3. If MSS.SWSYSCMD = 1, read the MSD register to receive the system data sent from MediaLB Controller.

43.5.3 Low Power Mode

MLB doesn't provide dedicated low power mode features.

In case the clocks of digital IP need to shut down to save power, the following operations are recommended before entering low power mode:

- Finish any active MLB transfer
- Disable MLB (clear the MLBEN and MLBPEN bits in MLBC0)
- Disable HBI (clear all bits in HCMR0 and HCMR1, clear EN bit in HCTL)
- Mask AHB interrupts (clear all bits in ACMR0 and ACMR1)

For information on configuring the MLB IP if the clocks are re-enabled, see Configure the Hardware.

43.6 MLB Memory Map/Register Definition

The MLB registers are divided into 4 sections.

The first section begins at offset 0h00 and describes the MediaLB block registers.

The second section begins at offset 0h80 and it shows the address mapping of the Internal HBI Registers.

The third set begins at offset 0hC0 and implements ten 32-bit I/O registers for CTR transfers, including: data registers (MDATn), write enable registers (MDWEn), a control register (MCTL), and an address register (MADR).

The fourth set, the AMBA AHB registers, begins at offset 0h3C0. They consist of:

- one 32-bit register for control
- two 32-bit registers for interrupt status
- two 32-bit registers for channel interrupt masks

MLB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_C000	MediaLB Control 0 Register (MLB_MLBC0)	32	R/W	0000_0000h	43.6.1/3808
218_C00C	MediaLB Channel Status 0 Register (MLB_MS0)	32	R	0000_0000h	43.6.2/3810
218_C00D	MediaLB 6-pin Control 2 Register (MLB_MLBPC2)	32	R	0000_0000h	43.6.3/3811
218_C014	MediaLB Channel Status1 Register (MLB_MS1)	32	R	0000_0000h	43.6.4/3811
218_C020	MediaLB System Status Register (MLB_MSS)	32	R	0000_0000h	43.6.5/3812
218_C024	MediaLB System Data Register (MLB_MSD)	32	R	0000_0000h	43.6.6/3813
218_C02C	MediaLB Interrupt Enable Register (MLB_MIEN)	32	R/W	0000_0000h	43.6.7/3814
218_C03C	MediaLB Control 1 Register (MLB_MLBC1)	32	R	0000_0000h	43.6.8/3815
218_C080	HBI Control Register (MLB_HCTL)	32	R/W	0000_0000h	43.6.9/3816
218_C088	HBI Channel Mask 0 Register (MLB_HCMR0)	32	R/W	0000_0000h	43.6.10/ 3817
218_C08C	HBI Channel Mask 1 Register (MLB_HCMR1)	32	R/W	0000_0000h	43.6.11/ 3817
218_C090	HBI Channel Error 0 Register (MLB_HCER0)	32	R	0000_0000h	43.6.12/ 3818
218_C094	HBI Channel Error 1 Register (MLB_HCER1)	32	R	0000_0000h	43.6.13/ 3818
218_C098	HBI Channel Busy 0 Register (MLB_HCBR0)	32	R	0000_0000h	43.6.14/ 3819
218_C09C	HBI Channel Busy 1 Register (MLB_HCBR1)	32	R	0000_0000h	43.6.15/ 3819
218_C0C0	MIF Data 0 Register (MLB_MDAT0)	32	R/W	0000_0000h	43.6.16/ 3820
218_C0C4	MIF Data 1 Register (MLB_MDAT1)	32	R/W	0000_0000h	43.6.17/ 3820
218_C0C8	MIF Data 2 Register (MLB_MDAT2)	32	R/W	0000_0000h	43.6.18/ 3820
218_C0CC	MIF Data 3 Register (MLB_MDAT3)	32	R/W	0000_0000h	43.6.19/ 3821
218_C0D0	MIF Data Write Enable 0 Register (MLB_MDWE0)	32	R/W	0000_0000h	43.6.20/ 3821
218_C0D4	MIF Data Write Enable 1 Register (MLB_MDWE1)	32	R/W	0000_0000h	43.6.21/ 3821
218_C0D8	MIF Data Write Enable 2 Register (MLB_MDWE2)	32	R/W	0000_0000h	43.6.22/ 3822
218_C0DC	MIF Data Write Enable 3 Register (MLB_MDWE3)	32	R/W	0000_0000h	43.6.23/ 3822
218_C0E0	MIF Control Register (MLB_MCTL)	32	R	0000_0000h	43.6.24/ 3823
218_C0E4	MIF Address Register (MLB_MADR)	32	R/W	0000_0000h	43.6.25/ 3823
218_C3C0	AHB Control Register (MLB_ACTL)	32	R/W	0000_0000h	43.6.26/ 3824

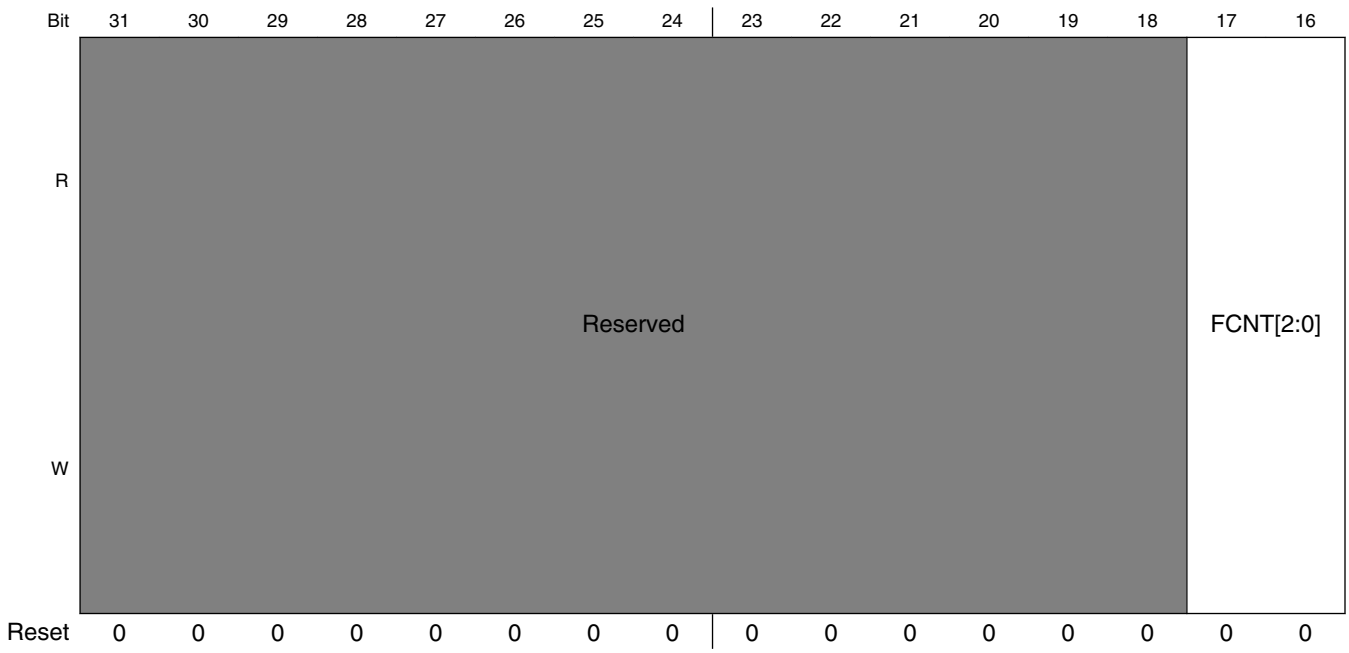
Table continues on the next page...

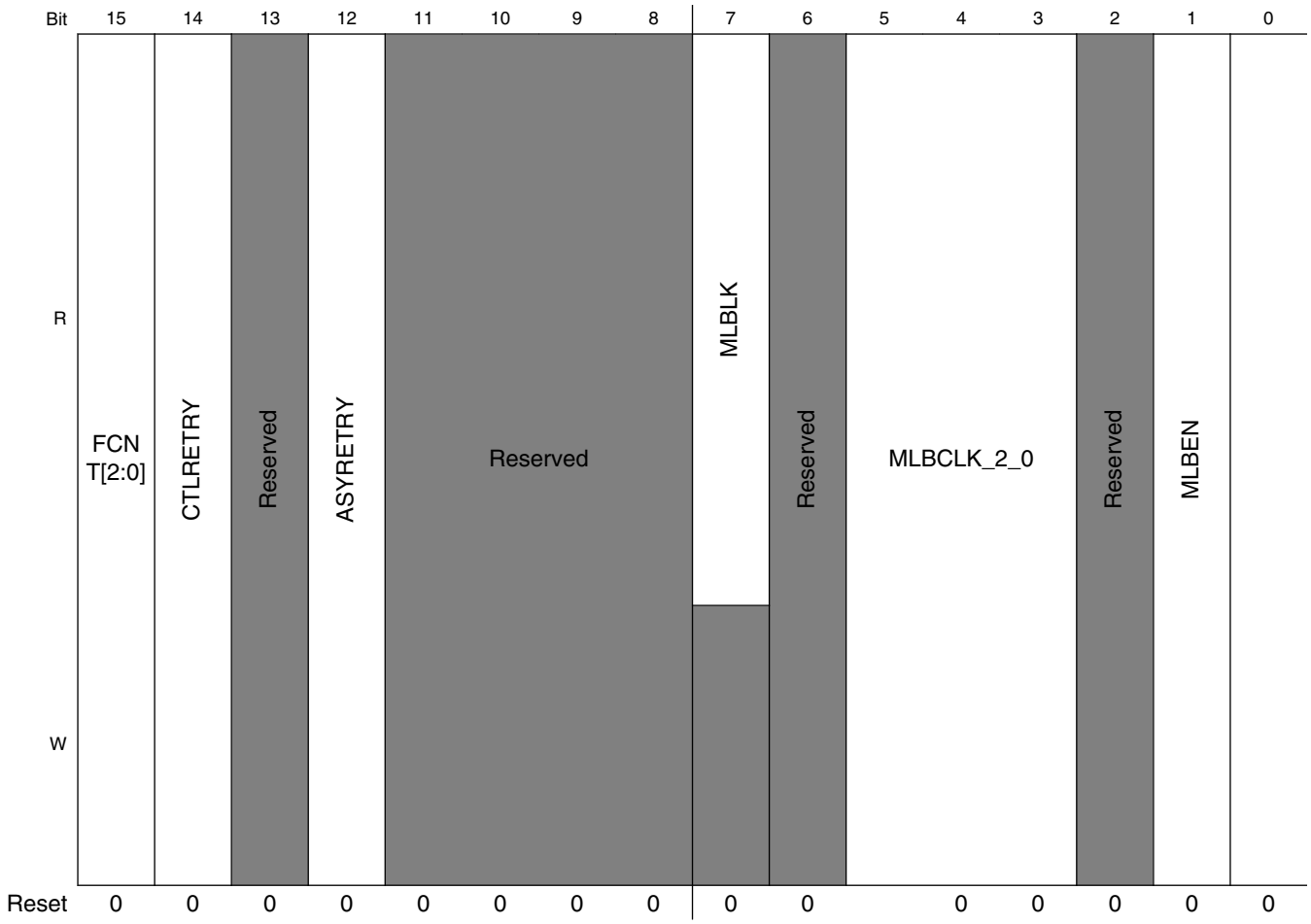
MLB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_C3D0	AHB Channel Status 0 Register (MLB_ACSR0)	32	R	0000_0000h	43.6.27/ 3825
218_C3D4	AHB Channel Status 1 Register (MLB_ACSR1)	32	R	0000_0000h	43.6.28/ 3826
218_C3D8	AHB Channel Mask 0 Register (MLB_ACMR0)	32	R/W	0000_0000h	43.6.29/ 3826
218_C3DC	AHB Channel Mask 1 Register (MLB_ACMR1)	32	R/W	0000_0000h	43.6.30/ 3827

43.6.1 MediaLB Control 0 Register (MLB_MLBC0)

Address: 218_C000h base + 0h offset = 218_C000h





MLB_MLBC0 field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17–15 FCNT[2:0]	The number of frames per sub-buffer for synchronous channels. 000 1 frame per sub-buffer (Operation is the same as Standard mode.) 001 2 frames per sub-buffer 010 4 frames per sub-buffer 011 8 frames per sub-buffer 100 16 frames per sub-buffer 101 32 frames per sub-buffer 110 64 frames per sub-buffer 111 Reserved
14 CTLRETRY	Control Tx packet retry. When set, a control packet that is flagged with a Break or ProtocolError by the receiver is retransmitted. When cleared, a control packet that is flagged with a Break or ProtocolError by the receiver is skipped.
13 -	This field is reserved. Reserved

Table continues on the next page...

MLB_MLBC0 field descriptions (continued)

Field	Description
12 ASYRETRY	Asynchronous Tx packet retry. When set, an asynchronous packet that is flagged with a Break or ProtocolError by the receiver is retransmitted. When cleared, an asynchronous packet that is flagged with a Break or ProtocolError by the receiver is skipped.
11–8 -	This field is reserved. Reserved
7 MLBLK	MediaLB lock status. When set, indicates that the MediaLB block is synchronized to the incoming MediaLB frame. If **MLBLK** is clear (unlocked), **MLBLK** is set after FRAMESYNC is detected at the same position for three consecutive frames. If **MLBLK** is set (locked), **MLBLK** is cleared after not receiving FRAMESYNC at the expected time for two consecutive frames. While **MLBLK** is set, FRAMESYNC patterns occurring at locations other than the expected one are ignored. (read-only)
6 -	This field is reserved. Reserved
4–2 MLBCLK_2_0	MLB_CLK (MediaLB clock) speed select. 000 256xFs (for **MLBPEN** = 0) 001 512xFs (for **MLBPEN** = 0) 010 1024xFs (for **MLBPEN** = 0)
1 -	This field is reserved. Reserved
0 MLBEN	MediaLB enable. When set, MLB_CLK (MediaLB clock), MLB_SIG (signal), and MLB_DATA (data) are received and transmitted on the appropriate MediaLB pins.

43.6.2 MediaLB Channel Status 0 Register (MLB_MS0)

Address: 218_C000h base + Ch offset = 218_C00Ch

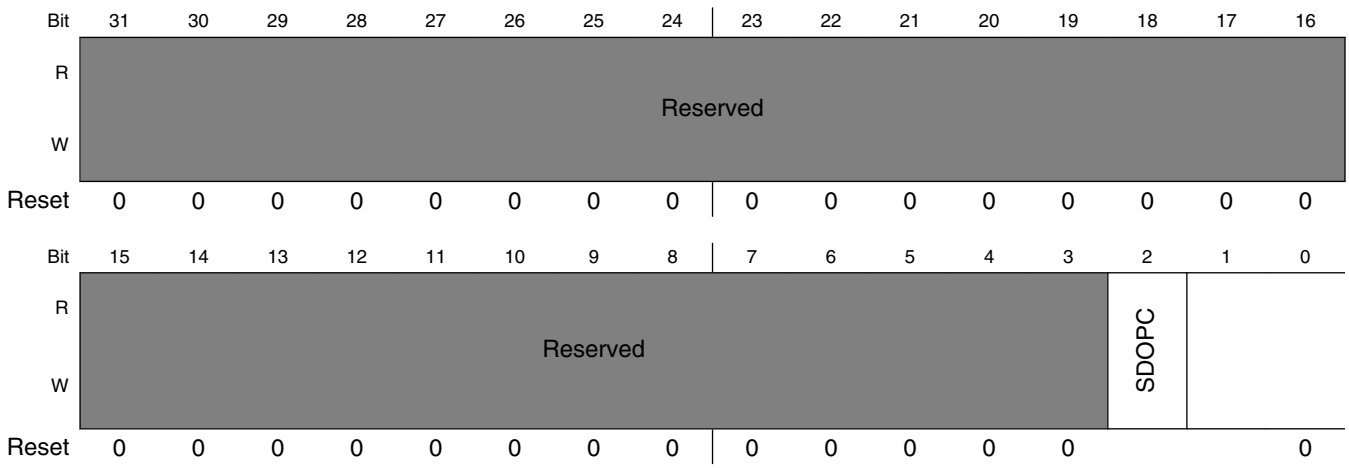
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MCS_31_0																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MS0 field descriptions

Field	Description
MCS_31_0	MediaLB channel status. Indicates the channel status for MediaLB channels 31 to 0. Channel status bits are set by hardware and cleared by software. Status is only set if the appropriate bits in the MIEN register are set.

43.6.3 MediaLB 6-pin Control 2 Register (MLB_MLBPC2)

Address: 218_C000h base + Dh offset = 218_C00Dh

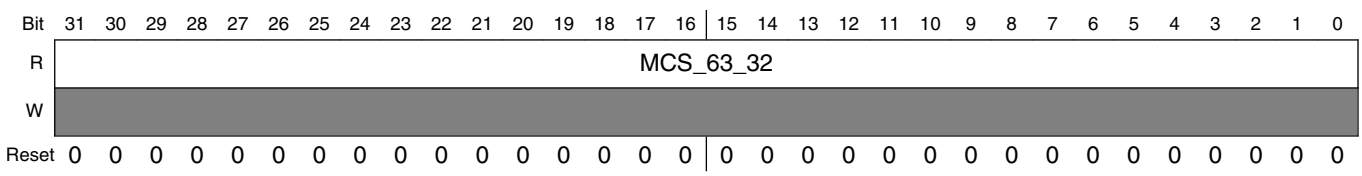


MLB_MLBPC2 field descriptions

Field	Description
31–3 -	This field is reserved. Reserved.
0 SDOPC	MLB 3-pin interface: Signal/Data output phase control. 0 MLB_SIG / MLB_DATA launch at rising edge of MLB_CLK(default) 1 MLB_SIG / MLB_DATA launch at falling edge of MLB_CLK

43.6.4 MediaLB Channel Status1 Register (MLB_MS1)

Address: 218_C000h base + 14h offset = 218_C014h

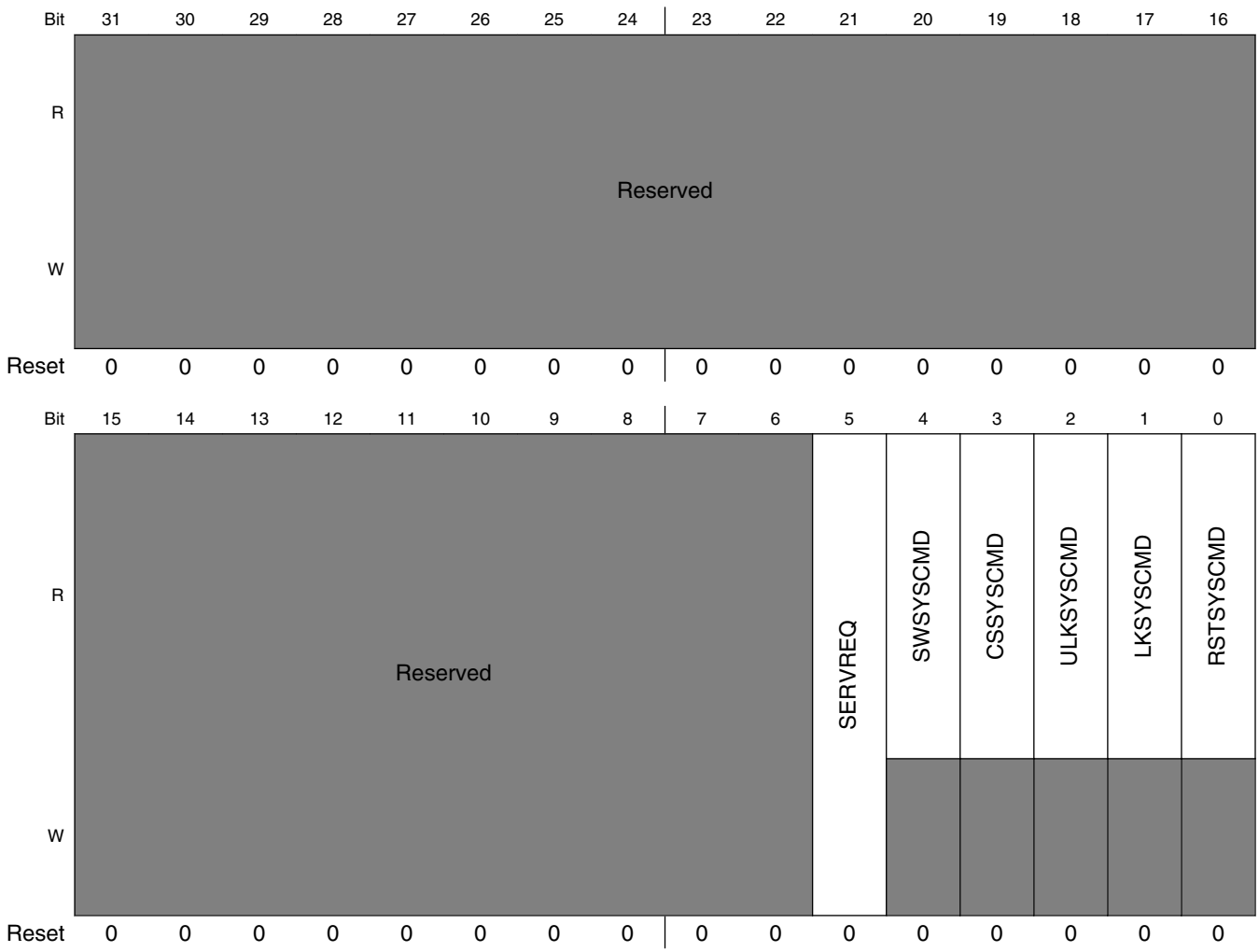


MLB_MS1 field descriptions

Field	Description
MCS_63_32	MediaLB channel status. Indicates the channel status for MediaLB channels 63 to 32. Channel status bits are set by hardware and cleared by software. Status is only set if the appropriate bits in the MIEN register are set.

43.6.5 MediaLB System Status Register (MLB_MSS)

Address: 218_C000h base + 20h offset = 218_C020h



MLB_MSS field descriptions

Field	Description
31–6 -	This field is reserved. Reserved
5 SERVREQ	Service request enabled. When set, the MediaLB block responds with a "device present, request service" system response if a matching channel scan system command is detected. When cleared, the MediaLB block responds with a "device present" system response.
4 SWSYSCMD	Software system command detected (in the system quadlet). Set by hardware, cleared by software. Data is stored in the MSD register for this command.
3 CSSYSCMD	Channel scan system command detected (in the system quadlet). Set by hardware, cleared by software. If the node address specified in *Data* quadlet matches the value in **MLBC1.NDA**, the device responds either "device present" or "device present, request service" system response in the next system quadlet.

Table continues on the next page...

MLB_MSS field descriptions (continued)

Field	Description
2 ULKSYSCMD	Network unlock system command detected (in the system quadlet). Set by hardware, cleared by software.
1 LKSYSCMD	Network lock system command detected (in the system quadlet). Set by hardware, cleared by software.
0 RSTSYSCMD	Reset system command detected (in the system quadlet). Set by hardware, cleared by software.

43.6.6 MediaLB System Data Register (MLB_MSD)

Address: 218_C000h base + 24h offset = 218_C024h

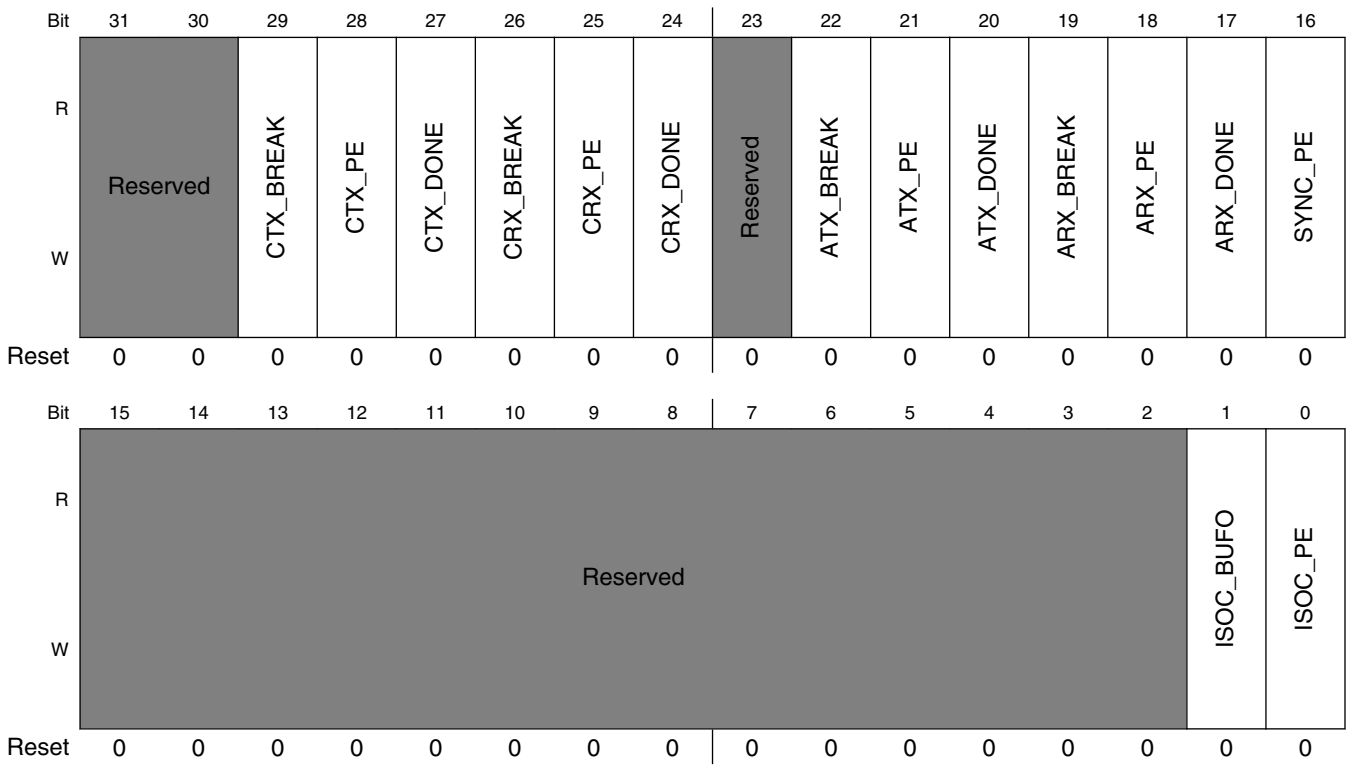
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SD3_7_0								SD2_7_0								SD1_7_0								SD0_7_0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MSD field descriptions

Field	Description
31–24 SD3_7_0	System data (byte 3). Updated with MediaLB Data[31:24] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD3 is not updated. (read-only)
23–16 SD2_7_0	System data (byte 2). Updated with MediaLB Data[23:16] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD2 is not updated. (read-only)
15–8 SD1_7_0	System data (byte 1). Updated with MediaLB Data[15:8] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD1 is not updated. (read-only)
SD0_7_0	System data (byte 0). Updated with MediaLB Data[7:0] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD0 is not updated. (read-only)

43.6.7 MediaLB Interrupt Enable Register (MLB_MIEN)

Address: 218_C000h base + 2Ch offset = 218_C02Ch



MLB_MIEN field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29 CTX_BREAK	Control Tx break enable. When set, a *ReceiverBreak* response received from the receiver on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
28 CTX_PE	Control Tx protocol error enable. When set, a *ProtocolError* generated by the receiver on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
27 CTX_DONE	Control Tx packet done enable. When set, a packet transmitted with no errors on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
26 CRX_BREAK	Control Rx break enable. When set, a *ControlBreak* command received from the transmitter on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
25 CRX_PE	Control Rx protocol error enable. When set, a *ProtocolError* detected on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
24 CRX_DONE	Control Rx packet done enable. When set, a packet received with no errors on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
23 -	This field is reserved. Reserved
22 ATX_BREAK	Asynchronous Tx break enable. When set, a *ReceiverBreak* response received from the receiver on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.

Table continues on the next page...

MLB_MIEN field descriptions (continued)

Field	Description
21 ATX_PE	Asynchronous Tx protocol error enable. When set, a *ProtocolError* generated by the receiver on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
20 ATX_DONE	Asynchronous Tx packet done enable. When set, a packet transmitted with no errors on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
19 ARX_BREAK	Asynchronous Rx break enable. When set, a *AsyncBreak* command received from the transmitter on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
18 ARX_PE	Asynchronous Rx protocol error enable. When set, a *ProtocolError* detected on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
17 ARX_DONE	Asynchronous Rx done enable. When set, a packet received with no errors on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
16 SYNC_PE	Synchronous protocol error enable. When set, a *ProtocolError* detected on a synchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
15–2 -	This field is reserved. Reserved
1 ISOC_BUFO	Isochronous Rx buffer overflow enable. When set, a buffer overflow on an isochronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. This occurs only when isochronous flow control is disabled.
0 ISOC_PE	Isochronous Rx protocol error enable. When set, a *ProtocolError* detected on an isochronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.

43.6.8 MediaLB Control 1 Register (MLB_MLBC1)

Address: 218_C000h base + 3Ch offset = 218_C03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
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	NDA_7_0								CLKM	LOCK	Reserved					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MLBC1 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 NDA_7_0	Node device address. Used for system commands directed to individual MediaLB nodes.
7 CLKM	MediaLB clock missing status. Set when MLB_CLK (MediaLB clock) is not toggling at the pin; cleared by software.
6 LOCK	MediaLB lock error status. Set when MediaLB is unlocked; cleared by software.
-	This field is reserved. Reserved

43.6.9 HBI Control Register (MLB_HCTL)

The HC can control and monitor general operation of the HBI block by reading and writing the HBI Control Register (HCTL) through the I/O interface. Each bit of HCTL is read/write.

Address: 218_C000h base + 80h offset = 218_C080h

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	EN	Reserved													RST1	RST0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_HCTL field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 EN	HBI enable 1 enabled 0 disabled
14–2 -	This field is reserved. Reserved
1 RST1	AGU1 software reset 1 reset 0 active

Table continues on the next page...

MLB_HCTL field descriptions (continued)

Field	Description
0 RST0	AGU0 software reset 1 reset 0 active

43.6.10 HBI Channel Mask 0 Register (MLB_HCMR0)

The HC can control which channel(s) are able to generate an HBI interrupt by writing the HBI Channel Mask Registers (HCMRn). The HCMRn registers mask the channel interrupt on the *hbi_hintb* signal (i.e. *hbi_hintb* will not become active for any masked channel). Each bit of HCMRn is read/write.

Address: 218_C000h base + 88h offset = 218_C088h

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	CHM_31_0_P																															
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

MLB_HCMR0 field descriptions

Field	Description
CHM_31_0_P	Bitwise channel mask bit 0 masked 1 unmasked

43.6.11 HBI Channel Mask 1 Register (MLB_HCMR1)

Address: 218_C000h base + 8Ch offset = 218_C08Ch

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	CHM_63_32																															
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

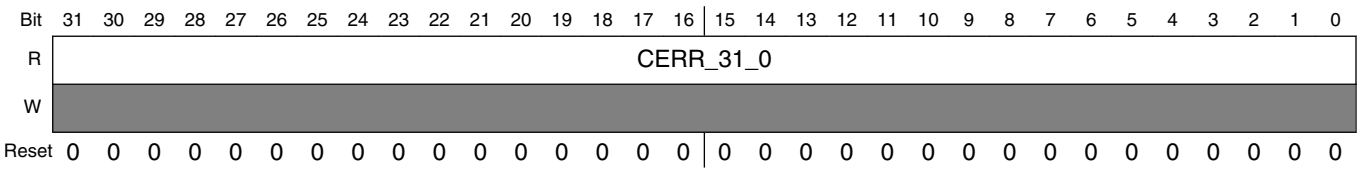
MLB_HCMR1 field descriptions

Field	Description
CHM_63_32	Bitwise channel mask bit 0 masked 1 unmasked

43.6.12 HBI Channel Error 0 Register (MLB_HCER0)

The HBI Channel Error Registers (HCERn) indicate which channel(s) have encountered fatal errors.

Address: 218_C000h base + 90h offset = 218_C090h



MLB_HCER0 field descriptions

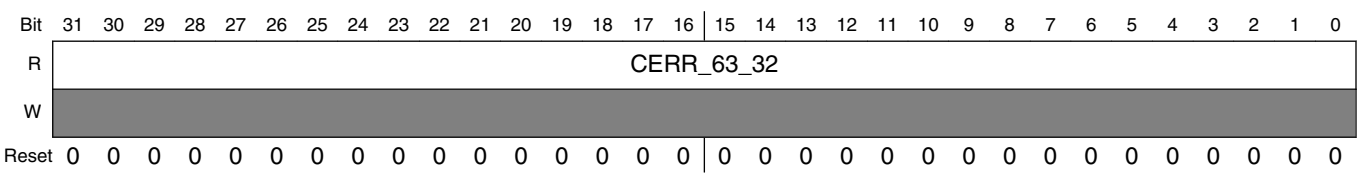
Field	Description
CERR_31_0	Bitwise channel error bit

43.6.13 HBI Channel Error 1 Register (MLB_HCER1)

HCERn status bits are set when hardware detects hardware errors on the given logical channel, including:

- Channel opened, but not enabled,
- Channel programmed with invalid channel type, or
- Out-of-range PML for asynchronous or control Tx channels

Address: 218_C000h base + 94h offset = 218_C094h



MLB_HCER1 field descriptions

Field	Description
CERR_63_32	Bitwise channel error bit

43.6.14 HBI Channel Busy 0 Register (MLB_HCBR0)

The HC can determine which channel(s) are busy by reading the HBI Channel Busy Registers (HCBRn). An HBI channel is busy if:

- it is currently loaded into one of the two AGUs
- the channel is enabled, CE = 1 from the Channel Allocation Table (Table 43-5), and
- the DMA is active

When an HBI channel is busy, hardware may write back its local copy of the channel descriptor at any time. System software should not write a CDT descriptor for a channel that is busy. Only two HBI channels can be busy at any given time. Each bit of HCBRn is read-only.

Address: 218_C000h base + 98h offset = 218_C098h

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	CHB_31_0																															
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

MLB_HCBR0 field descriptions

Field	Description
CHB_31_0	Bitwise channel busy bit 0 idle 1 busy

43.6.15 HBI Channel Busy 1 Register (MLB_HCBR1)

Address: 218_C000h base + 9Ch offset = 218_C09Ch

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	CHB_63_32																															
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

MLB_HCBR1 field descriptions

Field	Description
CHB_63_32	Bitwise channel busy bit 0 idle 1 busy

43.6.16 MIF Data 0 Register (MLB_MDAT0)

Address: 218_C000h base + C0h offset = 218_C0C0h

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	DATA_31_0																															
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

MLB_MDAT0 field descriptions

Field	Description
DATA_31_0	CTR data - bits[31:0] of 128-bit entry or DBR data - bits[7:0] of 8-bit entry

43.6.17 MIF Data 1 Register (MLB_MDAT1)

Address: 218_C000h base + C4h offset = 218_C0C4h

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	DATA_63_32																															
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

MLB_MDAT1 field descriptions

Field	Description
DATA_63_32	CTR data - bits[63:32] of 128-bit entry

43.6.18 MIF Data 2 Register (MLB_MDAT2)

Address: 218_C000h base + C8h offset = 218_C0C8h

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	DATA_95_64																															
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

MLB_MDAT2 field descriptions

Field	Description
DATA_95_64	CTR data - bits[95:64] of 128-bit entry

43.6.19 MIF Data 3 Register (MLB_MDAT3)

Address: 218_C000h base + CCh offset = 218_C0CCh

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	DATA_127_96																															
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

MLB_MDAT3 field descriptions

Field	Description
DATA_127_96	CTR data - bits[127:96] of 128-bit entry

43.6.20 MIF Data Write Enable 0 Register (MLB_MDWE0)

Address: 218_C000h base + D0h offset = 218_C0D0h

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	MASK_31_0																															
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

MLB_MDWE0 field descriptions

Field	Description
MASK_31_0	Bitwise write enable for CTR data - bits[31:0] 0 disabled 1 enabled

43.6.21 MIF Data Write Enable 1 Register (MLB_MDWE1)

Address: 218_C000h base + D4h offset = 218_C0D4h

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	MASK_63_32																															
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

MLB_MDWE1 field descriptions

Field	Description
MASK_63_32	Bitwise write enable for CTR data - bits[63:32] 0 disabled 1 enabled

43.6.22 MIF Data Write Enable 2 Register (MLB_MDWE2)

Address: 218_C000h base + D8h offset = 218_C0D8h

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	MASK_95_64																															
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

MLB_MDWE2 field descriptions

Field	Description
MASK_95_64	Bitwise write enable for CTR data - bits[95:64] 0 disabled 1 enabled

43.6.23 MIF Data Write Enable 3 Register (MLB_MDWE3)

Address: 218_C000h base + DCh offset = 218_C0DCh

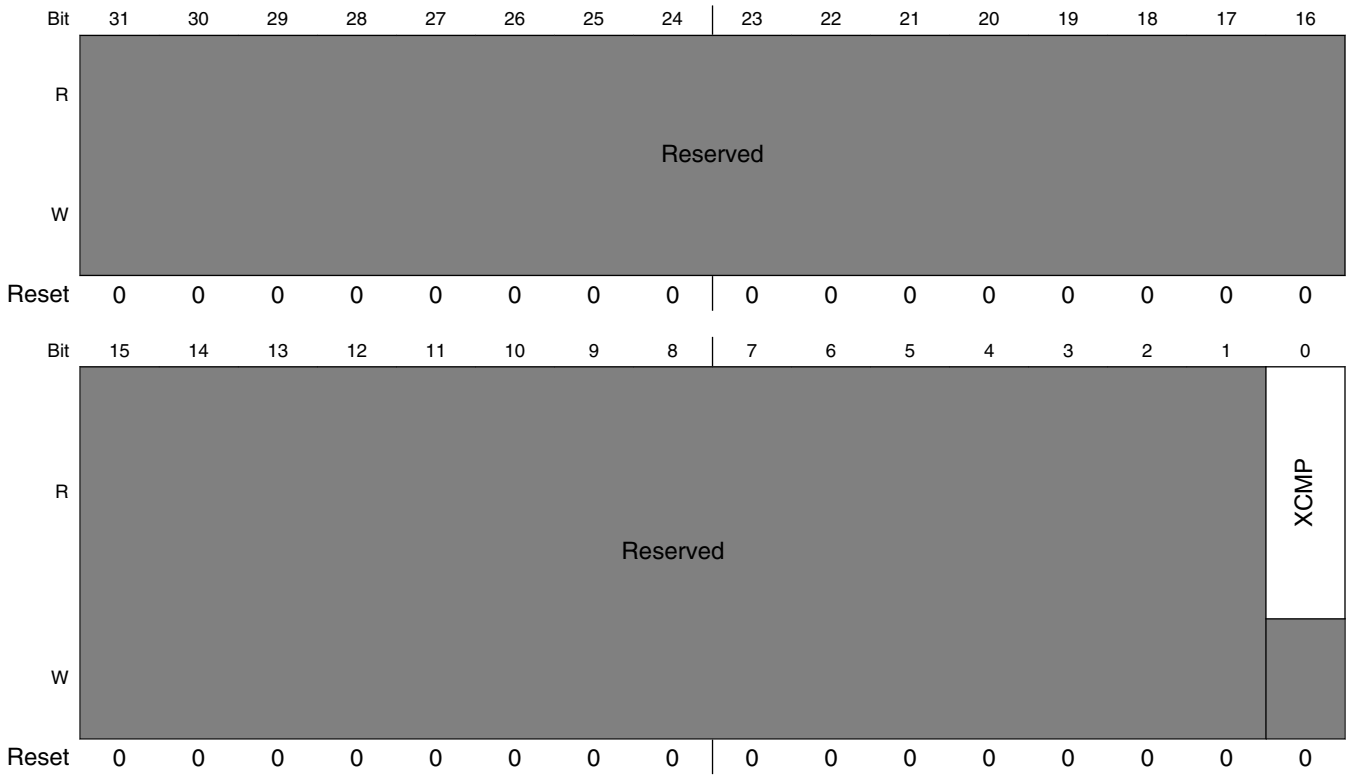
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_127_96																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDWE3 field descriptions

Field	Description
MASK_127_96	Bitwise write enable for CTR data - bits[127:96] 0 disabled 1 enabled

43.6.24 MIF Control Register (MLB_MCTL)

Address: 218_C000h base + E0h offset = 218_C0E0h

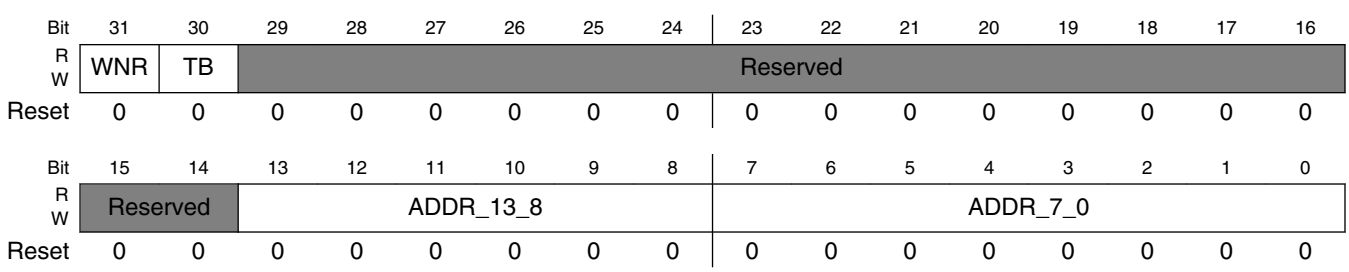


MLB_MCTL field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 XCMP	Transfer complete (write 0 to clear)

43.6.25 MIF Address Register (MLB_MADR)

Address: 218_C000h base + E4h offset = 218_C0E4h



MLB_MADR field descriptions

Field	Description
31 WNR	Write-Not-Read selection 0 read 1 write
30 TB	Target location bit 0 selects CTR 1 selects DBR
29–14 -	This field is reserved. Reserved
13–8 ADDR_13_8	DBR address of 8-bit entry - bits[13:8]
ADDR_7_0	CTR address of 128-bit entry or DBR address of 8-bit entry - bits[7:0]

43.6.26 AHB Control Register (MLB_ACTL)

The AHB Control (ACTL) register is written by the HC to configure the AMBA AHB block for channel interrupts. ACTL contains three configuration fields, one is used to select the DMA mode, one is used to mux channel interrupts onto a single interrupt signal, and the last selects the method of clearing channel interrupts (either software or hardware).

Address: 218_C000h base + 3C0h offset = 218_C3C0h

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											MPB	-	DMA_MODE	SMX	SCE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_ACTL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 MPB	DMA Packet buffering mode. 0 Single-packet mode 1 Multiple-packet mode
3 -	Reserved.
2 DMA_MODE	DMA Mode: 0 DMA Mode 0 1 DMA Mode 1
1 SMX	AHB interrupt mux enable: 0 ACSR0 generates an interrupt on *ahb_int[0]*; ACSR1 generates an interrupt on *ahb_int[1]* 1 ACSR0 and ACSR1 generate an interrupts on *ahb_int[0]* only
0 SCE	Software clear enable: 0 Hardware clears interrupt after a ACSRn register read 1 Software clears interrupt

43.6.27 AHB Channel Status 0 Register (MLB_ACSR0)

The AHB Channel Status (ACSRn) registers contain interrupt bits for each of the 64 physical channels. When an ACSRn register bit is set, it indicates that the corresponding physical channel has an interrupt pending.

An AHB interrupt is triggered when either DNEn or ERRn is set within the AHB Channel Descriptor. The HC is notified of the channel interrupt via *ahb_int[1:0]*. When an interrupt occurs in ACSR0 (for channels 31 to 0) *ahb_int[0]* is set. When an interrupt occurs in ACSR1 (for channels 63 to 32) *ahb_int[1]* is set.

Interrupts in ACSR0 and ACSR1 can be optionally muxed onto a single interrupt signal, *ahb_int[0]*, if ACTL.SMX = 1. If ACTL.SCE = 0, hardware automatically clears the interrupt bit(s) after the HC reads the ACSRn register. Alternatively, if ACTL.SCE = 1, software must write a 1 to the appropriate bit(s) of ACSRn to clear the interrupt(s).

Address: 218_C000h base + 3D0h offset = 218_C3D0h

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	CHS																															
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

MLB_ACSR0 field descriptions

Field	Description
CHS	Interrupt status for logical channels 31 to 0: 0 None 1 Interrupt

43.6.28 AHB Channel Status 1 Register (MLB_ACSR1)

Address: 218_C000h base + 3D4h offset = 218_C3D4h

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	CHS[63:32]																															
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

MLB_ACSR1 field descriptions

Field	Description
CHS[63:32]	Interrupt status for logical channels 63 to 32: 0 None 1 Interrupt

43.6.29 AHB Channel Mask 0 Register (MLB_ACMR0)

Using the AHB Channel Mask (ACMRn) register, the HC can control which channel(s) generate interrupts on *ahb_int[1:0]*. All ACMRn register bits default as '0' ("masked"); therefore, the HC must initially write ACMRn to enable interrupts. Each bit of ACMRn is read/write accessible.

Address: 218_C000h base + 3D8h offset = 218_C3D8h

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	CHM_31_0																															
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

MLB_ACMR0 field descriptions

Field	Description
CHM_31_0	Bitwise channel mask bit:

MLB_ACMR0 field descriptions (continued)

Field	Description
	0 Masked 1 Unmasked

43.6.30 AHB Channel Mask 1 Register (MLB_ACMR1)

Address: 218_C000h base + 3DCh offset = 218_C3DCh

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	CHM[63:32]																															
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

MLB_ACMR1 field descriptions

Field	Description
CHM[63:32]	Bitwise channel mask bit: 0 Masked 1 Unmasked