RW-WLAN nX PLATFORM DMA

Functional Specification

RW-WLAN-nX-PLF-DMA-FS

Version 1.05

2019-01-04

RivieraWaves
http://www.rivierawaves.com/

Revision History

Version	Date	Revision Description	Author
0.01	2014-09-16	Initial release	JV, OR
1.01	2015-10-07	Upstream channel: - Dini Interface replaced by AXI4 interface Downstream channel: - Dini interface replaced by AXI4 interface - AHB port added Midstream channel added (Shared Ram / AHB)	LV
1.02	2016-05-09	AXI_BURST_LENGTH_LIMIT register added AXI Interface updated.	LV
1.03	2017-04-04	Updated Channel 0-4 root pointer registers description.	LV
1.04	2017-07-07	Updated block diagram, Removed reference to Dini, Updated register description and format	JV
1.05	2017-07-17	Corrected typo in DADDR	JV

Table of Contents

List of Figures

List of Tables

1 Overview

1.1 Document Overview

The document describes the software and hardware interfaces of the RW-WLAN-NX DMA.

1.2 Architecture Overview

The following diagram shows the RW-WLAN-NX DMA and its integration in the platform.

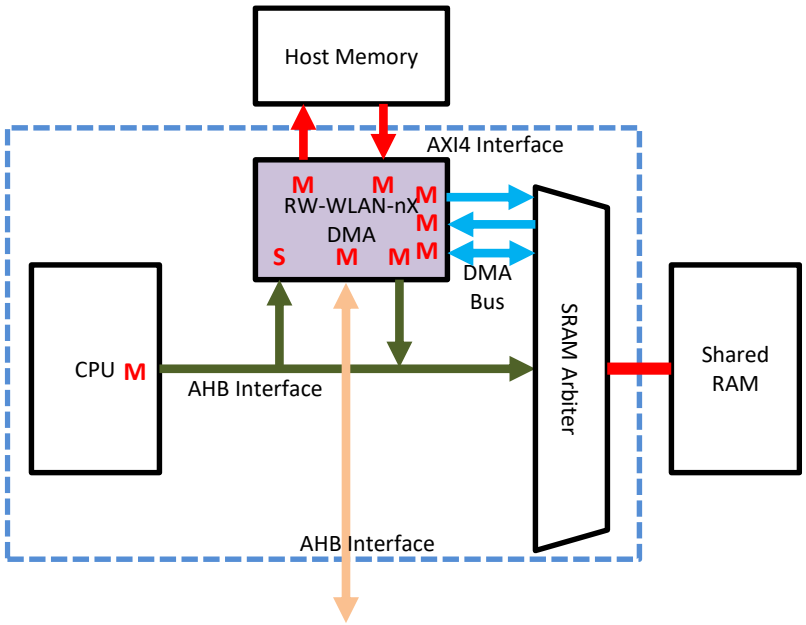

Figure 1 - RW-WLAN-nX DMA

1.3 About wording

1. The words of host or application sub-system are used to designate the processor sub-system executing the network applications, the mac80211 and the RW-WLAN nX driver.
2. The words of the embedded platform, reference platform or firmware sub-system are used to designate the processor sub-system executing the RW-WLAN nX firmware.
3. DMA descriptor or link list item (LLI) word designates the same thing, a control structure defining a fragment.
4. A fragment is a contiguous data quantity described by a DMA descriptor.
5. A frame is a set of fragments and is described by a list of DMA descriptors.

2 Software interface

Some registers or fields in registers have been defined for debug purposes alone and will be read/set only through the alternate debug interface to the platform, not by SW.

2.1 Registers

2.1.1 Register Map

The following table summarizes the registers displayed by the DMA device.

Address	Register Name
0x0	CH_LLI_ROOT0
0x4	CH_LLI_ROOT1
0x8	CH_LLI_ROOT2
0xC	CH_LLI_ROOT3
0x10	DMA_STATUS
0x14	INT_RAWSTATUS
0x18	INT_UNMASK_SET
0x1C	INT_UNMASK_CLEAR
0x20	INT_ACK
0x24	INT_STATUS
0x34	ARBITRATION
0x38	CHANNEL_MUTEX_SET
0x3C	CHANNEL_MUTEX_CLEAR
0x40	CH_LLI_ROOT4
0x44	AXI_BURST_LENGTH_LIMIT
0x80	LLI_COUNTER0
0x84	LLI_COUNTER1
0x88	LLI_COUNTER2
0x8C	LLI_COUNTER3
0x90	LLI_COUNTER4
0x94	LLI_COUNTER5
0x98	LLI_COUNTER6
0x9C	LLI_COUNTER7
0xA0	LLI_COUNTER8
0xA4	LLI_COUNTER9
0xA8	LLI_COUNTER10
0xAC	LLI_COUNTER11
0xB0	LLI_COUNTER12
0xB4	LLI_COUNTER13
0xB8	LLI_COUNTER14
0xBC	LLI_COUNTER15

Table 2.1 - Register Map

Channels 0 and 1 are dedicated for DMA transfer from the embedded platform to the host memory.

Channels 2 and 3 are dedicated for DMA transfer from the host memory to the embedded platform.

Channels 4 are dedicated for DMA transfer between the shared ram and the platform AHB address space.

2.1.2 Register List

2.1.2.1 Register CH_LLI_ROOT0

The reference platform processor triggers a DMA between the shared RAM and the host memory when it writes to this register the address of the LLI node describing the transfer. The LLI node shall be located in the shared SRAM. The register is readable at any time. The hardware automatically updates the register indicating the next LLI node to be processed. The channel activity can be canceled by writing a null value to this register. Any non-null value written to this register is ignored while the channel is active.

Address	Access		CH_LLI_ROOT0																													
	HW	SW	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
+0'H	RW	RW	CH_LLI_ROOT0[31]	CH_LLI_ROOT0[30]	CH_LLI_ROOT0[29]	CH_LLI_ROOT0[28]	CH_LLI_ROOT0[27]	CH_LLI_ROOT0[26]	CH_LLI_ROOT0[25]	CH_LLI_ROOT0[24]	CH_LLI_ROOT0[23]	CH_LLI_ROOT0[22]	CH_LLI_ROOT0[21]	CH_LLI_ROOT0[20]	CH_LLI_ROOT0[19]	CH_LLI_ROOT0[18]	CH_LLI_ROOT0[17]	CH_LLI_ROOT0[16]	CH_LLI_ROOT0[15]	CH_LLI_ROOT0[14]	CH_LLI_ROOT0[13]	CH_LLI_ROOT0[12]	CH_LLI_ROOT0[11]	CH_LLI_ROOT0[10]	CH_LLI_ROOT0[9]	CH_LLI_ROOT0[8]	CH_LLI_ROOT0[7]	CH_LLI_ROOT0[6]	CH_LLI_ROOT0[5]	CH_LLI_ROOT0[4]	CH_LLI_ROOT0[3]	CH_LLI_ROOT0[2]	CH_LLI_ROOT0[1]	CH_LLI_ROOT0[0]
Reset Value			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
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 2.2 - Register CH_LLI_ROOT0

Name	Type	Size	Description
CH_LLI_ROOT0[31:0]	U	32	Pointer of the control structure describing the transfer.

Table 2.3 - Register CH_LLI_ROOT0 fields description

2.1.2.2 Register CH_LLI_ROOT1

The reference platform processor triggers a DMA between the shared RAM and the host memory when it writes to this register the address of the LLI node describing the transfer. The LLI node shall be located in the shared SRAM. The register is readable at any time. The hardware automatically updates the register indicating the next LLI node to be processed. The channel activity can be canceled by writing a null value to this register. Any non-null value written to this register is ignored while the channel is active.

Address	Access		CH_LLI_ROOT1																													
	HW	SW	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
+4'H	RW	RW	CH_LLI_ROOT1[31]	CH_LLI_ROOT1[30]	CH_LLI_ROOT1[29]	CH_LLI_ROOT1[28]	CH_LLI_ROOT1[27]	CH_LLI_ROOT1[26]	CH_LLI_ROOT1[25]	CH_LLI_ROOT1[24]	CH_LLI_ROOT1[23]	CH_LLI_ROOT1[22]	CH_LLI_ROOT1[21]	CH_LLI_ROOT1[20]	CH_LLI_ROOT1[19]	CH_LLI_ROOT1[18]	CH_LLI_ROOT1[17]	CH_LLI_ROOT1[16]	CH_LLI_ROOT1[15]	CH_LLI_ROOT1[14]	CH_LLI_ROOT1[13]	CH_LLI_ROOT1[12]	CH_LLI_ROOT1[11]	CH_LLI_ROOT1[10]	CH_LLI_ROOT1[9]	CH_LLI_ROOT1[8]	CH_LLI_ROOT1[7]	CH_LLI_ROOT1[6]	CH_LLI_ROOT1[5]	CH_LLI_ROOT1[4]	CH_LLI_ROOT1[3]	CH_LLI_ROOT1[2]	CH_LLI_ROOT1[1]	CH_LLI_ROOT1[0]
Reset Value			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
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 2.4 - Register CH_LLI_ROOT1

Name	Type	Size	Description
CH_LLI_ROOT1[31:0]	U	32	Pointer of the control structure describing the transfer.

Table 2.5 - Register CH_LLI_ROOT1 fields description

2.1.2.3 Register CH_LLI_ROOT2

The reference platform processor triggers a DMA between the shared RAM and the host memory when it writes to this register the address of the LLI node describing the transfer. The LLI node shall be located in the shared SRAM. The register is readable at any time. The hardware automatically updates the register indicating the next LLI node to be processed. The channel activity can be canceled by writing a null value to this register. Any non-null value written to this register is ignored while the channel is active.

Address	Access		CH_LLI_ROOT2																													
	HW	SW	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
+8'H	RW	RW	CH_LLI_ROOT2[31]	CH_LLI_ROOT2[30]	CH_LLI_ROOT2[29]	CH_LLI_ROOT2[28]	CH_LLI_ROOT2[27]	CH_LLI_ROOT2[26]	CH_LLI_ROOT2[25]	CH_LLI_ROOT2[24]	CH_LLI_ROOT2[23]	CH_LLI_ROOT2[22]	CH_LLI_ROOT2[21]	CH_LLI_ROOT2[20]	CH_LLI_ROOT2[19]	CH_LLI_ROOT2[18]	CH_LLI_ROOT2[17]	CH_LLI_ROOT2[16]	CH_LLI_ROOT2[15]	CH_LLI_ROOT2[14]	CH_LLI_ROOT2[13]	CH_LLI_ROOT2[12]	CH_LLI_ROOT2[11]	CH_LLI_ROOT2[10]	CH_LLI_ROOT2[9]	CH_LLI_ROOT2[8]	CH_LLI_ROOT2[7]	CH_LLI_ROOT2[6]	CH_LLI_ROOT2[5]	CH_LLI_ROOT2[4]	CH_LLI_ROOT2[3]	CH_LLI_ROOT2[2]	CH_LLI_ROOT2[1]	CH_LLI_ROOT2[0]
Reset Value			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
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 2.6 - Register CH_LLI_ROOT2

Name	Type	Size	Description
CH_LLI_ROOT2[31:0]	U	32	Pointer of the control structure describing the transfer.

Table 2.7 - Register CH_LLI_ROOT2 fields description

2.1.2.4 Register CH_LLI_ROOT3

The reference platform processor triggers a DMA between the shared RAM and the host memory when it writes to this register the address of the LLI node describing the transfer. The LLI node shall be located in the shared SRAM. The register is readable at any time. The hardware automatically updates the register indicating the next LLI node to be processed. The channel activity can be canceled by writing a null value to this register. Any non-null value written to this register is ignored while the channel is active.

Address	Access		CH_LLI_ROOT3																													
	HW	SW	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
+C'H	RW	RW	CH_LLI_ROOT3[31]	CH_LLI_ROOT3[30]	CH_LLI_ROOT3[29]	CH_LLI_ROOT3[28]	CH_LLI_ROOT3[27]	CH_LLI_ROOT3[26]	CH_LLI_ROOT3[25]	CH_LLI_ROOT3[24]	CH_LLI_ROOT3[23]	CH_LLI_ROOT3[22]	CH_LLI_ROOT3[21]	CH_LLI_ROOT3[20]	CH_LLI_ROOT3[19]	CH_LLI_ROOT3[18]	CH_LLI_ROOT3[17]	CH_LLI_ROOT3[16]	CH_LLI_ROOT3[15]	CH_LLI_ROOT3[14]	CH_LLI_ROOT3[13]	CH_LLI_ROOT3[12]	CH_LLI_ROOT3[11]	CH_LLI_ROOT3[10]	CH_LLI_ROOT3[9]	CH_LLI_ROOT3[8]	CH_LLI_ROOT3[7]	CH_LLI_ROOT3[6]	CH_LLI_ROOT3[5]	CH_LLI_ROOT3[4]	CH_LLI_ROOT3[3]	CH_LLI_ROOT3[2]	CH_LLI_ROOT3[1]	CH_LLI_ROOT3[0]
Reset Value			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
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 2.8 - Register CH_LLI_ROOT3

Name	Type	Size	Description
CH_LLI_ROOT3[31:0]	U	32	Pointer of the control structure describing the transfer.

Table 2.9 - Register CH_LLI_ROOT3 fields description

2.1.2.5 Register DMA_STATUS

This register indicates miscellaneous information about the DMA state machines.

Address	Access		DMA_STATUS																													
	HW	SW	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
+10'H	W	R	CH4_STOPPED	MID_STREAM_BSY	DOWN_STREAM_BSY	UP_STREAM_BSY	ARB_Q4_VALID	ARB_Q3_VALID	ARB_Q2_VALID	ARB_Q1_VALID	ARB_Q0_VALID	REQUEST_STATE[2]	REQUEST_STATE[1]	REQUEST_STATE[0]	CH3_STOPPED	CH2_STOPPED	CH1_STOPPED	CH0_STOPPED	OFT_FREE[15]	OFT_FREE[14]	OFT_FREE[13]	OFT_FREE[12]	OFT_FREE[11]	OFT_FREE[10]	OFT_FREE[9]	OFT_FREE[8]	OFT_FREE[7]	OFT_FREE[6]	OFT_FREE[5]	OFT_FREE[4]	OFT_FREE[3]	OFT_FREE[2]	OFT_FREE[1]	OFT_FREE[0]
Reset Value			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
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.10 - Register DMA_STATUS

Name	Type	Size	Description
CH4_STOPPED	U	1	channel 4 is stopped because it has reached the last node of the descriptor list while the corresponding channel spinlock bit has been set by the software
MID_STREAM_BSY	U	1	This bit indicates that the midstream_data-path can't process new midstream request.
DOWN_STREAM_BSY	U	1	This bit indicates that the upstream_data-path can't process new upstream request.
UP_STREAM_BSY	U	1	This bit indicates that the downstream_data-path can't process new downstream request.
ARB_Q4_VALID	U	1	This bit indicates if the Queue 3 has a valid entry
ARB_Q3_VALID	U	1	This bit indicates if the Queue 3 has a valid entry
ARB_Q2_VALID	U	1	This bit indicates if the Queue 2 has a valid entry
ARB_Q1_VALID	U	1	This bit indicates if the Queue 1 has a valid entry
ARB_Q0_VALID	U	1	This bit indicates if the Queue 0 has a valid entry
REQUEST_STATE[2:0]	U	3	This field indicates the state of the LLI state machine: 0=IDLE 1=ADDR 2=D0 3=D1 4=D2 5=D3 6=REQUEST 7=ACCEPT
CH3_STOPPED	U	1	channel 3 is stopped because it has reached the last node of the descriptor list while the corresponding channel spinlock bit has been set by the software
CH2_STOPPED	U	1	channel 2 is stopped because it has reached the last node of the descriptor list while the corresponding channel spinlock bit has been set by the software

CH1_STOPPED	U	1	channel 1 is stopped because it has reached the last node of the descriptor list while the corresponding channel spinlock bit has been set by the software
CH0_STOPPED	U	1	channel 0 is stopped because it has reached the last node of the descriptor list while the corresponding channel spinlock bit has been set by the software
OFT_FREE[15:0]	U	16	This field indicates the free entries of the outstanding fragment table.

Table 2.11 - Register DMA_STATUS fields description

2.1.2.6 Register INT_RAWSTATUS

This register indicates the DMA active interrupt sources. This register is not affected by the mask applied on interrupt sources. chn_EOT bit indicates that the logic channel has finished the processing of its assigned linked list and that the latest fragment has been transferred. error bit indicates that the DMA fragment management state machine is desynchronized with the host sub-system. This should never happen and is provided for software assertion and debug purpose. lli_irqn bit indicate that a fragment with a descriptor having its lli_irq set to n has been transferred.

Address	Access		INT_RAWSTATUS																													
	HW	SW	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
+14'H	W	R								CH4_EOT	CH3_EOT	CH2_EOT	CH1_EOT	CH0_EOT				ERROR	LLI_IRQ[15]	LLI_IRQ[14]	LLI_IRQ[13]	LLI_IRQ[12]	LLI_IRQ[11]	LLI_IRQ[10]	LLI_IRQ[9]	LLI_IRQ[8]	LLI_IRQ[7]	LLI_IRQ[6]	LLI_IRQ[5]	LLI_IRQ[4]	LLI_IRQ[3]	LLI_IRQ[2]	LLI_IRQ[1]	LLI_IRQ[0]
Reset Value										0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type										U	U	U	U	U				U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access										W	W	W	W	W				W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access										R	R	R	R	R				R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.12 - Register INT_RAWSTATUS

Name	Type	Size	Description
CH4_EOT	U	1	channel4 interrupt enable
CH3_EOT	U	1	channel3 interrupt enable
CH2_EOT	U	1	channel2 interrupt enable
CH1_EOT	U	1	channel1 interrupt enable
CH0_EOT	U	1	channel0 interrupt enable
ERROR	U	1	Error interrupt enable
LLI_IRQ[15:0]	U	16	LLI interrupts enable

Table 2.13 - Register INT_RAWSTATUS fields description

2.1.2.7 Register INT_UNMASK_SET

This register enables the assertion of the DMA interrupt line by the different interrupt sources. It is a set register, setting a bit to one enables the corresponding sources while writing 0 has no effect.

Address	Access		INT_UNMASK_SET																													
	HW	SW	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
+18'H	RW	S								CH4_EOT	CH3_EOT	CH2_EOT	CH1_EOT	CH0_EOT				ERROR	LLI_IRQ[15]	LLI_IRQ[14]	LLI_IRQ[13]	LLI_IRQ[12]	LLI_IRQ[11]	LLI_IRQ[10]	LLI_IRQ[9]	LLI_IRQ[8]	LLI_IRQ[7]	LLI_IRQ[6]	LLI_IRQ[5]	LLI_IRQ[4]	LLI_IRQ[3]	LLI_IRQ[2]	LLI_IRQ[1]	LLI_IRQ[0]
Reset Value										0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type										U	U	U	U	U				U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access										RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

SW Access								S	S	S	S	S				S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S

Table 2.14 - Register INT_UNMASK_SET

Name	Type	Size	Description
CH4_EOT	U	1	channel4 interrupt disable
CH3_EOT	U	1	channel3 interrupt disable
CH2_EOT	U	1	channel2 interrupt disable
CH1_EOT	U	1	channel1 interrupt disable
CH0_EOT	U	1	channel0 interrupt disable
ERROR	U	1	Error interrupt disable
LLI_IRQ[15:0]	U	16	LLI interrupts disable

Table 2.15 - Register INT_UNMASK_SET fields description

2.1.2.8 Register INT_UNMASK_CLEAR

This register disables the assertion of the DMA interrupt line by the different interrupt sources. It is a set register, setting a bit to one disables the corresponding sources while writing 0 has no effect.

Address	Access		INT_UNMASK_CLEAR																													
	HW	SW	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
+1C'H	RW	C								CH4_EOT	CH3_EOT	CH2_EOT	CH1_EOT	CH0_EOT				ERROR	LLI_IRQ[15]	LLI_IRQ[14]	LLI_IRQ[13]	LLI_IRQ[12]	LLI_IRQ[11]	LLI_IRQ[10]	LLI_IRQ[9]	LLI_IRQ[8]	LLI_IRQ[7]	LLI_IRQ[6]	LLI_IRQ[5]	LLI_IRQ[4]	LLI_IRQ[3]	LLI_IRQ[2]	LLI_IRQ[1]	LLI_IRQ[0]
Reset Value										0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type										U	U	U	U	U				U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access										RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
SW Access										C	C	C	C	C				C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C

Table 2.16 - Register INT_UNMASK_CLEAR

Name	Type	Size	Description
CH4_EOT	U	1	channel4 interrupt acknowledge
CH3_EOT	U	1	channel3 interrupt acknowledge
CH2_EOT	U	1	channel2 interrupt acknowledge
CH1_EOT	U	1	channel1 interrupt acknowledge
CH0_EOT	U	1	channel0 interrupt acknowledge
ERROR	U	1	Error interrupt acknowledge
LLI_IRQ[15:0]	U	16	LLI interrupts acknowledge

Table 2.17 - Register INT_UNMASK_CLEAR fields description

2.1.2.9 Register INT_ACK

This register reset the different interrupt sources. It is a set register, setting a bit to one reset the corresponding sources while writing 0 has no effect.

Address	Access		INT_ACK																													
	HW	SW	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

+20'H	RW	C								CH4_EOT	CH3_EOT	CH2_EOT	CH1_EOT	CH0_EOT				ERROR	LLI_IRQ[15]	LLI_IRQ[14]	LLI_IRQ[13]	LLI_IRQ[12]	LLI_IRQ[11]	LLI_IRQ[10]	LLI_IRQ[9]	LLI_IRQ[8]	LLI_IRQ[7]	LLI_IRQ[6]	LLI_IRQ[5]	LLI_IRQ[4]	LLI_IRQ[3]	LLI_IRQ[2]	LLI_IRQ[1]	LLI_IRQ[0]
Reset Value										0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type										U	U	U	U	U				U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access										RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
SW Access										C	C	C	C	C				C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C

Table 2.18 - Register INT_ACK

Name	Type	Size	Description
CH4_EOT	U	1	channel4 interrupt is pending and un-masked
CH3_EOT	U	1	channel3 interrupt is pending and un-masked
CH2_EOT	U	1	channel2 interrupt is pending and un-masked
CH1_EOT	U	1	channel1 interrupt is pending and un-masked
CH0_EOT	U	1	channel0 interrupt is pending and un-masked
ERROR	U	1	Error interrupt is pending and un-masked
LLI_IRQ[15:0]	U	16	LLI interrupts is pending and un-masked

Table 2.19 - Register INT_ACK fields description

2.1.2.10 Register INT_STATUS

This register indicates the DMA pending interrupts. An interrupt is pending whever it's source is active and un-masked.

Address	Access		INT_STATUS																															
	HW	SW	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
+24'H	W	R								CH4_EOT	CH3_EOT	CH2_EOT	CH1_EOT	CH0_EOT				ERROR	LLI_IRQ[15]	LLI_IRQ[14]	LLI_IRQ[13]	LLI_IRQ[12]	LLI_IRQ[11]	LLI_IRQ[10]	LLI_IRQ[9]	LLI_IRQ[8]	LLI_IRQ[7]	LLI_IRQ[6]	LLI_IRQ[5]	LLI_IRQ[4]	LLI_IRQ[3]	LLI_IRQ[2]	LLI_IRQ[1]	LLI_IRQ[0]
Reset Value										0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type										U	U	U	U	U				U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access										W	W	W	W	W				W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access										R	R	R	R	R				R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.20 - Register INT_STATUS

Name	Type	Size	Description
CH4_EOT	U	1	channel4 interrupt is pending and un-masked
CH3_EOT	U	1	channel3 interrupt is pending and un-masked
CH2_EOT	U	1	channel2 interrupt is pending and un-masked
CH1_EOT	U	1	channel1 interrupt is pending and un-masked
CH0_EOT	U	1	channel0 interrupt is pending and un-masked
ERROR	U	1	Error interrupt is pending and un-masked
LLI_IRQ[15:0]	U	16	LLI interrupts is pending and un-masked

Table 2.21 - Register INT_STATUS fields description

2.1.2.11 Register ARBITRATION

This register limits the tag usage of the downstream datapath. It prevents the tag starvation of the upstream data path. It is recommended to not change the default value.

Address	Access		ARBITRATION																													
	HW	SW	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
+34'H	R	RW																													DOWNSTREAM_TAG_USAGE[3]	DOWNSTREAM_TAG_USAGE[2]	DOWNSTREAM_TAG_USAGE[1]	DOWNSTREAM_TAG_USAGE[0]
Reset Value																															1	1	0	0
Type																															U	U	U	U
HW Access																															R	R	R	R
SW Access																															RW	RW	RW	RW

Table 2.22 - Register ARBITRATION

Name	Type	Size	Description
DOWNSTREAM_TAG_USAGE[3:0]	U	4	Tag usage of the downstream datapath.

Table 2.23 - Register ARBITRATION fields description

2.1.2.12 Register CHANNEL_MUTEX_SET

The firmware software can append further linked control structures to a link list that is currently processed by the DMA. This register provides one spinlock per list preventing race condition between a software update and the DMA engine. The software shall respect the following procedure to append a new LLI node to an active link list. 1. Set the spinlock bit of the list to be appended. 2. Read back the corresponding channel LLI root register. 3. If the read value is null, it means that the DMA has already read the last LLI node, and hence it's too late to append anything. The software shall write the corresponding channel root pointer register with the root of the nodes list to be appended. In this case, the spinlock has not been set by the hardware. 4. If the read value is non-null, it means that the DMA has not already parsed the last LLI node. The mutex has been set by the hardware and the DMA engine will not process a LLI node with its next pointer filled with a null value. The software can safely update the next pointer field of the last LLI node with the root of the new nodes list. Once done, the software shall clear the mutex bit. IMPORTANT: In the case 4), once the next pointer field updated, the software shall no modify anymore the content of the new list. If it is required, then it shall clear the mutex and re-start the procedure from point 1).

Address	Access		CHANNEL_MUTEX_SET																															
	HW	SW	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
+38'H	RW	S																												CH4_MUTEX	CH3_MUTEX	CH2_MUTEX	CH1_MUTEX	CH0_MUTEX
Reset Value																														0	0	0	0	0
Type																														U	U	U	U	U
HW Access																														RW	RW	RW	RW	RW
SW Access																														S	S	S	S	S

Table 2.24 - Register CHANNEL_MUTEX_SET

Name	Type	Size	Description
CH4_MUTEX	U	1	Semaphore for channel 4 link list clear.
CH3_MUTEX	U	1	Semaphore for channel 3 link list clear.
CH2_MUTEX	U	1	Semaphore for channel 2 link list clear.
CH1_MUTEX	U	1	Semaphore for channel 1 link list clear.
CH0_MUTEX	U	1	Semaphore for channel 0 link list clear.

Table 2.25 - Register CHANNEL_MUTEX_SET fields description

2.1.2.13 Register CHANNEL_MUTEX_CLEAR

This register enables the software to clear link list access spinlocks previously enabled with the register CHANNEL_MUTEX_SET.

Address	Access		CHANNEL_MUTEX_CLEAR																											
	HW	SW	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
+3C'H	RW	C																												CH4_MUTEX	CH3_MUTEX	CH2_MUTEX	CH1_MUTEX	CH0_MUTEX
Reset Value																														0	0	0	0	0
Type																														U	U	U	U	U
HW Access																														RW	RW	RW	RW	RW
SW Access																														C	C	C	C	C

Table 2.26 - Register CHANNEL_MUTEX_CLEAR

Name	Type	Size	Description
CH4_MUTEX	U	1	Semaphore for channel 4 link list clear.
CH3_MUTEX	U	1	Semaphore for channel 3 link list clear.
CH2_MUTEX	U	1	Semaphore for channel 2 link list clear.
CH1_MUTEX	U	1	Semaphore for channel 1 link list clear.
CH0_MUTEX	U	1	Semaphore for channel 0 link list clear.

Table 2.27 - Register CHANNEL_MUTEX_CLEAR fields description

2.1.2.14 Register CH_LLI_ROOT4

Pointer of the control structure describing the transfer.

Address	Access		CH_LLI_ROOT4																															
	HW	SW	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
+40'H	R	RW	CH_LLI_ROOT4[31]	CH_LLI_ROOT4[30]	CH_LLI_ROOT4[29]	CH_LLI_ROOT4[28]	CH_LLI_ROOT4[27]	CH_LLI_ROOT4[26]	CH_LLI_ROOT4[25]	CH_LLI_ROOT4[24]	CH_LLI_ROOT4[23]	CH_LLI_ROOT4[22]	CH_LLI_ROOT4[21]	CH_LLI_ROOT4[20]	CH_LLI_ROOT4[19]	CH_LLI_ROOT4[18]	CH_LLI_ROOT4[17]	CH_LLI_ROOT4[16]	CH_LLI_ROOT4[15]	CH_LLI_ROOT4[14]	CH_LLI_ROOT4[13]	CH_LLI_ROOT4[12]	CH_LLI_ROOT4[11]	CH_LLI_ROOT4[10]	CH_LLI_ROOT4[9]	CH_LLI_ROOT4[8]	CH_LLI_ROOT4[7]	CH_LLI_ROOT4[6]	CH_LLI_ROOT4[5]	CH_LLI_ROOT4[4]	CH_LLI_ROOT4[3]	CH_LLI_ROOT4[2]	CH_LLI_ROOT4[1]	CH_LLI_ROOT4[0]
Reset Value			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
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

SW Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 2.28 - Register CH_LLI_ROOT4

Name	Type	Size	Description
CH_LLI_ROOT4[31:0]	U	32	Pointer of the control structure describing the transfer.

Table 2.29 - Register CH_LLI_ROOT4 fields description

2.1.2.15 Register AXI_BURST_LENGTH_LIMIT

This register sets the burst max length on the AXI port.

Address	Access		AXI_BURST_LENGTH_LIMIT																													
	HW	SW	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
+44'H	R	RW																													AXI_BURST_LENGTH_LIMIT[3]	AXI_BURST_LENGTH_LIMIT[2]	AXI_BURST_LENGTH_LIMIT[1]	AXI_BURST_LENGTH_LIMIT[0]
Reset Value																															1	0	0	0
Type																															U	U	U	U
HW Access																															R	R	R	R
SW Access																															RW	RW	RW	RW

Table 2.30 - Register AXI_BURST_LENGTH_LIMIT

Name	Type	Size	Description
AXI_BURST_LENGTH_LIMIT[3:0]	U	4	This field limits the AXI burst length of the downstream/upstream AXI interface to 2axi_burst_length_limit transfers, up to 256. 0: 1 transfer 1: 2 transfers … 7: 128 transfers 8: 256 transfers 9-15: reserved This register must be modified only when DMA is IDLE.

Table 2.31 - Register AXI_BURST_LENGTH_LIMIT fields description

2.1.2.16 Register LLI_COUNTER0

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access	LLI_COUNTER0

	HW	SW	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
+80'H	W	R																	COUNTER0[15]	COUNTER0[14]	COUNTER0[13]	COUNTER0[12]	COUNTER0[11]	COUNTER0[10]	COUNTER0[9]	COUNTER0[8]	COUNTER0[7]	COUNTER0[6]	COUNTER0[5]	COUNTER0[4]	COUNTER0[3]	COUNTER0[2]	COUNTER0[1]	COUNTER0[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.32 - Register LLI_COUNTER0

Name	Type	Size	Description
COUNTER0[15:0]	U	16	Value of the LLI counter #n

Table 2.33 - Register LLI_COUNTER0 fields description

2.1.2.17 Register LLI_COUNTER1

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER1																															
	HW	SW	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
+84'H	W	R																	COUNTER1[15]	COUNTER1[14]	COUNTER1[13]	COUNTER1[12]	COUNTER1[11]	COUNTER1[10]	COUNTER1[9]	COUNTER1[8]	COUNTER1[7]	COUNTER1[6]	COUNTER1[5]	COUNTER1[4]	COUNTER1[3]	COUNTER1[2]	COUNTER1[1]	COUNTER1[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.34 - Register LLI_COUNTER1

Name	Type	Size	Description
COUNTER1[15:0]	U	16	Value of the LLI counter #n

Table 2.35 - Register LLI_COUNTER1 fields description

2.1.2.18 Register LLI_COUNTER2

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER2																															
	HW	SW	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
+88'H	W	R																	COUNTER2[15]	COUNTER2[14]	COUNTER2[13]	COUNTER2[12]	COUNTER2[11]	COUNTER2[10]	COUNTER2[9]	COUNTER2[8]	COUNTER2[7]	COUNTER2[6]	COUNTER2[5]	COUNTER2[4]	COUNTER2[3]	COUNTER2[2]	COUNTER2[1]	COUNTER2[0]

Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.36 - Register LLI_COUNTER2

Name	Type	Size	Description
COUNTER2[15:0]	U	16	Value of the LLI counter #n

Table 2.37 - Register LLI_COUNTER2 fields description

2.1.2.19 Register LLI_COUNTER3

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER3																											
	HW	SW	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
+8C'H	W	R																	COUNTER3[15]	COUNTER3[14]	COUNTER3[13]	COUNTER3[12]	COUNTER3[11]	COUNTER3[10]	COUNTER3[9]	COUNTER3[8]	COUNTER3[7]	COUNTER3[6]	COUNTER3[5]	COUNTER3[4]	COUNTER3[3]	COUNTER3[2]	COUNTER3[1]	COUNTER3[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.38 - Register LLI_COUNTER3

Name	Type	Size	Description
COUNTER3[15:0]	U	16	Value of the LLI counter #n

Table 2.39 - Register LLI_COUNTER3 fields description

2.1.2.20 Register LLI_COUNTER4

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER4																											
	HW	SW	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
+90'H	W	R																	COUNTER4[15]	COUNTER4[14]	COUNTER4[13]	COUNTER4[12]	COUNTER4[11]	COUNTER4[10]	COUNTER4[9]	COUNTER4[8]	COUNTER4[7]	COUNTER4[6]	COUNTER4[5]	COUNTER4[4]	COUNTER4[3]	COUNTER4[2]	COUNTER4[1]	COUNTER4[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.40 - Register LLI_COUNTER4

Name	Type	Size	Description
COUNTER4[15:0]	U	16	Value of the LLI counter #n

Table 2.41 - Register LLI_COUNTER4 fields description

2.1.2.21 Register LLI_COUNTER5

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER5																													
	HW	SW	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
+94'H	W	R																	COUNTER5[15]	COUNTER5[14]	COUNTER5[13]	COUNTER5[12]	COUNTER5[11]	COUNTER5[10]	COUNTER5[9]	COUNTER5[8]	COUNTER5[7]	COUNTER5[6]	COUNTER5[5]	COUNTER5[4]	COUNTER5[3]	COUNTER5[2]	COUNTER5[1]	COUNTER5[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.42 - Register LLI_COUNTER5

Name	Type	Size	Description
COUNTER5[15:0]	U	16	Value of the LLI counter #n

Table 2.43 - Register LLI_COUNTER5 fields description

2.1.2.22 Register LLI_COUNTER6

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER6																													
	HW	SW	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
+98'H	W	R																	COUNTER6[15]	COUNTER6[14]	COUNTER6[13]	COUNTER6[12]	COUNTER6[11]	COUNTER6[10]	COUNTER6[9]	COUNTER6[8]	COUNTER6[7]	COUNTER6[6]	COUNTER6[5]	COUNTER6[4]	COUNTER6[3]	COUNTER6[2]	COUNTER6[1]	COUNTER6[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.44 - Register LLI_COUNTER6

Name	Type	Size	Description
COUNTER6[15:0]	U	16	Value of the LLI counter #n

Table 2.45 - Register LLI_COUNTER6 fields description

2.1.2.23 Register LLI_COUNTER7

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER7																											
	HW	SW	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
+9C'H	W	R																	COUNTER7[15]	COUNTER7[14]	COUNTER7[13]	COUNTER7[12]	COUNTER7[11]	COUNTER7[10]	COUNTER7[9]	COUNTER7[8]	COUNTER7[7]	COUNTER7[6]	COUNTER7[5]	COUNTER7[4]	COUNTER7[3]	COUNTER7[2]	COUNTER7[1]	COUNTER7[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.46 - Register LLI_COUNTER7

Name	Type	Size	Description
COUNTER7[15:0]	U	16	Value of the LLI counter #n

Table 2.47 - Register LLI_COUNTER7 fields description

2.1.2.24 Register LLI_COUNTER8

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER8																											
	HW	SW	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
+A0'H	W	R																	COUNTER8[15]	COUNTER8[14]	COUNTER8[13]	COUNTER8[12]	COUNTER8[11]	COUNTER8[10]	COUNTER8[9]	COUNTER8[8]	COUNTER8[7]	COUNTER8[6]	COUNTER8[5]	COUNTER8[4]	COUNTER8[3]	COUNTER8[2]	COUNTER8[1]	COUNTER8[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.48 - Register LLI_COUNTER8

Name	Type	Size	Description
COUNTER8[15:0]	U	16	Value of the LLI counter #n

Table 2.49 - Register LLI_COUNTER8 fields description

2.1.2.25 Register LLI_COUNTER9

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER9																													
	HW	SW	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
+A4'H	W	R																	COUNTER9[15]	COUNTER9[14]	COUNTER9[13]	COUNTER9[12]	COUNTER9[11]	COUNTER9[10]	COUNTER9[9]	COUNTER9[8]	COUNTER9[7]	COUNTER9[6]	COUNTER9[5]	COUNTER9[4]	COUNTER9[3]	COUNTER9[2]	COUNTER9[1]	COUNTER9[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.50 - Register LLI_COUNTER9

Name	Type	Size	Description
COUNTER9[15:0]	U	16	Value of the LLI counter #n

Table 2.51 - Register LLI_COUNTER9 fields description

2.1.2.26 Register LLI_COUNTER10

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER10																													
	HW	SW	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
+A8'H	W	R																	COUNTER10[15]	COUNTER10[14]	COUNTER10[13]	COUNTER10[12]	COUNTER10[11]	COUNTER10[10]	COUNTER10[9]	COUNTER10[8]	COUNTER10[7]	COUNTER10[6]	COUNTER10[5]	COUNTER10[4]	COUNTER10[3]	COUNTER10[2]	COUNTER10[1]	COUNTER10[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.52 - Register LLI_COUNTER10

Name	Type	Size	Description
COUNTER10[15:0]	U	16	Value of the LLI counter #n

Table 2.53 - Register LLI_COUNTER10 fields description

2.1.2.27 Register LLI_COUNTER11

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER11																											
	HW	SW	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
+AC'H	W	R																	COUNTER11[15]	COUNTER11[14]	COUNTER11[13]	COUNTER11[12]	COUNTER11[11]	COUNTER11[10]	COUNTER11[9]	COUNTER11[8]	COUNTER11[7]	COUNTER11[6]	COUNTER11[5]	COUNTER11[4]	COUNTER11[3]	COUNTER11[2]	COUNTER11[1]	COUNTER11[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.54 - Register LLI_COUNTER11

Name	Type	Size	Description
COUNTER11[15:0]	U	16	Value of the LLI counter #n

Table 2.55 - Register LLI_COUNTER11 fields description

2.1.2.28 Register LLI_COUNTER12

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER12																											
	HW	SW	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
+B0'H	W	R																	COUNTER12[15]	COUNTER12[14]	COUNTER12[13]	COUNTER12[12]	COUNTER12[11]	COUNTER12[10]	COUNTER12[9]	COUNTER12[8]	COUNTER12[7]	COUNTER12[6]	COUNTER12[5]	COUNTER12[4]	COUNTER12[3]	COUNTER12[2]	COUNTER12[1]	COUNTER12[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.56 - Register LLI_COUNTER12

Name	Type	Size	Description
COUNTER12[15:0]	U	16	Value of the LLI counter #n

Table 2.57 - Register LLI_COUNTER12 fields description

2.1.2.29 Register LLI_COUNTER13

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER13																											
	HW	SW	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

+B4'H	W	R															COUNTER13[15]	COUNTER13[14]	COUNTER13[13]	COUNTER13[12]	COUNTER13[11]	COUNTER13[10]	COUNTER13[9]	COUNTER13[8]	COUNTER13[7]	COUNTER13[6]	COUNTER13[5]	COUNTER13[4]	COUNTER13[3]	COUNTER13[2]	COUNTER13[1]	COUNTER13[0]
Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.58 - Register LLI_COUNTER13

Name	Type	Size	Description
COUNTER13[15:0]	U	16	Value of the LLI counter #n

Table 2.59 - Register LLI_COUNTER13 fields description

2.1.2.30 Register LLI_COUNTER14

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER14																													
	HW	SW	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
+B8'H	W	R																	COUNTER14[15]	COUNTER14[14]	COUNTER14[13]	COUNTER14[12]	COUNTER14[11]	COUNTER14[10]	COUNTER14[9]	COUNTER14[8]	COUNTER14[7]	COUNTER14[6]	COUNTER14[5]	COUNTER14[4]	COUNTER14[3]	COUNTER14[2]	COUNTER14[1]	COUNTER14[0]
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.60 - Register LLI_COUNTER14

Name	Type	Size	Description
COUNTER14[15:0]	U	16	Value of the LLI counter #n

Table 2.61 - Register LLI_COUNTER14 fields description

2.1.2.31 Register LLI_COUNTER15

The DMA provides 16 LLI counters. The LLI counter n each time a fragment having its descriptor field LLI counter set to n is transferred. A LLI counter wraps to zero after crossing the 16'hFFFF boundary value and can be set to any value by the software.

Address	Access		LLI_COUNTER15																													
	HW	SW	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
+BC'H	W	R																	COUNTER15[15]	COUNTER15[14]	COUNTER15[13]	COUNTER15[12]	COUNTER15[11]	COUNTER15[10]	COUNTER15[9]	COUNTER15[8]	COUNTER15[7]	COUNTER15[6]	COUNTER15[5]	COUNTER15[4]	COUNTER15[3]	COUNTER15[2]	COUNTER15[1]	COUNTER15[0]

Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type																	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access																	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access																	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.62 - Register LLI_COUNTER15

Name	Type	Size	Description
COUNTER15[15:0]	U	16	Value of the LLI counter #n

Table 2.63 - Register LLI_COUNTER15 fields description

2.2 LLI node structure

The characteristics of the data fragment to be transferred are described by a LLI node located in the shared memory.

The structure address location shall be 32 bits word aligned.

Address offset	Range	Name field	Description
0	31-0	SADDR	Source address of the data fragment. It is a host address if the node belongs to a list attached to the downstream channel or an embedded address is the node belongs to a list attached to an upstream channel.
4	31-0	DADDR	Destination address of the data fragment. It is an embedded address if the node belongs to a list attached to the downstream channel or a host address is the node belongs to a list attached to an upstream channel.
8	31-29	-	Reserved
	28	IRQ_EN	If the bit IRQ_EN is set, the LLI interrupt corresponding to field IRQ_NO is asserted once the fragment processed.
	27-24	IRQ_NO	
	23-21	-	Reserved
	20	COUNTER_EN	If the bit COUNTER_EN is set, the DMA counter pointed by the COUNTER_NO field is incremented once the fragment processed.
	19-16	COUNTER_NO	
	15-0	LENGTH	The byte length of the fragment to be transferred. Valid range values starts from 1 byte to 65535 bytes.

Table 2.64 - Link list item structure

3 Hardware Interface

This chapter describes the port map of rw_platform_dma module.

Name	Type	Size	Description
Resets			
prst_n	input	1	Active low hard reset signal synchronized to the clk.
Clocks			
clk	input	1	Platform Clock
Interrupt sources			
lli_irq	output	16	LLI Interrupt from host
channel_irq	output	4	Channel Interrupts to processor
error_irq	output	1	Error Interrupt to processor
Interface to SharedRAM MultiPort Arbiter			
Upstream interface			
dma0_ready	Input	1	Upstream Bus Ready
dma0_addr	output	32	Upstream Address
dma0_trans	output	1	Upstream Trans Enable
dma0_rdata	input	64	Upstream Read Data
Downstream interface			
dma1_ready	input	1	Downstream Bus Ready
dma1_addr	output	32	Downstream Address
dma1_trans	output	1	Downstream Trans Enable
dma1_wdata	output	64	Downstream Write Data
dma1_we	output	8	Downstream Write Bytes Enable
Midstream interface			
dma2_addr	output	32	Midstream Address
dma2_trans	output	1	Midstream Trans Enable
dma2_write	output	1	Midstream Write Enable
dma2_rdata	input	64	Midstream Read Data
dma2_wdata	output	64	Midstream Write Data
dma2_we	output	8	Midstream Write Bytes Enable
dma2_ready	input	1	Midstream Bus Ready
Processor AHB interface			
hready_regb	output	1	AHB Slave hready
haddr_regb	input	8	AHB Slave haddr
htrans_regb	input	2	AHB Slave htrans
hready_in_regb	input	1	AHB Slave hready_in
hwrite_regb	input	1	AHB Slave hwrite
hsize_regb	input	2	AHB Slave hsize
hrdata_regb	output	32	AHB Slave hrdata
hwdata_regb	input	32	AHB Slave hwdata
hresp_regb	output	2	AHB Slave hresp
AXI interfaces			
Upstream interface			
dma0_awid	output	4	Write address ID
dma0_awaddr	output	32	Write address
dma0_awlen	output	8	Write burst length
dma0_awsize	output	3	Write burst size
dma0_awburst	output	2	Write burst type
dma0_awuser	output	11	Write burst length (in bytes)
dma0_awvalid	output	1	Write address valid
dma0_awready	input	1	Write address ready
dma0_wid	output	4	Write ID tag
dma0_wdata	output	64	Write data
dma0_wstrb	output	8	Write strobes

dma0_wlast	output	1	Write last
dma0_wvalid	output	1	Write valid
dma0_wready	input	1	Write ready
dma0_bid	input	4	Write response ID tag
dma0_bresp	input	2	Write response
dma0_bvalid	input	1	Write response valid
dma0_bready	output	1	Write response ready
Downstream interface			
dma1_arid	output	4	Read address ID
dma1_araddr	output	32	Read address
dma1_arlen	output	8	Read burst length
dma1_arsize	output	3	Read burst size
dma1_arburst	output	2	Read burst type
dma1_arvalid	output	1	Read address valid
dma1_arready	input	1	Read address ready
dma1_rid	input	4	Read ID tag
dma1_rdata	input	64	Read data
dma1_rresp	input	2	Read response
dma1_rlast	input	1	Read last
dma1_rvalid	input	1	Read valid
dma1_rready	output	1	Read ready
AHB Master Interface (LLI)			
hresp_lli	input	2	LLI hresp
haddr_lli	output	32	LLI haddr
htrans_lli	output	2	LLI htrans
hwrite_lli	output	1	LLI hwrite
hrdata_lli	input	32	LLI hrdata
hready_lli	input	1	LLI hready
AHB master (AHB memory map)			
dma_hready	input	1	AHB memory map ready
dma_haddr	output	32	AHB memory map addr
dma_htrans	output	2	AHB memory map htrans
dma_hwrite	output	1	AHB memory map write
dma_hsize	output	2	AHB memory map size
dma_hwdata	output	32	AHB memory map write data
dma_hrdata	input	32	AHB memory map read data
dma_hresp	input	2	AHB memory map response

Table 3.1 - rw_platform_dma module port map

3.1 AXI integration guide

The DMA interface can be connected to an AXI3 or AXI4 slave interface. The Table 3.2 shows the upstream connection to an AXI3 or AXI4 slave interface. The Table 3.3 shows the downstream connection to an AXI3 or AXI4 slave interface.

dma0_awuser signal is reserved and should not be connected.

dma0_wid signal should be connected to an AXI3 slave only.

dma0_awlen[7:4] & dma1_arlen[7:4] should be connected to an AXI4 slave only.

When connected to an AXI3 slave, the **Error! Reference source not found.** register should be configured with a burst length lower or equal to 16 transfers.

Upstream AXI Interface	AXI3 Interface	AXI4 Interface
dma0_awid[3:0]	axi3_awid[3:0]	axi4_awid[3:0]
dma0_awaddr[31:0]	axi3_awaddr[31:0]	axi4_awaddr[31:0]
dma0_awlen[3:0]	axi3_awlen[3:0]	axi4_awlen[3:0]

dma0_awlen[7:4]	-	axi4_awlen[7:4]
dma0_awsize[2:0]	axi3_awsize[2:0]	axi4_awsize[2:0]
dma0_awburst[1:0]	axi3_awburst[1:0]	axi4_awburst[1:0]
dma0_awuser[10:0]	-	-
dma0_awvalid	axi3_awvalid	axi4_awvalid
dma0_awready	axi3_awready	axi4_awready
dma0_wid[3:0]	axi3_wid[3:0]	-
dma0_wdata[63:0]	axi3_wdata[63:0]	axi4_wdata[63:0]
dma0_wstrb[7:0]	axi3_wstrb[7:0]	axi4_wstrb[7:0]
dma0_wlast	axi3_wlast	axi4_wlast
dma0_wvalid	axi3_wvalid	axi4_wvalid
dma0_wready	axi3_wready	axi4_wready
dma0_bid[3:0]	axi3_bid[3:0]	axi4_bid[3:0]
dma0_bresp[1:0]	axi3_bresp[1:0]	axi4_bresp[1:0]
dma0_bvalid	axi3_bvalid	axi4_bvalid
dma0_bready	axi3_bready	axi4_bready

Table 3.2 - upstream AXI integration

Downstream AXI Interface	**AXI3 Interface**	**AXI4 Interface**
dma1_arid[3:0]	axi3_arid[3:0]	axi4_arid[3:0]
dma1_araddr[31:0]	axi3_araddr[31:0]	axi4_araddr[31:0]
dma1_arlen[3:0]	axi3_arlen[3:0]	axi4_arlen[3:0]
dma1_arlen[7:4]	-	axi4_arlen[7:4]
dma1_arsize[2:0]	axi3_arsize[2:0]	axi4_arsize[2:0]
dma1_arburst[1:0]	axi3_arburst[1:0]	axi4_arburst[1:0]
dma1_arvalid	axi3_arvalid	axi4_arvalid
dma1_arready	axi3_arready	axi4_arready
dma1_rid[3:0]	axi3_rid[3:0]	axi4_rid[3:0]
dma1_rdata[63:0]	axi3_rdata[63:0]	axi4_rdata[63:0]
dma1_rresp[1:0]	axi3_rresp[1:0]	axi4_rresp[1:0]
dma1_rlast	axi3_rlast	axi4_rlast
dma1_rvalid	axi3_rvalid	axi4_rvalid
dma1_rready	axi3_rready	axi4_rready

Table 3.3 - downstream AXI integration

4 DMA Host Interface

4.1 Overview

The DMA Host Interface is an AXI master and was designed to reach very high data rate DMA transfers (8Gb/s) between the user FPGAs and the Host Memory (DDR for example).

The interface is **transaction oriented** and hides low-level protocols such PCI-E to the user FPGA design.

A transaction is **always initiated by the user FPGA design** and can have **two directions**. For clarity, we call **upstream** the data transfer from the reference platform to the Host memory, and **downstream** the data transfer from the Host memory to the reference platform.

These **suffixes** simply indicate the **command/data signals direction** and are **not related to transaction direction**.

A transaction is decomposed into three parts:

1. The **request descriptor**: The request part is a short structure (2 words of 64 bits) sent to the host, it describes the requested DMA transfer and contains information such transfer size, host address, tag ID, transfer direction. Because a request is only initiated by the reference platform, the request only moves from the reference platform to the host through the xxx_tohost signals.
2. The **data**: The data part is the data buffer to be transferred, it is sent per 64 bits words. Depending of the transaction direction given by the request descriptor, the data moves through xxx_tohost for upstream transaction, or through xxx_fromhost for downstream transaction.
3. The **acknowledge descriptor**: Depending of the transaction direction, the acknowledge descriptor is always returned by the host. For upstream transactions, a acknowledge descriptor is returned once the buffer is fully written to the host memory. Because many outstanding upstream/downstream transactions can occur, the acknowledge descriptor returns the tag ID which identifies the corresponding request and buffer. For downstream transaction, the acknowledge descriptor precedes the returned data.

The diagram below shows an example of transaction exchange between the host and the reference platform.

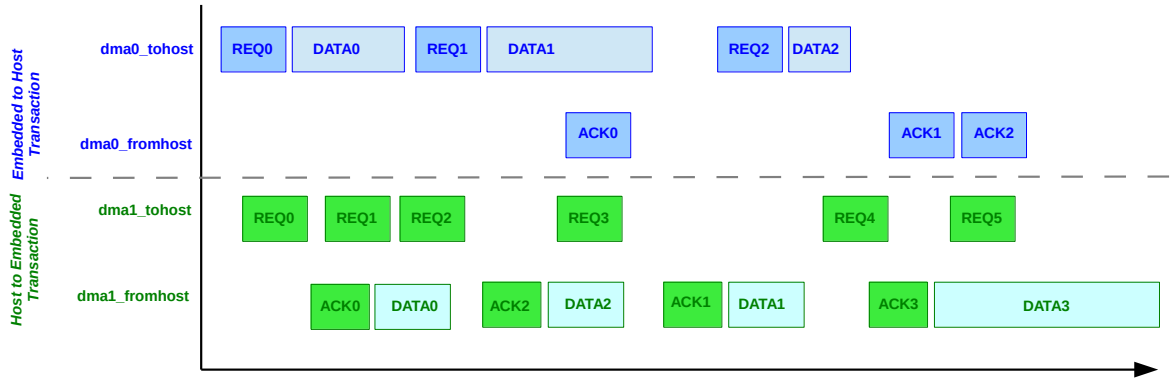

Figure 2 - Example of upstream and downstream transactions

4.2 Full duplex operations

To permit full duplex operations, the RW DMA uses the **dma0 interface for upstream** transaction and **dma1 interface for downstream** transaction.

For clarity, the diagram below shows how is integrated the DMA with the reference platform, and also how the DMA is interconnected with the Host infrastructure:

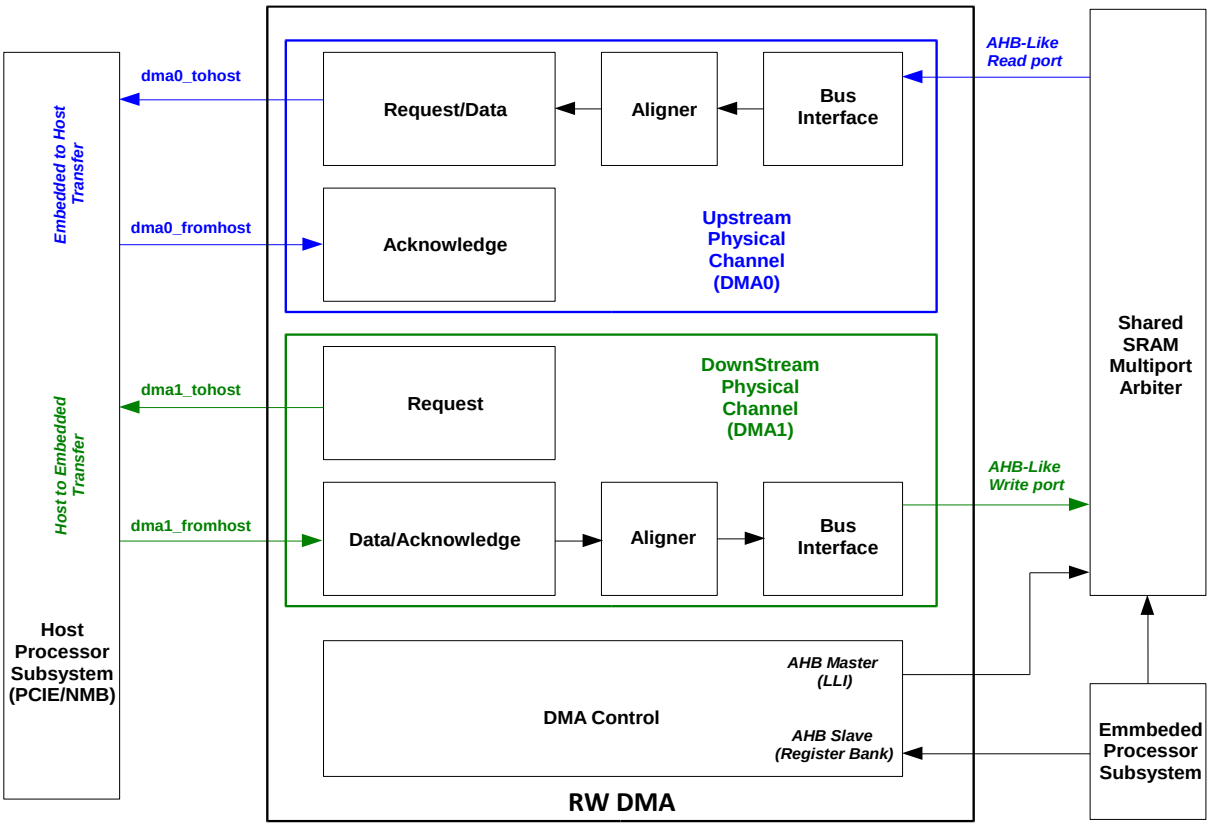

Figure 3 - Host and Reference platform DMA interconnection