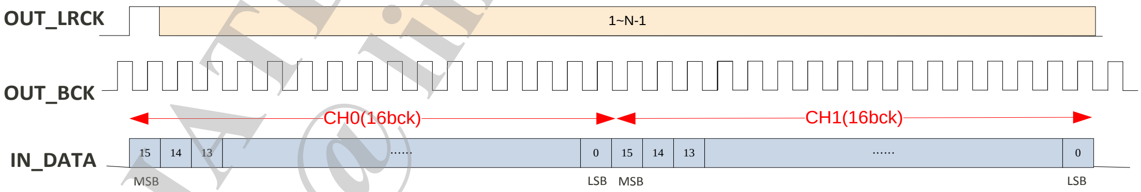


Figure 6-10. Wave Form of TDM IN Interface (2 ch, 16 bck, 16 bit, EIAJ format)

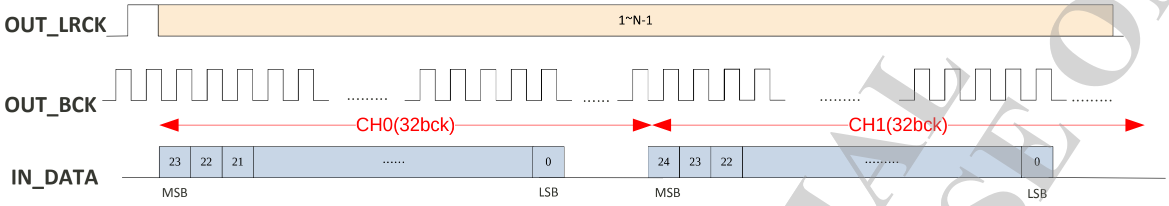


Figure 6-11. Wave Form of TDM IN Interface (2 ch, 32 bck, 24 bit, I2S format)

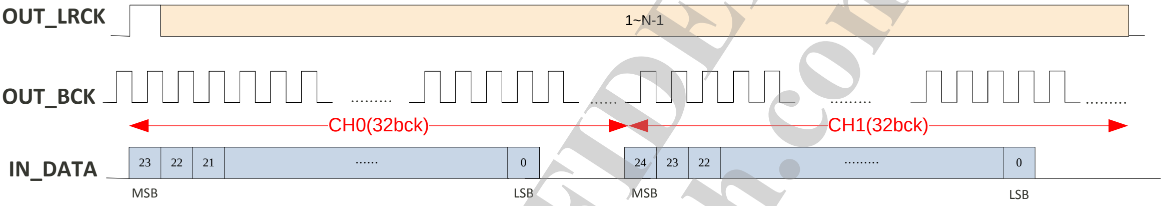


Figure 6-12. Wave Form of TDM IN Interface (2 ch, 32 bck, 24 bit, EIAJ format)

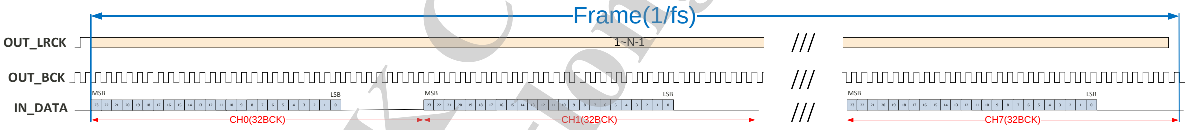


Figure 6-13. Wave Form of TDM IN Interface (8 ch, 32 bck, 24 bit, I2S format)

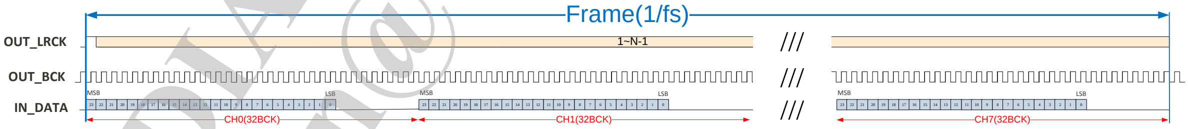


Figure 6-14. Wave Form of TDM IN Interface (8 ch, 32 bck, 24 bit, EIAJ format)

6.1.5 Register Definitions

For register details refer to chapter 4.1 of "MT8516A Application Processor Registers."

6.1.6 Programming Guide

6.1.6.1 UL and DL Path Power Off

The UL and DL path can be turned on/off through changing the value of ABB_AFE_CON0, AFE_ADDA_UL_DL_CON0 and AFE_DAC_CON0. See chapter 4.1 of "MT8516A Application Processor Register s." for these configurable registers.

Register Name	Register Address	Setting	HW Description	SW Description
8516B abb here set digital part				
ABB_AFE_CON0	0x0F00	0x0000	dl/ul turn off, [0]=dl_en, [1]=ul_en	
8516B afe here set digital part				
AFE_ADDA_UL_DL_CON0	0x0124	0x0000_0000	AP side adda global enable	
AFE_DAC_CON0	0x0010	0x0000_0000	AP side afe global enable	
Hereafter, it can be turned off completely.				

6.1.6.2 DL Path Audio Mode Power On

The table below provides the digital part settings step by step excluding analog part for DL Audio mode power on sequence. See chapter 4.1 of “MT8516A Application Processor Registers” for these configurable registers.

Table 6-1. Audio Downlink Turn On Procedure

Register Name	Register Address	Setting	HW Description	SW Description
8516B afe here set digital part				
AFE_ADDA_UL_SRC_CON0	0x0114	0x001E_0001	UL sample rate [20:19]=[17:18]=voice_mode : 0 : 8K, 1 : 16K, 2 : 32K, 3 : 48K [0]=ul_src_on here is 48K example	ul FS and on
AFE_ADDA_UL_SRC_CON1	0x0118	0x0000_0000	UL receiving path muxing : [27] c_dac_en_ctl : 0: Uplink comb filter data path output. 1: Sine table output.	ul muxing double check
AFE_SINEGEN_CON0	0x01F0	0x942E_A2E5	12K example, 1.5KHZ tone. [31:28] c_inner_loopback_mode=9. [26]=1=sinetable_on [25]=[24]=0 sine_mute for ch1/ch2 [23:20]=2=ch2 sine_1x_en for 12k [19:17]=7=ch2 amp 0db [16:12]=A=ch2 1.875KHZ for 12K sample rate [11:8]=2=ch1 sine_1x_en for 12K [7:5]=7=ch1 amp 0db [4:0]=5=ch1 0.9375KHZ for 12K sample rate	sinetable spec and sinetable muxing instead of input from i03/i04
AFE_ADDA_DL_SRC2_CON0	0x0108	0x0300_1803	[31:28] dl_input_mode [5]=voice_mode=0 [1]=gain_on [0]=dl_on	dl FS and dl on and dl gain
AFE_ADDA_TOP_CON0	0x0120	0x0000_0000	[15:12]=c_loop_back_mode_ctl=4 for adda sinetable source [0]=c_ext_adc_ctl	sinetable muxing double confirm
AFE_I2S_CON1	0x0034	0x0000_000B	[11:8]=I2S_OUT FS = 0 for 8K. [3]=1 I2S format [1]=1 32bit [0] i2s_on	I2S OUT FS and i2s_on
AFE_ADDA_UL_DL_CON0	0x0124	0x0000_0001	AP side adda global enable	
AFE_DAC_CON0	0x0010	0x0000_0001	AP side afe global enable	
8516B abb here set digital part				

Register Name	Register Address	Setting	HW Description	SW Description
ABB_AFE_CON1	0x0F04	0x000A	[3:0] dl_rate : 8K*8 [4] ul_rate : 64K	DL, UL sample rate
ABB_AFE_CON2	0x0F08	0x000a	[11] lch_mute (HQA xtalk testing) [10] rch_mute (HQA xtalk testing) [8] ul_lr_swap [3] dl_ul_lpbk=1 [1] dl_sine_on=1	dl sine table on and dl->ul loopback on
ABB_AFE_CON8	0x0F20	0x9010	[15] sine_on [12]=sine_dl_en [11:4]=sine_freq [3:0]=sine amp, 0 for 0dB	sinetable spec
ABB_AFE_CON11	0x0F2C	0x0100	[9] = dc compensation update take effect [8] = top control update for ul/dl rate take effect.	just for trigger
ABB_AFE_CON11	0x0F2C	0x0000	inverse for update rate	just for trigger
ABB_AFE_CON0	0x0F00	0x0003	dl/ul turn on	
ABB_AFE_CON2	0x0F08	0x0008	[3] dl_ul_lpbk=1 [1] dl_sine_on=0	choose from AP sinetable spec
Special for DC compensation register				
ABB_AFE_CON3	0x0F0C	0x0000	15:0 lch_dccomp_val L-ch DC compensation value. S0.15	L-ch DC offset
ABB_AFE_CON4	0x0F10	0x0000	15:0 rch_dccomp_val R-ch DC compensation value. S0.15	R-ch DC offset
ABB_AFE_CON10	0x0F28	0x0001	0 dccomp_en DC compensation enable sequence.	R-ch DC offset
ABB_AFE_CON0	0x0F28	0x0001	downlink turn on	

8516B abb here set digital part				
ABB_AFE_CON0	0x0F00	0x0000	dl/ul turn off, [0]=dl_en, [1]=ul_en	
AFE_ADDA_UL_DL_CON0	0x0124	0x0000_0000	AP side adda global enable	
AFE_DAC_CON0	0x0010	0x0000_0000	AP side afe global enable	
Hereafter, it can be turned off completely.				

6.1.6.3 DL Path Voice Mode Power On

The table below provides the digital part settings step by step excluding analog part for DL Voice mode power on sequence. See Chapter 4.1 of “MT8516A Application Processor Registers” for these configurable registers.

Table 6-2. Voice Downlink turn On Procedure

Register Name	Register Address	Setting	HW Description	SW Description
8516B afe here set digital part				
AFE_ADDA_UL_SRC_CON0	0x0114	0x0014_0001	UL sample rate [20:19]=[17:18]=voice_mode : 0 : 8K, 1 : 16K, 2 : 32K, 3 : 48K [0]=ul_src_on here is 16K example	ul FS and on
AFE_ADDA_UL_SRC_CON1	0x0118	0x0000_0000	UL receiving path muxing : [27] c_dac_en_ctl : 0: Uplink comb filter data path output. 1: Sine table output.	ul muxing double check
AFE_ADDA_DL_SRC2_CON0	0x0108	0x0300_1823	[31:28] dl_input_mode [5]=voice_mode=1 [1]=gain_on [0]=dl_on	dl FS and dl on and dl gain
AFE_ADDA_TOP_CON0	0x0120	0x0000_0000	[15:12]=c_loop_back_mode_ctl=4 for adda sinetable source [0]=c_ext_adc_ctl	sinetable muxing double confirm
AFE_I2S_CON1	0x0034	0x0000_040B	[11:8]=I2S_OUT FS = 4 for 16K. [3]=1 I2S format [1]=1 32bit [0] i2s_on	I2S OUT FS and i2s_on
AFE_ADDA_UL_DL_CON0	0x0124	0x0000_0001	AP side adda global enable	
AFE_DAC_CON0	0x0010	0x0000_0001	AP side afe global enable	
8516B abb here set digital part				
ABB_AFE_CON1	0x0F04	0x0004	[3:0] dl_rate : 16K*8 [4] ul_rate : 64K	DL, UL sample rate
ABB_AFE_CON2	0x0F08	0x000a	[11] lch_mute (HQA xtalk testing) [10] rch_mute (HQA xtalk testing) [8] ul_lr_swap [3] dl_ul_lpbk=1 [1] dl_sine_on=1	dl sine table on and dl->ul loopback on
ABB_AFE_CON8	0x0F20	0x9010	[15] sine_on [12]=sine_dl_en [11:4]=sine_freq [3:0]=sine amp, 0 for 0dB	6323 sinetable spec

Register Name	Register Address	Setting	HW Description	SW Description
ABB_AFE_CON11	0x0F2C	0x0100	[9] = dc compensation update take effect [8] = top control update for ul/dl rate take effect.	just for trigger
ABB_AFE_CON11	0x0F2C	0x0000	inverse for update rate	just for trigger
ABB_AFE_CON0	0x0F00	0x0003	dl/ul turn on	

8516B abb here set digital part				
ABB_AFE_CON0	0x0F00	0x0000	dl/ul turn off, [0]=dl_en, [1]=ul_en	
8516B afe here set digital part				
AFE_ADDA_UL_DL_CON0	0x0124	0x0000_0000	AP side adda global enable	
AFE_DAC_CON0	0x0010	0x0000_0000	AP side afe global enable	
Hereafter, it can be turned off completely.				

6.1.6.4 UL Path Power On

There are digital part settings step by step excluding analog part for UL power on sequence. See chapter 4.1 of "MT8516A Application Processor Registers" for these configurable registers.

Table 6-3. Digital Part AFE Initialization

Register Name	Register Address	Setting	HW Description	SW Description
8516B afe here set digital part				
AFE_ADDA_UL_SRC_CON0	0x0114	0x001E_0001	UL sample rate [20:19]=[17:18]=voice_mode : 0 : 8K, 1 : 16K, 2 : 32K, 3 : 48K [0]=ul_src_on here is 48K example	ul FS and on
AFE_ADDA_UL_SRC_CON1	0x0118	0x0000_0000	UL receiving path muxing : [27] c_dac_en_ctl : 0: Uplink comb filter data path output. 1: Sine table output.	ul muxing double check
AFE_ADDA_TOP_CON0	0x0120	0x0000_0000	[15:12]=c_loop_back_mode_ctl=4 for adda sinetable source [0]=c_ext_adc_ctl	sinetable muxing double confirm
AFE_ADDA_UL_DL_CON0	0x0124	0x0000_0001	AP side adda global enable	
AFE_DAC_CON0	0x0010	0x0000_0001	AP side afe global enable	

Table 6-4. Voice Uplink Turn On Procedure

Register Name	Register Address	Setting	HW Description	SW Description
8516B abb digital part				
ABB_AFE_CON1	0x0F04	0x0010	CIC out = 96K (for 48K record)	
ABB_AFE_CON11	0x0F2C	0x0100	trigger	
ABB_AFE_CON11	0x0F2C	0x0000	inverse for update rate	
ABB_AFE_CON0	0x0F00	0x0002	uplink turn on	

7 Wi-Fi/ Bluetooth Connectivity

7.1 Introduction

MT8516A embodies wireless communication device, including WLAN, Bluetooth. With advanced radio technologies integrated into one single chip, MT8516A provides the best and most convenient connectivity solution among the industry. Advanced and sophisticated radio coexistence algorithms and hardware mechanisms are implemented with-in. It also supports single antenna sharing among 2.4 GHz antenna for Bluetooth and WLAN

7.2 Features

- Wi-Fi
  - Single-band (2.4GHz) single stream 802.11 b/g/n MAC/BB/RF
  - 802.11 d/h/k compliant
  - Security: WFA WPA/WPA2 personal, WPS2.0, WAPI (Hardware)
  - QoS: WFA WMM, WMM PS
  - Supports 802.11n optional features: STBC, A-MPDU, Blk-Ack, RIFS, MCS feedback, 20/40MHz coexistence (PCO), unscheduled PSMP
  - Supports 802.11w protected managed frames
  - Supports Wi-Fi Direct (WFA P-2-P standard)
  - Supports HotSpot 2.0 Passpoint
  - Per packet TX power control
- Bluetooth
  - Bluetooth specification v2.1+EDR
  - Bluetooth specification 3.0+HS compliance
  - Bluetooth v4.0 Low Energy (LE)
  - Rx sensitivity: GFSK -95dBm, DQPSK -94dBm, 8-DPSK -88dBm
  - Best-in-class BT/Wi-Fi coexistence performance
  - Up to 4 piconets simultaneously with background inquiry/page scan
  - Supports Scatternet
  - Packet Loss Concealment (PLC) function for better voice quality
  - Low-power scan function to reduce power consumption in scan modes

7.3 Connectivity System Block Diagram

The architecture and core blocks of the Connectivity system are shown in Figure 7-1, including the following parts: MCUSYS, WFSYS, BTSYS.

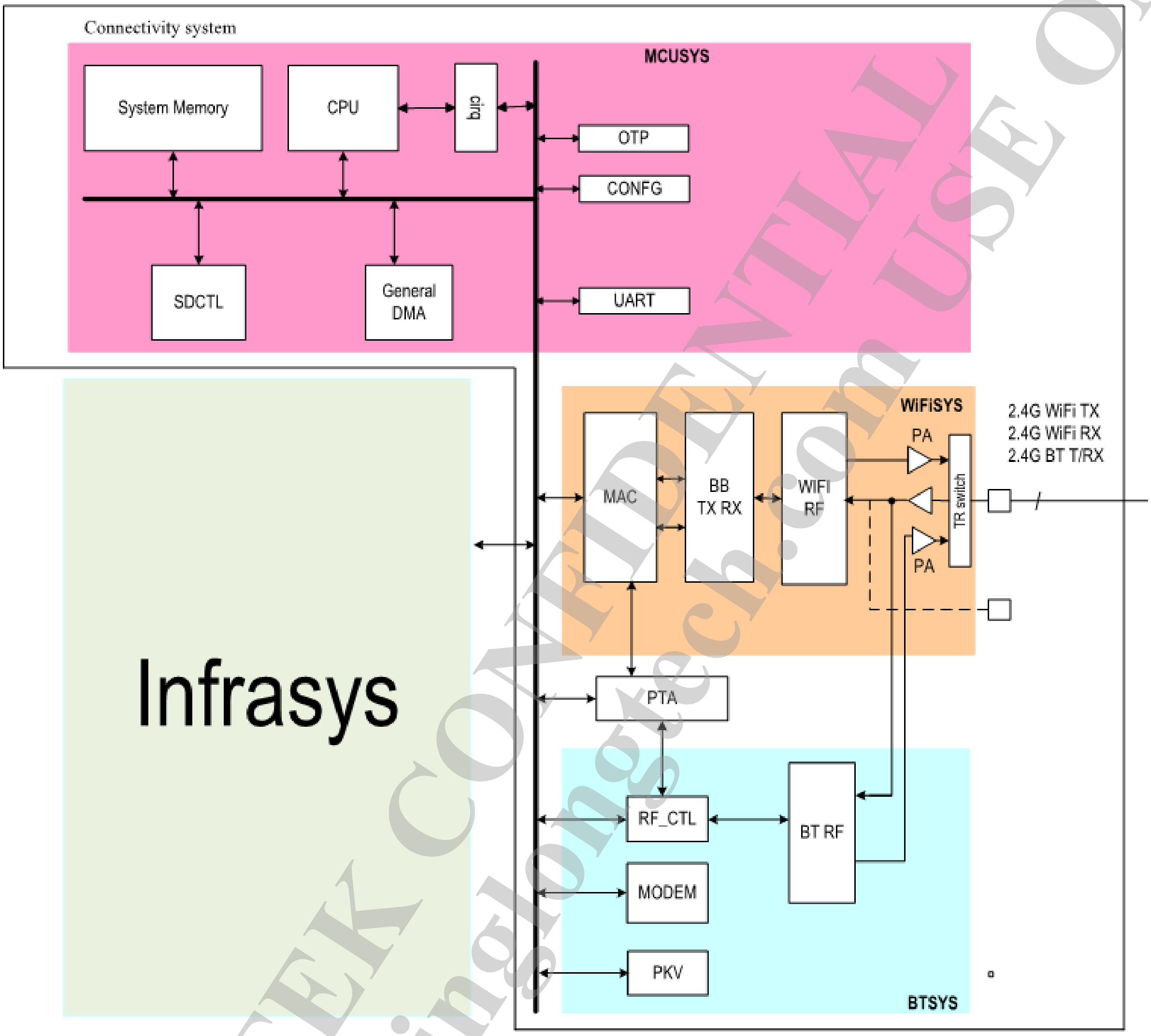


Figure 7-1. Connectivity System Block Diagram

.

7.4 Programming Guide

The Wi-Fi/Bluetooth connectivity system is programmed only by AP internal API, WMT(Wireless Management Task). Therefore customer doesn't need to program connectivity system individually.

7.5 Clocks

Connsys internal mcu bus clock is 138MHz.