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	Not For New Designs
Core Processor	ARM® Cortex®-A9
Number of Cores/Bus Width	2 Core, 32-Bit
Speed	1.2GHz
Co-Processors/DSP	Multimedia; NEON™ SIMD
RAM Controllers	LPDDR2, LVDDR3, DDR3
Graphics Acceleration	Yes
Display & Interface Controllers	Keypad, LCD
Ethernet	10/100/1000Mbps (1)
SATA	SATA 3Gbps (1)
USB	USB 2.0 + PHY (4)
Voltage - I/O	1.8V, 2.5V, 2.8V, 3.3V
Operating Temperature	-20°C ~ 105°C (TJ)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC, Tamper Detection
Package / Case	624-LFBGA, FCBGA
Supplier Device Package	624-FCPBGA (21x21)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcimx6d5eym12ac

Chapter 50
Power Management Unit (PMU)

23.5.36 Tx Multicast Packets Statistic Register (ENET_RMON_T_MC_PKT)

Address: 218_8000h base + 20Ch offset = 218_820Ch

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	0																TXPKTS															
W																																
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

ENET_RMON_T_MC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Multicast packets

23.5.37 Tx Packets with CRC/Align Error Statistic Register (ENET_RMON_T_CRC_ALIGN)

Address: 218_8000h base + 210h offset = 218_8210h

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	0																TXPKTS															
W																																
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

ENET_RMON_T_CRC_ALIGN field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Packets with CRC/align error

FLEXCANx_CTRL2 field descriptions (continued)

Field	Description
	Reserved

31.4.1.5.2 Mask BLT

For Mask BLT, the bit from the stream is used to toggle on/off a color in the source frame buffer. Mask BLT takes its color source from memory and its monochrome mask from the command stream. Clipping is supported and is performed on a per pixel basis.

31.4.1.6 Filter BLT

Filter blit performs high quality scaling, up or down, using an FIR re-sampling filter with up to 9 taps. The sub-pixel coordinates (locations between the pixel grids) are generated by the drawing engine. The filter block in the drawing engine uses the sub-pixel information to select the appropriate filter kernel. R2D GPU processes 1 pixel every cycle when performing filter blit.

A stretch- or shrink-factor of 15.16 fixed-point format is supported. To generate a single destination pixel requires 9 source pixels. An image is scaled in two passes, one for X-dimension (HOR_FILTER_BLT) and the other for Y-dimension (VER_FILTER_BLT). The software sets up the filter kernel/coefficient table and the kernel size, as well as a temporary buffer for storing intermediate results. After the first pass is completed, intermediate results are sent back to memory, and then the second pass starts to scale the first-pass image. Because of this two-step procedure, the throughput of FILTER BLT is lower than that of STRETCH BLT. Also the Filter Kernel Table may need to be reloaded, and some cycles are consumed in calculating the stepping parameters.

When the stretch or shrink factor is 1, the filterBlit works as a bitBlit copy. It can be used as format converter in that case, for instance, YUV to RGB converter. To use as a format converter, only one pass (HOR_FILTER_BLT or VER_FILTER_BLT) is needed. To optimize the memory bandwidth, when using filterBlit to do YUV to RGB filtering, the temporary target buffer format can be specified as YUY2 to process Y-dimension filtering (VER_FILTER_BLT). This is to avoid converting YUV to A8R8G8B8 in the 1st vertical pass to reduce the memory bandwidth and increase the pixel processing rate. This is the only special case that GPU may use YUY2 as target format.

The Filter BLT primitive supports the following source and destination image formats:

Table 31-2. Filter BLT Formats

Formats	Source Image	Destination Image
A1R5G5B5	Yes	Yes
A4R4G4B4	Yes	Yes
A8R8G8B8	Yes	Yes
R5G6B5	Yes	Yes
X1R5G5B5	Yes	Yes

Table continues on the next page...

- HDCP interface
- External ROM interface for key storage
- External RAM interface for revocation
- Random number generator interface
- Video input interface
- RGB 4:4:4
- YCbCr4:2:2
- YCbCr4:4:4
- Digital audio input interface
- AHB audio DMA
- System interface
- AMBA AHB
- Scan test interface
- HDMI TX PHY interface
- CEC interface

33.1.1.2 Features

HDMI TX includes the following features:

- Supported video formats:
- All CEA-861-E video formats up to 1080p at 60Hz and 720p/1080i at 120Hz
- Supported colorimetry:
- 24bit RGB 4:4:4
- 24bit YCbCr 4:4:4
- 16bit YCbCr 4:2:2
- xvYCC601
- xvYCC709
- Integrated color space converter:
- RGB(4:4:4) to/from YCbCr(4:4:4 or 4:2:2)
- Optional HDMI 1.4a supported video formats:
- HDMI 1.4a 3D video modes with up to 266MHz (TMDS clock)
- Optional HDMI 1.4a supported colorimetry:
- sYCC601
- Adobe RGB
- Adobe YCC601
- Optional HDMI 1.4a supported Infoframes:
- Audio InfoFrame packet extension to support LFE playback level information
- AVI infoFrame packet extension to support YCC Quantization range (Limited Range, Full
- Range)

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_025C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC)	32	R/W	0000_0005h	36.4.147/ 2079
20E_0260	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN)	32	R/W	0000_0005h	36.4.148/ 2080
20E_0264	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC)	32	R/W	0000_0005h	36.4.149/ 2081
20E_0268	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04)	32	R/W	0000_0005h	36.4.150/ 2082
20E_026C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05)	32	R/W	0000_0005h	36.4.151/ 2083
20E_0270	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06)	32	R/W	0000_0005h	36.4.152/ 2084
20E_0274	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07)	32	R/W	0000_0005h	36.4.153/ 2085
20E_0278	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08)	32	R/W	0000_0005h	36.4.154/ 2086
20E_027C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09)	32	R/W	0000_0005h	36.4.155/ 2087
20E_0280	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10)	32	R/W	0000_0005h	36.4.156/ 2088
20E_0284	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11)	32	R/W	0000_0005h	36.4.157/ 2089
20E_0288	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12)	32	R/W	0000_0005h	36.4.158/ 2090
20E_028C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13)	32	R/W	0000_0005h	36.4.159/ 2091
20E_0290	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14)	32	R/W	0000_0005h	36.4.160/ 2092
20E_0294	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15)	32	R/W	0000_0005h	36.4.161/ 2093
20E_0298	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16)	32	R/W	0000_0005h	36.4.162/ 2094
20E_029C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17)	32	R/W	0000_0005h	36.4.163/ 2095
20E_02A0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18)	32	R/W	0000_0005h	36.4.164/ 2096
20E_02A4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19)	32	R/W	0000_0005h	36.4.165/ 2097
20E_02A8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA7)	32	R/W	0000_0005h	36.4.166/ 2098
20E_02AC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA6)	32	R/W	0000_0005h	36.4.167/ 2099
20E_02B0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5)	32	R/W	0000_0005h	36.4.168/ 2100

Table continues on the next page...

36.4.102 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14)

Address: 20E_0000h base + 1A8h offset = 20E_01A8h

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

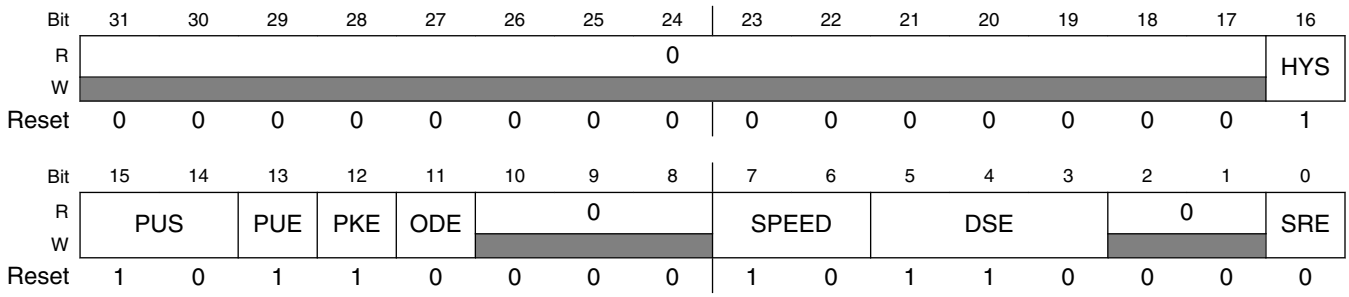
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT14. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT14. NOTE: Pad DISP0_DAT14 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA14. 001 **ALT1** — Select signal IPU2_DISP0_DATA14. 011 **ALT3** — Select signal AUD5_RXC. - Configure register IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO08.

36.4.438 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01)

Address: 20E_0000h base + 6E8h offset = 20E_06E8h



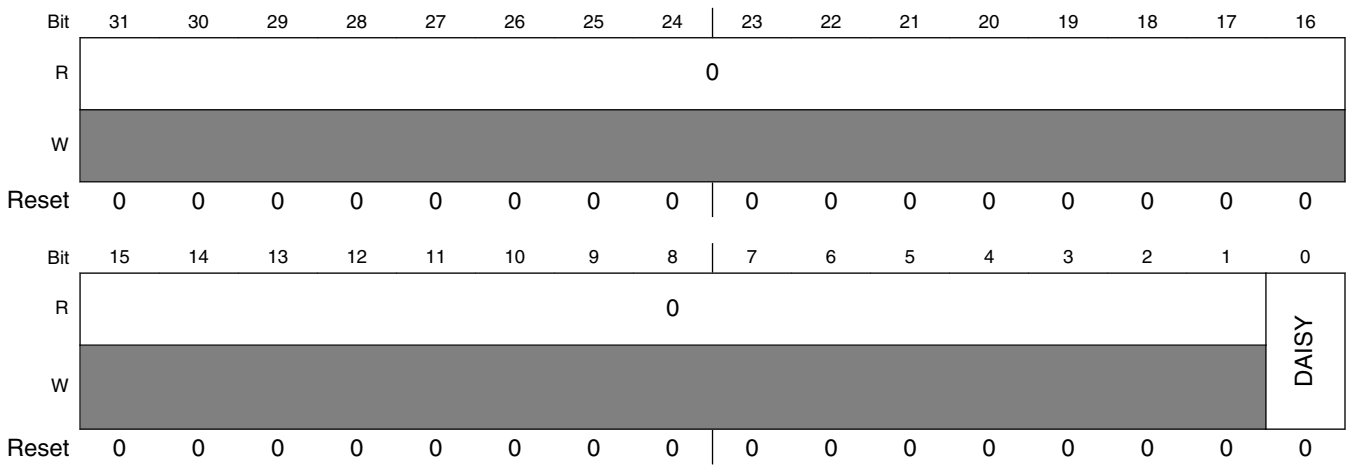
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

36.4.543 Select Input Register (IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT)

Address: 20E_0000h base + 890h offset = 20E_0890h

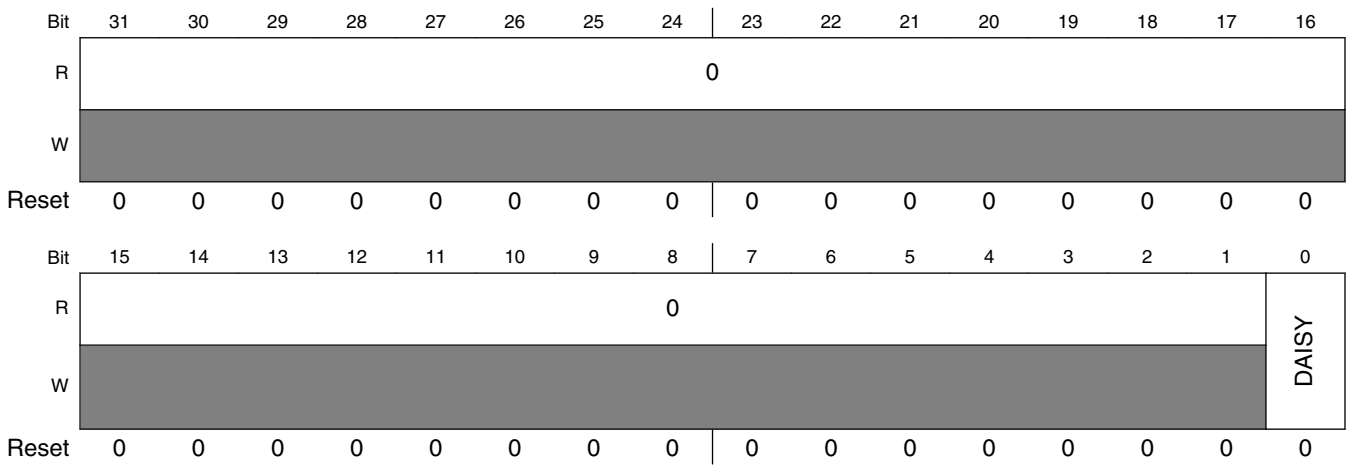


IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_EB2_B_ALT4** — Selecting ALT4 mode of pad EIM_EB2 for HDMI_TX_DDC_SCL. 1 **KEY_COL3_ALT2** — Selecting ALT2 mode of pad KEY_COL3 for HDMI_TX_DDC_SCL.

36.4.551 Select Input Register (IOMUXC_IPU2_SENS1_DATA10_SELECT_INPUT)

Address: 20E_0000h base + 8B0h offset = 20E_08B0h



IOMUXC_IPU2_SENS1_DATA10_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA22_ALT3** — Selecting ALT3 mode of pad EIM_D22 for IPU2_CSI1_DATA10. 1 **EIM_EB1_B_ALT2** — Selecting ALT2 mode of pad EIM_EB1 for IPU2_CSI1_DATA10.

IPUx_DMFC_WR_CHAN_DEF field descriptions (continued)

Field	Description
	This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
17 dmfc_wm_en_1c	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16 Reserved	This read-only field is reserved and always has the value 0.
15–13 dmfc_wm_clr_2	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
12–10 dmfc_wm_set_2	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_2	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
8 Reserved	This read-only field is reserved and always has the value 0.
7–5 dmfc_wm_clr_1	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
4–2 dmfc_wm_set_1	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
1 dmfc_wm_en_1	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
0 Reserved	This read-only field is reserved and always has the value 0.

5. MMDC drives one DQS pulse to the DDR external device
6. MMDC waits 16 cycles (to guarantee that the DQ prime data is stable) and samples the associated prime DQ bit (for example for DQS1 the MMDC samples DQ[8])
7. MMDC increments the write leveling delay line by 1/8 cycle and perform measurement process in order to load the updated value to the associated delay-line
8. MMDC repeates steps 5-7 till the write leveling delay is 1 cycle
9. MMDC checks the 8 bit prime DQ results for each DQS and finds the first transition from 0 to 1. If no transition is found then the MMDC indicates an error at MPWLGCR[HW_WL_ERR#]
10. MMDC stores the value that issues the last "0" on the prime DQ before the transition and loads it to the write leveling delay-line. The MMDC initiates a fine-tune process by incrementing the delay-line values by 1 step (which is 1/256 part of a cycle) till detecting the most accurate transition from 0 to 1
11. Upon completion of this process the MMDC de-asserts the MPWLGCR[HW_WL_EN] and update the most accurate value of the delay-line at the associated MPWLDECTRL#[WL_DL_ABS_OFFSET#]
12. MMDC perform measurement process in order to load the most accurate value to the associated delay-line
13. User should issue MRS command to exit write leveling mode
14. The user should read the results of the associated delay-line at MPWLDECTRL#[WL_DL_ABS_OFFSET#] and in case the user estimates that the reasonable delay may be above 1 cycle then the user should indicate it at MPWLDECTRL#[WL_CYC_DEL#]. Moreover the user should indicate it in MDMISC[WALAT] field. For example, if the result of the write leveling calibration is 100/256 parts of a cycle, but the user estimates that the delay is above 2 cycles then MPWLDECTRL#[WL_CYC_DEL#] should be configured to 2, so the total delay will be 2 and 100/256 parts of a cycle
15. Return the DQS output enable to functional mode by deasserting MDSCR[WL_EN]

44.11.6.2 SW Write Leveling Calibration

The following steps should be executed:

NOTE

It is recommended to perform the write calibration using the HW method. The SW method is provided for debug purposes only.

1. Configure the external DDR device to enter write leveling mode through MRS command
2. Activate the DQS output enable by setting MDSCR[WL_EN]

MMDCx_MPRDDQBY1DL field descriptions (continued)

Field	Description
	101 Add dq14 delay of 5 delay units. 110 Add dq14 delay of 6 delay units. 111 Add dq14 delay of 7 delay units.
23 Reserved	This read-only field is reserved and always has the value 0.
22–20 rd_dq13_del	Read dqs1 to dq13 delay fine-tuning. This field holds the number of delay units that are added to dq13 relative to dqs1. 000 No change in dq13 delay 001 Add dq13 delay of 1 delay unit 010 Add dq13 delay of 2 delay units. 011 Add dq13 delay of 3 delay units. 100 Add dq13 delay of 4 delay units. 101 Add dq13 delay of 5 delay units. 110 Add dq13 delay of 6 delay units. 111 Add dq13 delay of 7 delay units.
19 Reserved	This read-only field is reserved and always has the value 0.
18–16 rd_dq12_del	Read dqs1 to dq12 delay fine-tuning. This field holds the number of delay units that are added to dq12 relative to dqs1. 000 No change in dq12 delay 001 Add dq12 delay of 1 delay unit 010 Add dq12 delay of 2 delay units. 011 Add dq12 delay of 3 delay units. 100 Add dq12 delay of 4 delay units. 101 Add dq12 delay of 5 delay units. 110 Add dq12 delay of 6 delay units. 111 Add dq12 delay of 7 delay units.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 rd_dq11_del	Read dqs1 to dq11 delay fine-tuning. This field holds the number of delay units that are added to dq11 relative to dqs1. 000 No change in dq11 delay 001 Add dq11 delay of 1 delay unit 010 Add dq11 delay of 2 delay units. 011 Add dq11 delay of 3 delay units. 100 Add dq11 delay of 4 delay units. 101 Add dq11 delay of 5 delay units. 110 Add dq11 delay of 6 delay units. 111 Add dq11 delay of 7 delay units.
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 rd_dq10_del	Read dqs1 to dq10 delay fine-tuning. This field holds the number of delay units that are added to dq10 relative to dqs1. 000 No change in dq10 delay

Table continues on the next page...

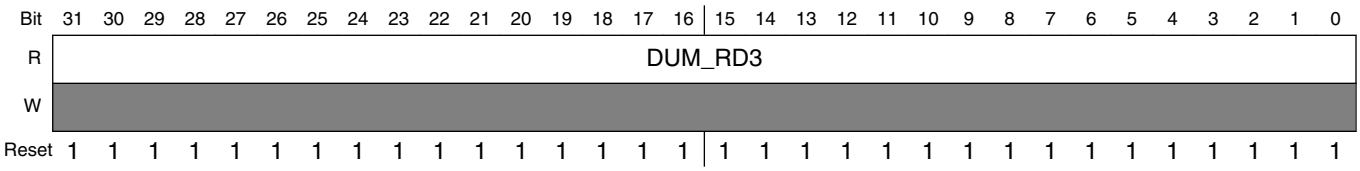
44.12.72 MMDC PHY SW Dummy Read Data Register 3 (MMDCx_MPSWDRDR3)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8A4h offset



MMDCx_MPSWDRDR3 field descriptions

Field	Description
DUM_RD3	Dummy read data3. This field holds the forth data that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted.

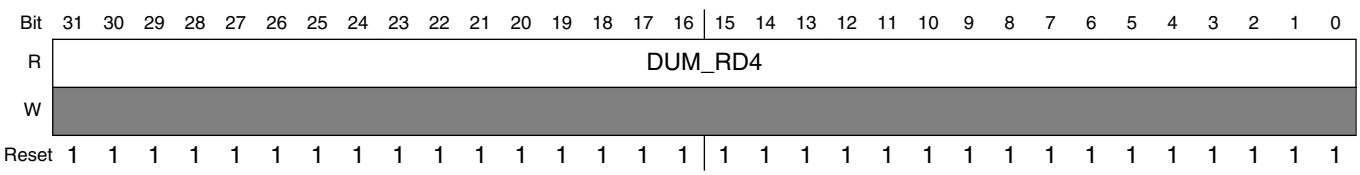
44.12.73 MMDC PHY SW Dummy Read Data Register 4 (MMDCx_MPSWDRDR4)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8A8h offset



MMDCx_MPSWDRDR4 field descriptions

Field	Description
DUM_RD4	Dummy read data4. This field holds the fifth data (only in case of burst length 8 (BL =1)) that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted.

Chapter 49
PCI Express PHY (PCIe_PHY)

49.1 Overview

PCIe 2.0 PHY is a complete mixed-signal semiconductor intellectual property (IP) solution, designed for single-chip integration into computer applications.

The PCIe 2.0 PHY supports both the 5 Gbp/s data rate of the PCI Express Gen 2.0 specifications as well as being backwards compatible to the 2.5Gb/s Gen 1.1 specification.

This chapter provides an introduction to the PCIe 2.0 PHY and its features.

49.2 Applications

Designed for low power, the PCIe 2.0 PHY allows designers to introduce competitive products using the latest generation of the PCI Express standard.

49.3 PCIe2 PHY Features

49.3.1 Standards Compliance

The PCIe2 PHY is fully compliant with all of the required features of the following standards:

- PCI Express Base Specification, Revision 2.0 (including legacy 2.5-Gbps support
- 5.0 Gbps data rate
- PCI Express Base Specification, Revision 1.1
- 2.5Gbps data rate

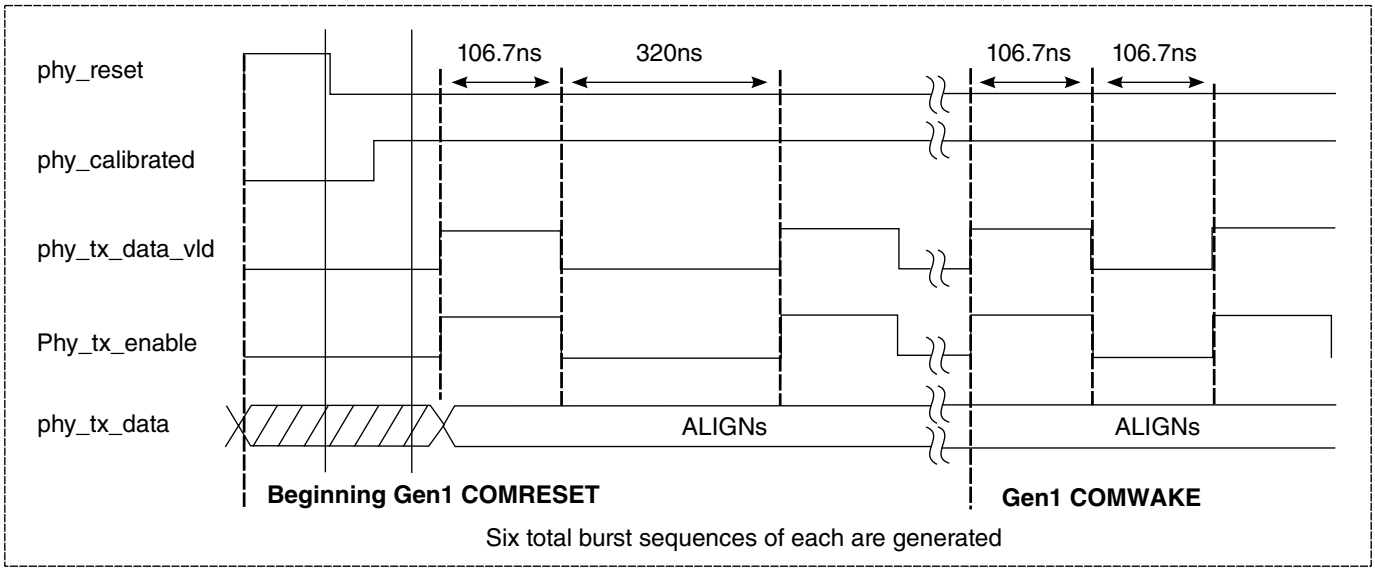


Figure 53-12. Link Layer Generated Tx OOB Signaling

53.3.4.5.3.3 Link Layer Rx OOB Sequence Detection

When the Link Layer detects the Rx OOB sequences, the Rx interface connections are different than when the PHY detects them.

Figure 53-12 depicts a small portion of an initialization phase, which includes the first part of a COMINIT condition, followed by a portion of a COMWAKE.

The actual OOB sequences are comprised of specifically timed ALIGN Primitives in combination with negation of the phy_sig_det signal (indicating NULL on the differential pair). However, only the phy_sig_det signal is used for qualifying Rx OOB sequences by the Link Layer.

NOTE

The PHY must “condition” (filter and delay) phy_sig_det so that the OOB detector accurately detects SATA sequences and prevents the Link Layer from accidentally misreading the Device as disconnected, thus avoiding reset. A filter is required to prevent phy_sig_det from detecting transient ‘loss of signal’ due to bits on the Rxp/Rxn wires crossing the zero voltage level.

To ensure that power modes function properly, phy_sig_det should be timed to the data delay through the PHY so that it does not go low until at least four PMACKs are passed

55.4.2.3.6 Next Channel Decision Tree

The next channel number is computed from the runnable channels list, the current channel number, and their respective priorities.

It is re-evaluated every cycle, but is only used when the current channel yields or terminates by executing a yield, yieldge, or done instruction.

The decision tree is based on the selection of the runnable channel that has the highest priority.

The highest priority channel is selected according to the following rules:

- Runnable channels are sorted by priority.
- If one of the channels with the highest priority had been preempted by a channel with a higher priority, but did not want to yield to a channel of the same priority (for example, it executed a yield, not a yieldge), it is elected as the next channel.
- The channels that belong to the highest priority group are sorted by their number and the channel that has the highest number in this group becomes the next channel. For example, if priorities are the same, channel 31 will be selected before channel 30.

When the current channel requires a reschedule with a yield(ge) or a done instruction, the context switch decision is based on the instruction parameter, the current channel number and priority, and the next channel number and priority. The possible cases are all listed in the following table. The grayed cells correspond to unusual cases that should not occur with a typical usage of the SDMA.

Table 55-6. Channel Switching Decision with a yield, yield(ge), or done

Instruction	Current Channel	Next Channel	Priorities Comparison	New Running Channel/Comments
yield (done 0)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Current
			Current < Next	Next[1]
	Not runnable	Not runnable	none	none[2] (occurs when the channel was disabled by the Arm platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the Arm platform)
yieldge (done 1)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Next[1]
			Current < Next	Next[1]

Table continues on the next page...

There are also commands that interact with the core: dmov, run_core, exec_core, exec_once, and debug_rqst. These commands are core status dependent, as follows:

- During user mode only the debug_rqst is taken into account.
- During debug mode, all these commands are taken into account except the debug_rqst. For example, an exec_once command requested while not in debug mode has no effect.

55.4.13.3.2 Execution Request

The SDMA starts executing a task in debug mode when requested by the OnCE controller. The execution starting time depends on the type of access used to communicate with the OnCE.

If the JTAG is used, the request is send after decoding the update_dr state in the TAP controller. Therefore, always cross this state when sending a command through the JTAG. If the OnCE is driven from the Arm platform side, the request is sent after detecting a write access to the ONCE_CMD register. All the registers involved in this operation must be loaded first.

The following is an example of an exec_core command execution from the Arm platform side: After writing '010' in the ONCE_CMD register, the OnCE controller asks the SDMA to execute the instruction contained in the ONCE_INSTR register. The instruction involved should be available in the ONCE_INSTR register before the beginning of the execution.

55.4.13.3.3 Command Execution

The following list shows the commands and details how each command is executed:

- rstatus command execution-The rstatus command exports the content of the OnCE status register (OSR). If the JTAG is used, the status information is captured in the OnCE status register during the capture_dr state, and shifted out after 16 TCK clock cycles in the shift_dr state. The rstatus command is not supported on the Arm platform side, but a status register is provided instead. The rstatus may be performed in both debug and user modes.
- dmov command execution-The dmov command accesses SDMA internal registers. Executing a dmov instruction exchanges the 32-bit data values between the SDMA data register and the general register GReg[1].
- If the JTAG is used, the content of GReg1 is captured in the SDMA data register during the capture_dr state, then it is shifted out after 32 TCK clock cycles in the shift_dr state. During the update_dr state, GReg1 is updated with the new, shifted-in 32-bit data value. If the OnCE is driven from the Arm platform side, the data values

```
                    re-label the card as SDIO;
                    ignore the error or report it;
                    return; // card is identified as SDIO card
          } // of if (card is ...
          send_command(SEND_OP_COND, <voltage range>, <...>);
          if (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted,
either
                    label the card as UNKNOWN;
                    return;
          } // of if (RESP_TIMEOUT ...
} // of else
}
```

67.5.2.4 Card Registry

Card registry for the MMC and SD/SDIO/SD Combo cards are different. For the SD Card, the Identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the Card spec). At this time, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host will request the card to send their valid operation conditions.

The response to ACMD41 is the operation condition register of the card. The same command shall be send to all of the new cards in the system. Incompatible cards are put into the Inactive State. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready State), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification State.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA will be used to address the card for future data transfer operations. Once the RCA is received, the card changes its state to the Standby State. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in system.

For MMC operation, the host starts the card identification process in open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is activated the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the "wired OR" operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive State. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards