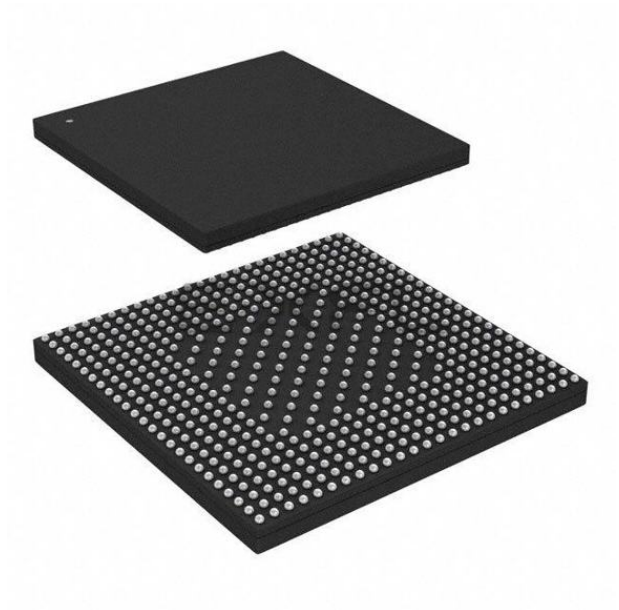


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	ARM® Cortex®-A7, ARM® Cortex®-M4
Number of Cores/Bus Width	2 Core, 32-Bit
Speed	1.0GHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR2, LPDDR3, DDR3, DDR3L
Graphics Acceleration	No
Display & Interface Controllers	Keypad, LCD, MIPI
Ethernet	10/100/1000Mbps (2)
SATA	-
USB	USB 2.0 + PHY (1), USB 2.0 OTG + PHY (2)
Voltage - I/O	1.8V, 3.3V
Operating Temperature	-20°C ~ 105°C (TJ)
Security Features	A-HAB, ARM TZ, CAAM, CSU, SJC, SNVS
Package / Case	488-TFBGA
Supplier Device Package	488-TFBGA (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcimx7d3evk10sc

MU_ATR3 field descriptions

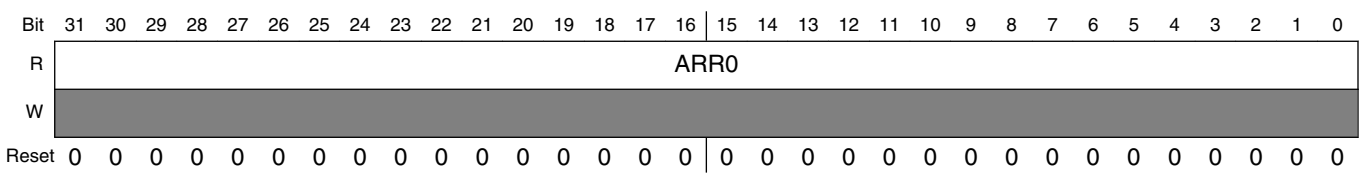
Field	Description
ATR3	Processor A Transmit Register 3. (Write-only) • Data written to the ATR3 register is reflected on the Processor B-side in the Processor B Receive Register 3 (BRR3). The ATR3 and BRR3 registers are not double-buffered—a write to the ATR3 register overrides the data readable at the BRR3 register. • A write to the transmit register clears a “transmitter empty” bit (TE3) in the Processor A Status Register (ASR) on the transmitter side, and sets a “receiver full” bit (RF3) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 3 on the Processor B-side). • Any write to the ATR3 register will update all status information.

4.3.5.5 Processor A Receive Register 0 (MU_ARR0)

Use Processor A Receive Register 0 (ARR0, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the BTR0 register is immediately reflected in the ARR0 register.
- You can only read the ARR0 register when the RF0 bit in the ASR register is set to “1”.
- Writing to the ARR0 register generates an error response to the Processor A.

Address: 30AA_0000h base + 10h offset = 30AA_0010h



MU_ARR0 field descriptions

Field	Description
ARR0	Processor A Receive Register 0. (Read-only) • Reflects the data written to Processor B Transmit Register 0 (BTR0). • Reading the ARR0 register clears the “receiver full” bit (RF0) in the Processor A Status Register (ASR) on the receiver side, and sets the “transmitter empty” bit (TE0) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 0 on the Processor B-side). • Any read of the ARR0 register will update all status information.

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9DB0	Pre Divider Register (CCM_PRE59)	32	R/W	0000_0000h	5.2.8.22/ 733
3038_9DB4	Pre Divider Register (CCM_PRE_ROOT59_SET)	32	R/W	0000_0000h	5.2.8.23/ 736
3038_9DB8	Pre Divider Register (CCM_PRE_ROOT59_CLR)	32	R/W	0000_0000h	5.2.8.24/ 739
3038_9DBC	Pre Divider Register (CCM_PRE_ROOT59_TOG)	32	R/W	0000_0000h	5.2.8.25/ 742
3038_9DF0	Access Control Register (CCM_ACCESS_CTRL59)	32	R/W	0000_0000h	5.2.8.26/ 745
3038_9DF4	Access Control Register (CCM_ACCESS_CTRL_ROOT59_SET)	32	R/W	0000_0000h	5.2.8.27/ 747
3038_9DF8	Access Control Register (CCM_ACCESS_CTRL_ROOT59_CLR)	32	R/W	0000_0000h	5.2.8.28/ 750
3038_9DFC	Access Control Register (CCM_ACCESS_CTRL_ROOT59_TOG)	32	R/W	0000_0000h	5.2.8.29/ 752
3038_9E00	Target Register (CCM_TARGET_ROOT60)	32	R/W	0000_0000h	5.2.8.10/ 709
3038_9E04	Target Register (CCM_TARGET_ROOT60_SET)	32	R/W	0000_0000h	5.2.8.11/ 711
3038_9E08	Target Register (CCM_TARGET_ROOT60_CLR)	32	R/W	0000_0000h	5.2.8.12/ 713
3038_9E0C	Target Register (CCM_TARGET_ROOT60_TOG)	32	R/W	0000_0000h	5.2.8.13/ 715
3038_9E10	Miscellaneous Register (CCM_MISC60)	32	R/W	0000_0000h	5.2.8.14/ 717
3038_9E14	Miscellaneous Register (CCM_MISC_ROOT60_SET)	32	R/W	0000_0000h	5.2.8.15/ 718
3038_9E18	Miscellaneous Register (CCM_MISC_ROOT60_CLR)	32	R/W	0000_0000h	5.2.8.16/ 719
3038_9E1C	Miscellaneous Register (CCM_MISC_ROOT60_TOG)	32	R/W	0000_0000h	5.2.8.17/ 720
3038_9E20	Post Divider Register (CCM_POST60)	32	R/W	0000_0000h	5.2.8.18/ 721
3038_9E24	Post Divider Register (CCM_POST_ROOT60_SET)	32	R/W	0000_0000h	5.2.8.19/ 724
3038_9E28	Post Divider Register (CCM_POST_ROOT60_CLR)	32	R/W	0000_0000h	5.2.8.20/ 727
3038_9E2C	Post Divider Register (CCM_POST_ROOT60_TOG)	32	R/W	0000_0000h	5.2.8.21/ 730
3038_9E30	Pre Divider Register (CCM_PRE60)	32	R/W	0000_0000h	5.2.8.22/ 733
3038_9E34	Pre Divider Register (CCM_PRE_ROOT60_SET)	32	R/W	0000_0000h	5.2.8.23/ 736

Table continues on the next page...

OCOTP_TIMING field descriptions (continued)

Field	Description
PROG	This count value specifies the strobe period in one time write OTP. tPRW = (PROG - 1) / IPG_CLK_FREQ. It is given in number of IPG_CLK periods.

6.4.5.3 OTP Controller Write Data Register (OCOTP_DATA0)

This register associates with HW_OCOTP_CTRL to perform one-time writes to the OTP. Please see the "Software Write Sequence" section for operating details.

EXAMPLE

```
Empty Example.
```

Address: 3035_0000h base + 20h offset = 3035_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_DATA0 field descriptions

Field	Description
DATA0	The program data for first fuse word in one 128 bits OTP. It is used to initiate a write to OTP. Please see the "Software Write Sequence" section for operating details.

6.4.5.4 OTP Controller Write Data Register (OCOTP_DATA1)

This register associates with HW_OCOTP_CTRL to perform one-time writes to the OTP. Please see the "Software Write Sequence" section for operating details.

EXAMPLE

```
Empty Example.
```

Address: 3035_0000h base + 30h offset = 3035_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_DATA1 field descriptions

Field	Description
DATA1	The program data for second fuse word in one 128 bits OTP.

6.6.5.3.3 SD, eSD and SDXC

After the normal boot mode initialization begins, the SD/eSD/SDXC frequency is set to 347.22 kHz. During the identification phase, SD/eSD/SDXC card voltage validation is performed. During voltage validation, boot code first checks with high voltage settings; if that fails, it checks with low voltage settings.

The capacity of the card is also checked. Boot code supports high capacity and low capacity SD/eSD/SDXC cards after voltage validation card initialization is done.

During card initialization, the ROM boot code attempts to set the boot partition for all SD, eSD, and SDXC devices. If this fails, the boot code assumes the card is a normal SD card or SDXC card. If it does not fail, the boot code assumes it is an eSD card. After the initialization phase is over, boot code switches to a higher frequency (25 MHz in Normal Speed mode or 50 MHz in High Speed Mode). ROM also supports FAST_BOOT mode booting from eSD card. This mode can be selected by BOOT_CFG[4] (Fast Boot) fuse described in Table 6-41.

For UHSI cards, clock speed fuses can be set to SDR50 or SDR104 on USDHC1, USDHC2, USDHC3 and USDHC4 ports. This will enable the voltage switch process to set the signaling voltage to 1.8V during voltage validation. The bus width is fixed at 4 bits wide and a sampling point tuning process is needed to calibrate the number of delay cells. If SD Loopback Clock eFuse is set, the feedback clock will come directly from the loopback SD clock, instead of the card clock (by default). The SD clock speed can be selected by BOOT_CFG[3:2], and the SD Loopback Clock is selected by BOOT_CFG[0].

UHSI calibration start value (MMC_DLL_DLY[6:0]) and step value can be set to optimize the sample point tuning process.

If SD Power Cycle Enable eFuse is 1, ROM will set SD_RST pad low, wait 5ms and then set SD_RST pad high. If SD_RST pad is connected to SD power supply enable logic on board, it enables power cycle of SD card. This may be crucial in case when SD logic is in 1.8V states and must be reset to 3.3V states.

SDR50 and SDR104 boot are not supported on the USDHC1 and USDHC2 ports because there are no reset signals for those ports when connected in the IOMUX.

6.6.5.3.4 IOMUX Configuration for SD/MMC

Table 6-44. SD/MMC IOMUX Pin Configuration

Signal	USDHC1	USDHC2	USDHC3
CLK	SD1_CLK.alt0	SD2_CLK.alt0	SD3_CLK.alt0

Table continues on the next page...

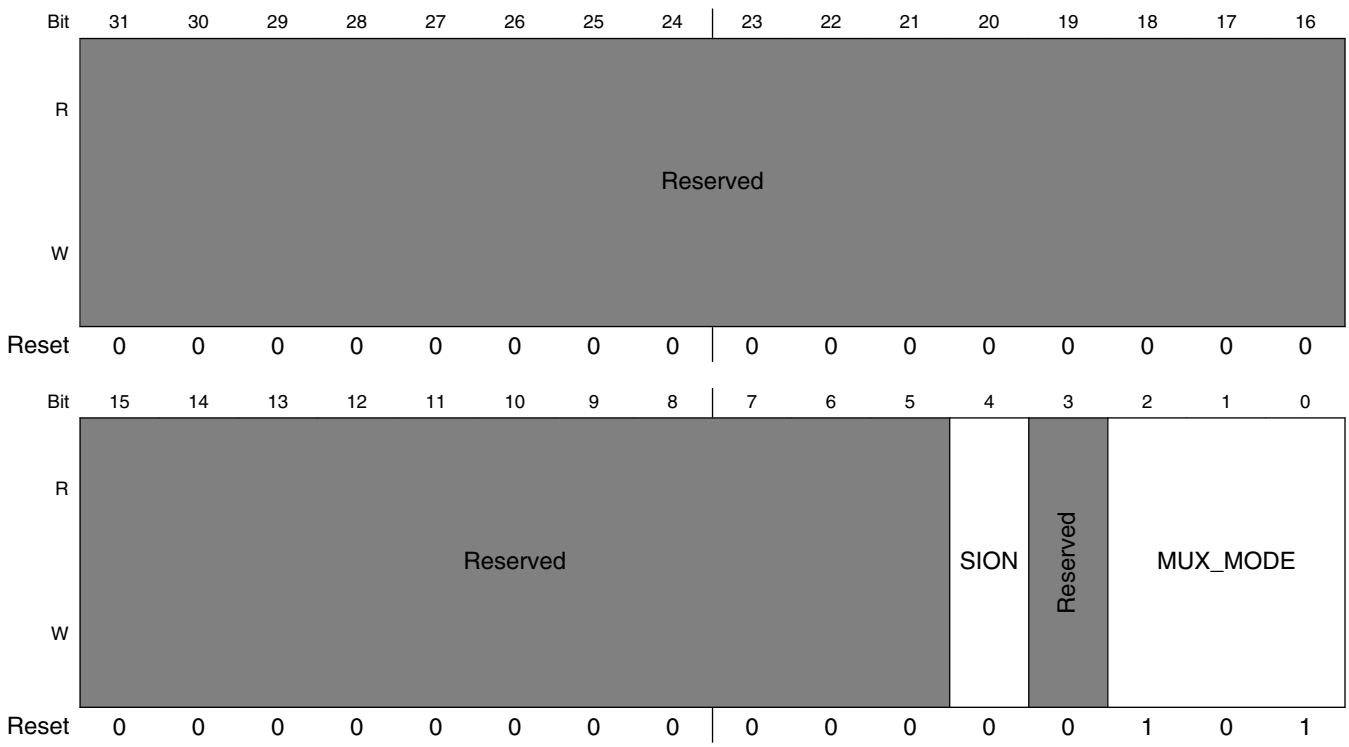
IOMUXC_SW_MUX_CTL_PAD_ENET1_RGMII_TD2 field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad ENET1_RGMII_TD2 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: ENET1_RGMII_TD2. 000 **ALT0_ENET1_RGMII_TD2** — Select mux mode: ALT0 mux port: RGMII_TD2 of instance: ENET1 001 **ALT1_FLEXCAN2_RX** — Select mux mode: ALT1 mux port: RX of instance: FLEXCAN2 010 **ALT2_ECSPI2_MISO** — Select mux mode: ALT2 mux port: MISO of instance: ECSPI2 011 **ALT3_I2C4_SCL** — Select mux mode: ALT3 mux port: SCL of instance: I2C4 100 **ALT4_EPDC_SDOED** — Select mux mode: ALT4 mux port: SDOED of instance: EPDC 101 **ALT5_GPIO7_IO8** — Select mux mode: ALT5 mux port: IO8 of instance: GPIO7

8.2.7.144 SW_MUX_CTL_PAD_ENET1_RGMII_TD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET1_RGMII_TD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 250h offset = 3033_0250h



8.2.7.321 CSI_DATA9_SELECT_INPUT DAISY Register (IOMUXC_CSI_DATA9_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 514h offset = 3033_0514h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CSI_DATA9_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **LCD_DATA08_ALT3** — Selecting Pad: LCD_DATA08 Mode: ALT3 for CSI_DATA9 1 **ECSPI2_SS0_ALT3** — Selecting Pad: ECSPI2_SS0 Mode: ALT3 for CSI_DATA9

8.3.4.3 Interrupt Control Unit

In addition to the general-purpose input/output function, the edge-detect logic in the GPIO peripheral reflects whether a transition has occurred on a given GPIO signal that is configured as an input (GDIR bit = 0). The interrupt control registers (GPIO_ICR1 and GPIO_ICR2) may be used to independently configure the interrupt condition of each input signal (low-to-high transition, high-to-low transition, low, or high). For information about GPIO_ICR1 and GPIO_ICR2 settings, see GPIO Memory Map/Register Definition.

The interrupt control unit is built of 32 interrupt control subunits, where each subunit handles a single interrupt line.

8.3.5 GPIO Memory Map/Register Definition

There are eight 32-bit GPIO registers. All registers are accessible from the IP interface. Only 32-bit access is supported.

The GPIO memory map is shown in the following table.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3020_0000	GPIO data register (GPIO1_DR)	32	R/W	0000_0000h	8.3.5.1/ 2123
3020_0004	GPIO direction register (GPIO1_GDIR)	32	R/W	0000_0000h	8.3.5.2/ 2124
3020_0008	GPIO pad status register (GPIO1_PSR)	32	R	0000_0000h	8.3.5.3/ 2124
3020_000C	GPIO interrupt configuration register1 (GPIO1_ICR1)	32	R/W	0000_0000h	8.3.5.4/ 2125
3020_0010	GPIO interrupt configuration register2 (GPIO1_ICR2)	32	R/W	0000_0000h	8.3.5.5/ 2129
3020_0014	GPIO interrupt mask register (GPIO1_IMR)	32	R/W	0000_0000h	8.3.5.6/ 2132
3020_0018	GPIO interrupt status register (GPIO1_ISR)	32	w1c	0000_0000h	8.3.5.7/ 2133
3020_001C	GPIO edge select register (GPIO1_EDGE_SEL)	32	R/W	0000_0000h	8.3.5.8/ 2134

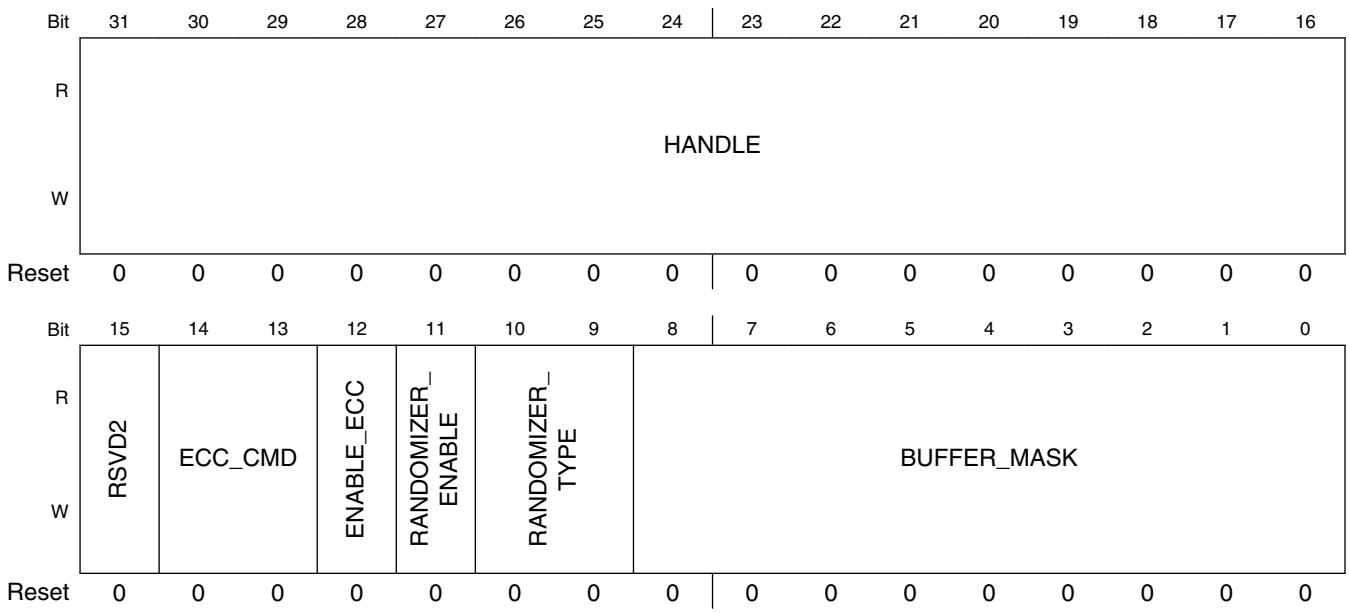
GPMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_2060	GPMI Control Register 1 Description (GPMI_CTRL1)	32	R/W	0004_0004h	9.6.6.7/ 2490
3300_2064	GPMI Control Register 1 Description (GPMI_CTRL1_SET)	32	R/W	0004_0004h	9.6.6.7/ 2490
3300_2068	GPMI Control Register 1 Description (GPMI_CTRL1_CLR)	32	R/W	0004_0004h	9.6.6.7/ 2490
3300_206C	GPMI Control Register 1 Description (GPMI_CTRL1_TOG)	32	R/W	0004_0004h	9.6.6.7/ 2490
3300_2070	GPMI Timing Register 0 Description (GPMI_TIMING0)	32	R/W	0001_0203h	9.6.6.8/ 2493
3300_2080	GPMI Timing Register 1 Description (GPMI_TIMING1)	32	R/W	0000_0000h	9.6.6.9/ 2493
3300_2090	GPMI Timing Register 2 Description (GPMI_TIMING2)	32	R/W	0302_3336h	9.6.6.10/ 2494
3300_20A0	GPMI DMA Data Transfer Register Description (GPMI_DATA)	32	R/W	0000_0000h	9.6.6.11/ 2495
3300_20B0	GPMI Status Register Description (GPMI_STAT)	32	R	0000_0005h	9.6.6.12/ 2495
3300_20C0	GPMI Debug Information Register Description (GPMI_DEBUG)	32	R	0000_0000h	9.6.6.13/ 2498
3300_20D0	GPMI Version Register Description (GPMI_VERSION)	32	R	0502_0000h	9.6.6.14/ 2499
3300_20E0	GPMI Debug2 Information Register Description (GPMI_DEBUG2)	32	R/W	0000_F100h	9.6.6.15/ 2499
3300_20F0	GPMI Debug3 Information Register Description (GPMI_DEBUG3)	32	R	0000_0000h	9.6.6.16/ 2502
3300_2100	GPMI Double Rate Read DLL Control Register Description (GPMI_READ_DDR_DLL_CTRL)	32	R/W	0000_0038h	9.6.6.17/ 2503
3300_2110	GPMI Double Rate Write DLL Control Register Description (GPMI_WRITE_DDR_DLL_CTRL)	32	R/W	0000_0038h	9.6.6.18/ 2504
3300_2120	GPMI Double Rate Read DLL Status Register Description (GPMI_READ_DDR_DLL_STS)	32	R	0000_0000h	9.6.6.19/ 2506
3300_2130	GPMI Double Rate Write DLL Status Register Description (GPMI_WRITE_DDR_DLL_STS)	32	R	0000_0000h	9.6.6.20/ 2507

9.6.6.3 GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL*n*)

The GPMI ECC control register handles configuration of the integrated ECC / Randomizer accelerator.

Address: 3300_2000h base + 20h offset + (4d × i), where i=0d to 3d



GPMI_ECCCTRL*n* field descriptions

Field	Description
31–16 HANDLE	This is a register available to software to attach an identifier to a transaction in progress. This handle will be available from the ECC register space when the completion interrupt occurs.
15 RSVD2	Always write zeroes to this bit field.
14–13 ECC_CMD	ECC Command information. DECODE = 0x0 Decode. ENCODE = 0x1 Encode. RESERVE2 = 0x2 Reserved. RESERVE3 = 0x3 Reserved.
12 ENABLE_ECC	Enable ECC processing of GPMI transfers. ENABLE = 0x1 Use integrated ECC for read and write transfers. DISABLE = 0x0 Integrated ECC remains in idle.
11 RANDOMIZER_ ENABLE	Enable randomizer function. If this bit is set to enable, ENABLE_ECC should be also enable.

Table continues on the next page...

NOTE

- Only 32-bit word size accesses are supported for burst mode accesses.
- Only 8-bit (1 byte), 16-bit (2 byte) and 32-bit (4 byte) word size supported for single access.
- Maximum number of burst length is 16.
- According to AXI protocol, burst access should not cross 4 KB blocks. In case EIM gets an access that crosses the 4 KB, memory address calculation is invalid.

AXI transfers shown in the table below are also supported. These AXI cycles will be translated into the necessary cycles on the memory side. For example, for optimal operation in case ARM cache is configured to 8 beat burst with wrap, a synchronous flash and cellular RAM memory should be configured in 16 word wrap burst mode when using a 16-bit data port, and in 8 word wrap burst mode when using a 32-bit data port. EIM uses BL bit field to support different memory configurations. The controller splits the transaction when needed in some cases. See Table 9-29.

Table 9-28. AXI Burst Cycles Supported

Burst Length - Number of data transfers	Burst size - Bytes in transfer	Burst type	Description
1	1	INCR	Single transfer
1	2	INCR	Single transfer
1	4	INCR	Single transfer
2	4	WRAP	2-beat wrapping burst
4	4	WRAP	4-beat wrapping burst
8	4	WRAP	8-beat wrapping burst
16	4	WRAP	16-beat wrapping burst
2	4	INCR	2-beat incrementing burst
3	4	INCR	3-beat incrementing burst
4	4	INCR	4-beat incrementing burst
5	4	INCR	5-beat incrementing burst
6	4	INCR	6-beat incrementing burst
7	4	INCR	7-beat incrementing burst
8	4	INCR	8-beat incrementing burst
9	4	INCR	9-beat incrementing burst
10	4	INCR	10-beat incrementing burst
11	4	INCR	11-beat incrementing burst
12	4	INCR	12-beat incrementing burst
13	4	INCR	13-beat incrementing burst
14	4	INCR	14-beat incrementing burst

Table continues on the next page...

1. Auto CMD12 response time-out. It is not certain whether the command is accepted by the card or not. The Driver should clear the Auto CMD12 error status bits and re-send the CMD12 until it is accepted by the card.
2. Auto CMD12 response CRC error. Since card responds to the CMD12, the card will abort the transfer. The Driver may ignore the error and clear the error status bit.
3. Auto CMD12 conflict error or not sent. The command is not sent, so the Driver shall send a CMD12 manually.

10.3.5.3.6 Card Interrupt

The external cards can inform the Host Controller by means of some special signals. For the SDIO card, it can be the low level on the DATA1 line during some special period. The uSDHC only monitors the DATA1 line and supports the SDIO interrupt.

When the SDIO interrupt is captured by the uSDHC, and the Host System is informed by the uSDHC asserting the uSDHC interrupt line, the interrupt service from the Host Driver is called.

As the interrupt source is controlled by the external card, the interrupt from the SDIO card must be serviced before the CINT bit is cleared by written. Refer to Card Interrupt Handling for the card interrupt handling flow.

10.3.5.4 Switch Function

A switch command shall be issued by the Host Driver to enable new features added to the SD/MMC spec. SD/MMC cards can transfer data at bus widths other than 1-bit. Different speed mode are also defined. To enable these features, a switch command shall be issued by the Host Driver.

For SDIO cards, the high speed mode/DDR50/SDR50/SDR104 are enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed. For SD cards, the high speed mode/DDR50/SDR50/SDR104 are queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For MMC cards , the high speed mode/HS400 are queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The SDR4-bit, SDR8-bit ,DDR4-bit and DDR8-bit width of the MMC is also enabled by the SWITCH command, but with a different argument.

11.2.2.2 Detailed Signal Descriptions

11.2.2.2.1 SIM_CLK

The SIM_CLK is an output from the chip to the SmartCard. This signal is the clock that the SIM module provides for the SmartCard. Typical frequencies are 1 MHz to 5 MHz. This clock is 372 times the data rate that is on pin SIM_TRXD. There is no required timing relationship between this clock signal and any of the other data signals. This is because of the asynchronous nature of the protocol.

11.2.2.2.2 SIM_RST_B

The SIM_RST_B is the reset signal from the SIM to the SmartCard.

11.2.2.2.3 SIM_SVEN

The SIM_SVEN is the SmartCard power supply enable control signal.

11.2.2.2.4 SIM_TRXD

The SIM_TRXD is the data transmitted/received from the SIM module to the SmartCard. In a 1-wire mode of operating, this output port must be bidirectional with a pull-up resister off chip.

11.2.2.2.5 SIM_PD

The SIM_PD is the SmartCard insertion detect.

11.2.3 SIM Functional Description

To best describe the organization of the SIM module from a user's point of view, it is instructive to view the SIM at a number of different levels of hierarchy. See Figure 11-31 for illustration of the organization of SIM and connection of the signals to the available serial port.

The SIM module is essentially a standard UART with some special provisions made for SIM card communication. The SIM consists of the following main parts:

- IP bus interface
- Bus interface

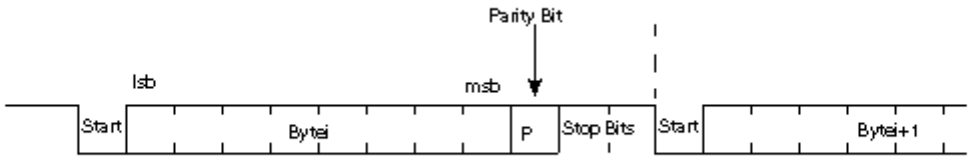


Figure 11-45. Parity Bit Diagram

When a parity error is detected on a given byte, the RCV_PF bit for that byte is set in the receive FIFO if NACK generation on errors is disabled (ANACK = 0 in the CNTL register). A parity error cannot cause an interrupt, but when NACK generation is enabled (ANACK = 1), it can signal the SIM transmitter to create a NACK pulse to the SIM card asking for a retransmission of the corrupted data.

11.2.3.5.5 Framing Error Detection

The receive state machine is responsible for detecting framing errors in the received data. A SIM data transaction is defined as 11 or 12 bits long consisting of the START bit, 8 data bits, 1 parity bit and 1 or 2 STOP bits. A framing error occurs when the STOP bit is not detected during the 11th bit time of a data transaction. The STOP bit is generally defined as two bit times (ETU's) of a high pulse following the parity bit. When the GUARD_CNTL register is programmed to 0xFF, the STOP bit is defined as one bit time. A framing error can only occur when the parity bit of the current byte is low, and the STOP bit arrives late. See the figure below for an illustration of a typical SIM data transaction with the STOP bits identified. Also, shown is a SIM data transaction with a late arriving STOP bit indicating a framing error.

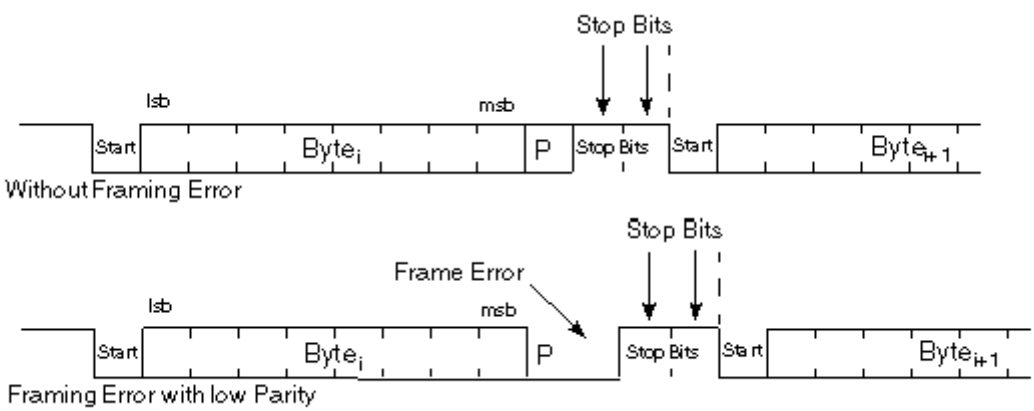


Figure 11-46. Framing Error Diagram

When a framing error is detected on a given byte, the RCV_FE bit for that byte is set in the receive FIFO. A framing error cannot cause an interrupt, nor can it create a NACK pulse to the SIM card asking for a retransmission of the corrupted data.

7. Enable the Character Wait Time Counter by setting the CWTEN bit in the CNTL register
8. Enable CRC or LRC error checking according to the ATR information by setting either the CRCEN or LRCEN bit in the CNTL register. These bits will never be set at the same time!

   For T=1 cards, the ATR is sent using a T=0 type of structure (12 ETU, no LRC or CRC). If a negotiation with the SIM card is desired, the software will send a PPS response to the SIM card. Otherwise, the protocol is initiated with a block transfer from the SIM module. In order to send the response or the first block, the following steps should be performed:

9. Set the desired transmit FIFO threshold level by writing the TDT[3:0] bits in the XMT_THRESHOLD register
10. Write the characters to be sent as response (max 16) to the transmit FIFO using the XMT_BUF register
11. Clear all transmit interrupt flags in the XMT_STAT register by writing a 1 to them
12. Enable the transmit interrupts desired by clearing the mask bits in the INT_MASK register. If more than 16 character are to be sent, it is suggested that the TDTF interrupt be used to signify when to write more characters to the transmit FIFO. This results in the most efficient transfer times to the SIM card.
13. Enable the transmission of the error checking characters (LRC or CRC) by setting the XMT_CRC_LRC bit in the CNTL register.

NOTE

If the card supports PPS, the software may not be allowed to send the LRC/CRC information until the PPS exchange is completed. If so, do not set the XMT_CRC_LRC bit during the PPS exchange.

14. Enable the transmitter by setting the XMT_EN bit in the ENABLE register

At this point, the SIM module will transmit the characters in the transmit FIFO. If more than 16 characters are to be sent, the transmit threshold interrupt will be set when the threshold number of characters are remaining in the FIFO. The software can then write an additional number of characters to be sent without interrupting transmission to the SIM card.

Once the transmission is complete, the SIM module should be completely configured for standard operation with the T=1 SIM card. The software can continue to service RDRF interrupts for received characters, and TDTF interrupts for transmitted characters.

11.2.5 SIM Memory Map/Register Definition

LCDIFx_THRES field descriptions

Field	Description
31–25 RSRVD2	This field is reserved. Reserved bits. Write as 0.
24–16 FASTCLOCK	This value should be set to a value of pixels, from 0 to 511. When the number of pixels in the input pixel FIFO is LESS than this value, the fast clock control output will be raised. This signal can be used to reduce the system bus clock frequency to save power during horizontal or vertical blanking intervals. This value should also be programmed to a value that is greater than the "PANIC" threshold value. This will allow a faster clock to recover the number of pixels in the FIFO before a "panic" level is encountered.
15–9 RSRVD1	This field is reserved. Reserved bits. Write as 0.
PANIC	This value should be set to a value of pixels from 0 to 511. When the number of pixels in the input pixel FIFO is less than this value, the internal panic control output will be raised. This signal can be used to raise the access eLCDIF's access priority.

EPDC_TCE_VSCAN*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved.
23–16 FRAME_END	Number of lines for frame end duration.
15–8 FRAME_BEGIN	Number of lines for frame begin duration.
FRAME_SYNC	Number of lines for frame sync duration.

13.6.4.28 EPDC Timing Control Engine OE timing control Register (EPDC_TCE_OE*n*)

This register contain delay programming values for the SDOEZ and SDOED source driver control signals

This register contain delay programming values for the SDOZ and SDOE source driver control signals

Address: 306F_0000h base + 2C0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SDOED_WIDTH								SDOED_DLY								SDOEZ_WIDTH								SDOEZ_DLY							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_TCE_OE*n* field descriptions

Field	Description
31–24 SDOED_WIDTH	Number of PIXCLK cycles from SDOED high to SDOED falling (Must be greater than 0)
23–16 SDOED_DLY	Number of PIXCLK cycles from SDOEZ low to SDOED rising (Must be greater than 0)
15–8 SDOEZ_WIDTH	Number of PIXCLK cycles from SDOEZ high to SDOEZ falling (Must be greater than 0)
SDOEZ_DLY	Number of PIXCLK cycles from SDLE falling edge to SDOEZ rising (Must be greater than 0)

13.6.4.29 EPDC Timing Control Engine Driver Polarity Register (EPDC_TCE_POLARITY*n*)

This registers allows for programming the polarity of source/gate driver control signals

This register houses FIFO control bits

- Bypass mode which directly outputs the data to the downstream connected Fetch Engine after shift operation. This is done through a valid/ready data interface.
- Support for 8x8 and 16x16 block mode and scanline mode.
- Support for 8, 16 and 32 bpp input data format.

13.7.9.2 Top-level architecture

The PXP consist of several pipelined blocks that perform the video source frame scaling, color space conversion, alpha-blending/color key algorithm, secondary CSC, pixel correction , input and/or output rotation, dithering and waveform processing. It is also capable of fetching data from and storing data to memory.

The legacy blocks operate within the requirements of the legacy PXP architecture, and perform operations on either 8x8 or 16x16 pixel blocks in the representative source buffers. The legacy pipeline operate within the context of two iteration counters that iterate through the appropriate grid of input blocks to produce the rotated output grid blocks in scan-line order. The dither blocks and the waveform processing engines (WFE) will operate in a scan line format based on the active size. The input fetch and store engines can work on either on a block by block basis or in the scan line format.

Figure 13-87 shows the high-level architecture of the scaling, color space conversion, blending, pixel correction , rotation engines, dithering, waveform processing, histogram along with four sets of fetch and store engines. The Alpha Surface Engine fetches one RGB graphics plane alpha surface (AS). The scaling engine fetches a single processed surface (PS), which can be blended with the AS surface. The scaling engine also supports an alpha channel for the PS image. Although the legacy PXP processes NxN pixel macro blocks, each of the AS or PS surfaces can have any pixel alignment within the output buffer. There are no restrictions and any pixel coordinates within the output buffer are valid. The upper left origin of the output buffer is defined as pixel 0,0. The upper left and lower right coordinates for each of the AS and PS are inclusive within the output buffer.

Figure 13-88 represents a sample output buffer configuration with both an AS and PS included. The alignment of each AS and PS within the output buffer can be at any arbitrary pixel locations. For example, the PS has an upper left coordinate (ULC) of 2,2 and a lower right coordinate (LRC) at pixel 13,13. The maximum value for the ULC and LRC for each of the AS and PS is bounded by the LRC of the output buffer, 15,15 for this example.

PXP_HW_PXP_DITHER_STORE_ADDR_0_CH1 field descriptions

Field	Description
OUT_BASE_ ADDR0	input base address0. For 2 channel, indicated the channel0 base address. For 1 channel and YUV422 2 plane, indicate the Y base address

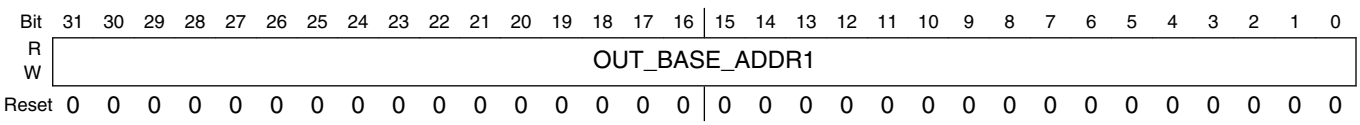
13.7.12.155 PXP_HW_PXP_DITHER_STORE_ADDR_1_CH1

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

EXAMPLE

Address: 3070_0000h base + AD0h offset = 3070_0AD0h



PXP_HW_PXP_DITHER_STORE_ADDR_1_CH1 field descriptions

Field	Description
OUT_BASE_ ADDR1	input base address1. For 2 channel, indicated the channel 1 base address. For 1 channel and YUV422 2 plane, indicate the UV base address

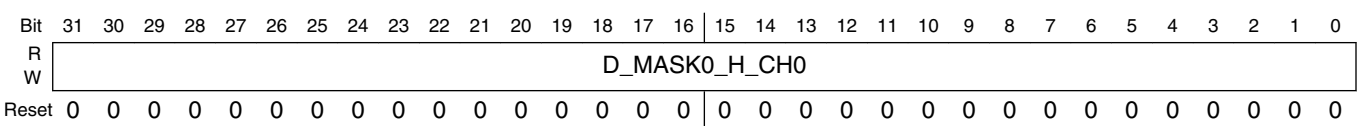
13.7.12.156 PXP_HW_PXP_DITHER_STORE_D_MASK0_H_CH0

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

EXAMPLE

Address: 3070_0000h base + AE0h offset = 3070_0AE0h



PXP_HW_PXP_DITHER_STORE_D_MASK0_H_CH0 field descriptions

Field	Description
D_MASK0_H_ CH0	data mask0 high byte

UARTx_UCR2 field descriptions (continued)

Field	Description
6 STPB	**Stop**. Controls the number of stop bits after a character. When STPB is low, 1 stop bit is sent. When STPB is high, 2 stop bits are sent. STPB also affects the receiver. 0 The transmitter sends 1 stop bit. The receiver expects 1 or more stop bits. 1 The transmitter sends 2 stop bits. The receiver expects 2 or more stop bits.
5 WS	**Word Size**. Controls the character length. When WS is high, the transmitter and receiver are in 8-bit mode. When WS is low, they are in 7-bit mode. The transmitter ignores bit 7 and the receiver sets bit 7 to 0. WS can be changed in-between transmission (reception) of characters, however not when a transmission (reception) is in progress, in which case the length of the current character being transmitted (received) is unpredictable. 0 7-bit transmit and receive character length (not including START, STOP or PARITY bits) 1 8-bit transmit and receive character length (not including START, STOP or PARITY bits)
4 RTSEN	**Request to Send Interrupt Enable.** Controls the RTS edge sensitive interrupt. When RTSEN is asserted and the programmed edge is detected on the RTS_B pin (the RTSF bit is asserted), an interrupt will be generated on the *interrupt_uart* pin. (See Table 15-31.) 0 Disable request to send interrupt 1 Enable request to send interrupt
3 ATEN	**Aging Timer Enable**. This bit is used to enable the aging timer interrupt (triggered with AGTIM) 0 AGTIM interrupt disabled 1 AGTIM interrupt enabled
2 TXEN	**Transmitter Enable**. Enables/Disables the transmitter. When TXEN is negated the transmitter is disabled and idle. When the UARTEN and TXEN bits are set the transmitter is enabled. If TXEN is negated in the middle of a transmission, the UART disables the transmitter immediately, and starts marking 1s. The transmitter FIFO cannot be written when this bit is cleared. 0 Disable the transmitter 1 Enable the transmitter
1 RXEN	**Receiver Enable**. Enables/Disables the receiver. When the receiver is enabled, if the RXD input is already low, the receiver does not recognize BREAK characters, because it requires a valid 1-to-0 transition before it can accept any character. 0 Disable the receiver 1 Enable the receiver
0 SRST	**Software Reset**. Once the software writes 0 to SRST_B, the software reset remains active for 4 *module_clock* cycles before the hardware deasserts SRST_B. The software can only write 0 to SRST_B. Writing 1 to SRST_B is ignored. 0 Reset the transmit and receive state machines, all FIFOs and register USR1, USR2, UBIR, UBMR, UBRC , URXD, UTXD and UTS[6-3]. 1 No reset