CoreHPDMACtrl v2.1

Handbook

Microsemi

Table of Contents

Introduction

General Description

The CoreHPDMACtrl soft IP is used to control the high performance direct memory access (HPDMA) of the microcontroller subsystem (MSS) in SmartFusion®2 or high-performance memory subsystem (HPMS) in IGLOO®2 and also monitors the transaction status.

CoreHPDMACtrl selects the buffer descriptor, initiates the transactions on the fabric interface controller (FIC) and configures the MSS or HPMS HPDMA. It also monitors the interrupts for transfer done and transfer error. The soft IP also re-configures the registers of HPDMA to prepare it for the next transfer.

Key Features

CoreHPDMACtrl has the following features:

- Provides four HPDMA buffer descriptors.
- Starts and resets the memory descriptor.
- Provides source and destination address for each descriptor.
- Provides size and direction of transfer for each descriptor.
- Provides DMA transfer complete and error interrupts.
- Provides advanced high-performance bus (AHB)-Lite master interface to the FIC on the MSS or HPMS.
- Supports word-aligned data transfers.

Core Version

This Handbook applies to CoreHPDMACtrl version 2.1.

Supported Families

- SmartFusion®2
- IGLOO®2

Utilization and Performance

Utilization and performance data for the SmartFusion2 and IGLOO2 device family is listed in Table 1. The data listed in this table is indicative only. The overall device utilization and performance of the core is system dependent.

Table 1 Device Utilization and Performance

Family	Device	Package	Logic Elements				Frequency (MHz)
			Sequential	Combinatorial	Total	%	
SmartFusion2	M2S150TS	1152FC	517	481	998	0.006	170
IGLOO2	M2GL150TS	1152FC	517	481	998	0.006	170

Note: The data in this table was achieved using typical synthesis and layout settings. Frequency (in MHz) was set to 150 and speed grade was -1. The performance is listed only for descriptor buffer 0. It is same for descriptor buffer 1, 2, and 3.

Functional Block Description

CoreHPDMACtrl soft IP is used to configure the HPDMA of the MSS or HPMS. This soft IP provides a user interface to initiate a transaction to each of the four available descriptors. The core has an AHB-Lite (AHBL) master interface and is interfaced with the fabric interface controller (FIC) AHBL slave. The core communicates with the MSS or HPMS HPDMA through the FIC interface.

The core initializes and selects the HPDMA buffer descriptor. It then initiates the transactions and also monitors the interrupts for transfer complete or transfer error. The core starts the transfer again for the next descriptor through an input port (HPD_VALID).

CoreHPDMACtrl soft IP consists of the following sub-blocks:

- User Interface Block
- Command Decoder Block
- FSM Control Block
- AHBL Master Interface Block

Figure 1 shows the functional block diagram of CoreHPDMACtrl.

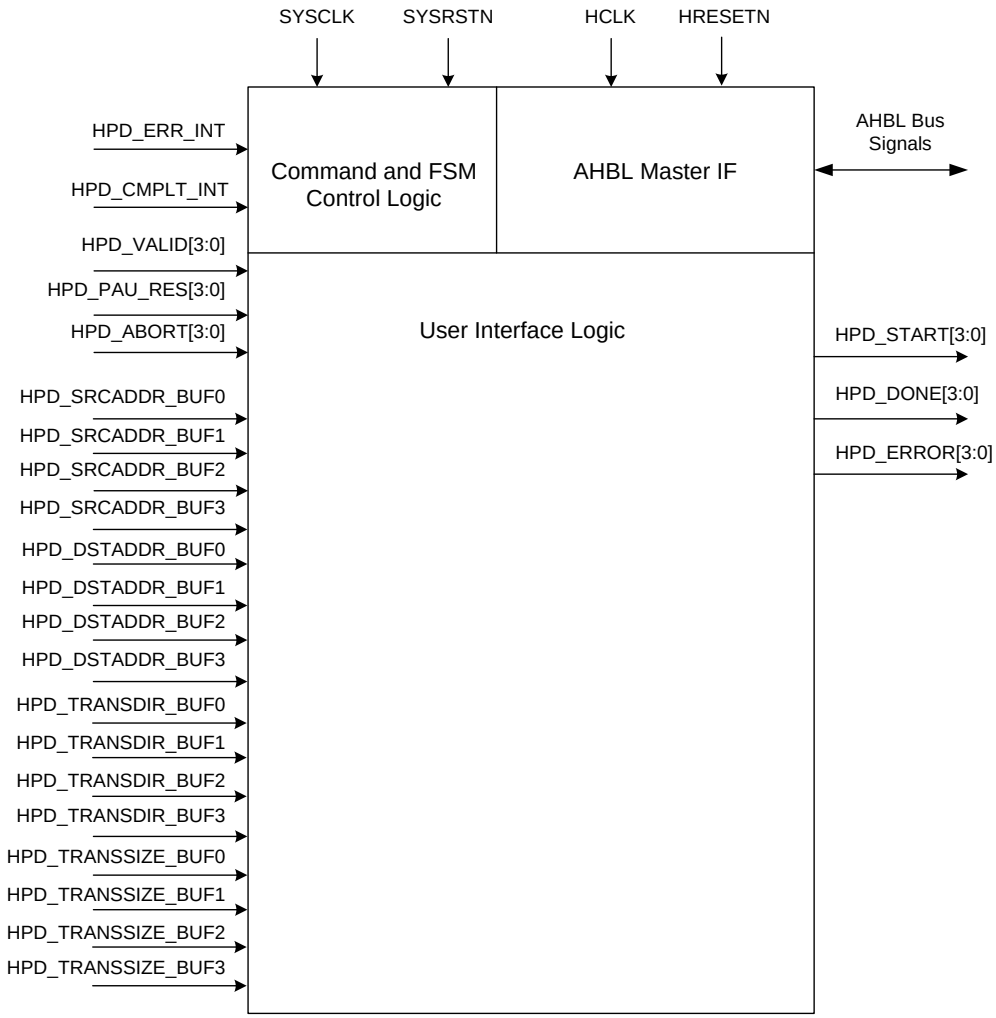

Figure 1 CoreHPDMACtrl Functional Block Diagram

Figure 2 shows the configuration and data flow diagram for HPDMA. This CoreHPDMACtrl soft IP flow diagram shows the following transactions of CoreHPDMACtrl IP:

1. Configure the fabric interface interrupt controller (FIIC) registers to enable interrupts from the MSS or HPMS to fabric.
2. Configure HPDMA Interrupt Clear register.
3. Configure the following HPDMA registers:
   - Source Address register
   - Destination Address register
   - Control register
     - Configure transfer size and transfer direction
     - Enable the transfer complete and transfer error interrupts
4. As per the configuration set in the HPDMA registers, the DMA transfers are initiated between the double data rate (DDR) memory and the AHB bus matrix mapped memory regions (eSRAM/eNVM/Fabric RAMs) as shown in Figure 2.

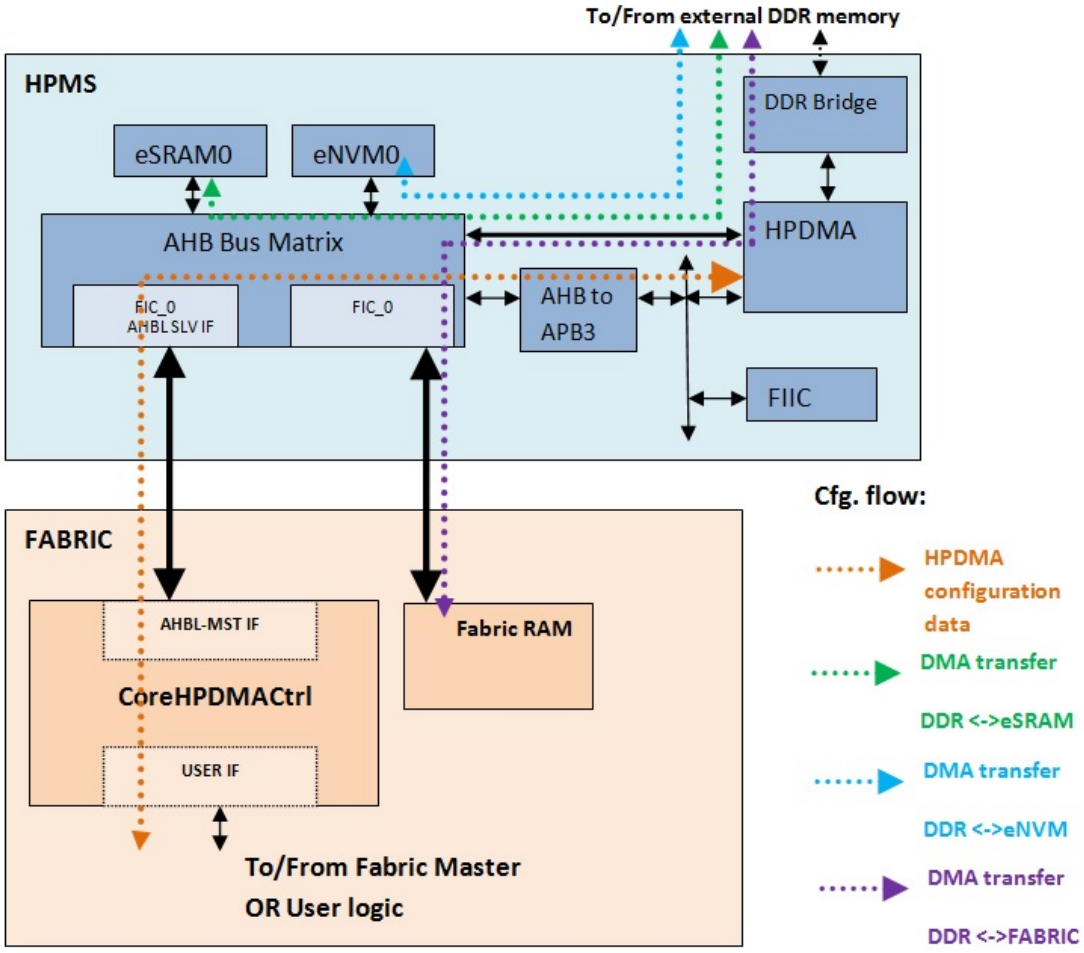

Figure 2 CoreHPDMACtrl Configuration and Data Flow Diagram

User Interface Block

The user interface block is the front-end to the user logic.

Each individual bit in the input HPD_VALID[3:0] represents one of the four available descriptors. When set, DMA transactions are initiated between the corresponding source and destination end. CoreHPDMACtrl soft IP also initiates the corresponding descriptor's source address, destination address, transfer size and transfer direction along with the HPD_VALID signal. Once the descriptor valid bit is set, descriptor configurations such as source address, destination address, transfer size, and transfer direction are not recommended to be changed.

For example, when the HPD_VALID[3:0] is set to 4'b0001, the valid bit is set in the descriptor 0. It invokes the DMA transactions to or from DDR bridge and AHB bus matrix memory depending on the configured transfer direction, source address, and destination address. HPD_START[3:0] is asserted to 4'b0001 indicating that descriptor 0 transfer is currently active.

Note:

1. The next transfer must be initiated only when the previous transfer is completed (with corresponding descriptor bit set in HPD_DONE[3:0] or HPD_ERROR[3:0] for a clock pulse) and the HPD_START[3:0] output is low. Refer to Figure 2 on page 8.
2. When HPD_ABORT[3:0] is asserted for the currently active descriptor, it is recommended to maintain it until the HPD_START[3:0] output is low.

Command Decoder Block

The command decoder block latches the HPDMA valid bit configuration received from the user interface block. It decodes and determines the descriptor requested out of the four descriptors.

It consists of a main state machine to enable the configuration of the FIIC registers. It is also responsible for the configuration of HPDMA Control register, Interrupt Clear register, Source Address register, and Destination Address register for the selected descriptor. It also provides inputs to the finite state machine (FSM) control block to drive the write/read enable, burst length, and target register address.

The command decoder block also handles the interrupts (HPD_ERR_INT and HPD_CMPLT_INT) coming from the MSS in SmartFusion2 or the HPMS in IGLOO2. Then it triggers the next transfer by invoking the HPD_START[3:0] input.

FSM Control Block

The FSM control block is the interface between the command decoder interface on one side and the AHBL master interface on the other. It contains a state machine to drive the AHBL master transactions on the AHB-Lite bus.

AHBL Master Interface Block

The AHBL master interface block is the standard advanced microcontroller bus architecture (AMBA®) to the FIC on the MSS or HPMS. CoreHPDMACtrl soft IP communicates with the MSS or HPMS HPDMA through the AHBL master FIC interface. Each descriptor configuration data is configured through the user interface and is translated into the corresponding AHBL master transaction. All the transactions are addressed to the MSS or HPMS HPDMA.

Timing Diagrams

Figure 3 shows the timings for generating the valid (HPD_VALID) corresponding to descriptor 0. It also shows the transfer done (HPD_DONE) generation. The timings shown for HPD_VALID are same for all the descriptors (descriptor 0 to descriptor 3).

Figure 3 shows the HPD_VALID for descriptor 0 generated for single clock cycle. The corresponding HPD_START for the descriptor 0 is asserted on the clock after the HPD_VALID input is initiated indicating that the transfer is active for the selected descriptor. It remains asserted till the current transfer is completed. On receiving HPD_DONE for the current transaction, HPD_START bit for the corresponding descriptor is de-asserted indicating that it is no longer active and is ready for another transfer.

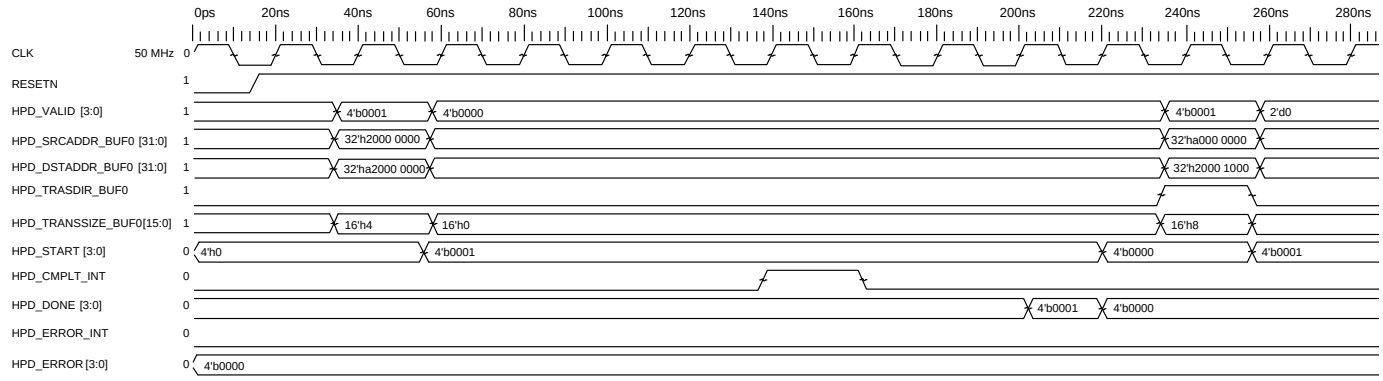

Figure 3 CoreHPDMACtrl Timings with Transfer Done

Figure 4 shows the timings for generating the valid (HPD_VALID) corresponding to descriptor 0. It also represents the error (HPD_ERROR) generation. The timings shown for HPD_VALID are same for all the descriptors (descriptor 0 to descriptor 3).

Figure 4 shows the HPD_VALID for descriptor 0 generated for single clock cycle. The corresponding HPD_START for the descriptor 0 is asserted on the clock after the HPD_VALID input is initiated indicating that the transfer is active for the selected descriptor. It remains asserted till the current transfer is complete. On receiving HPD_ERROR for the current transaction, HPD_START bit for the corresponding descriptor is de-asserted indicating that it is no longer active and is ready for another transfer.

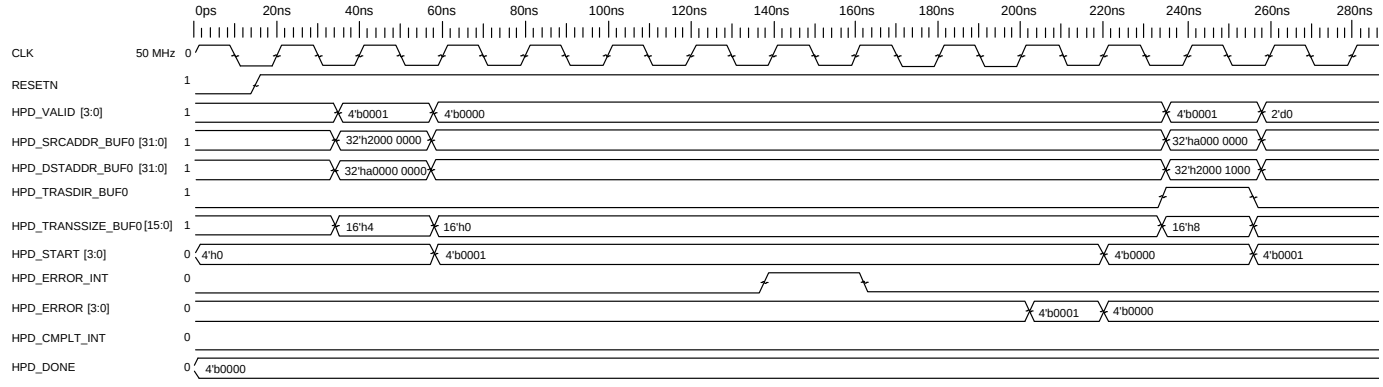

Figure 4 CoreHPDMACtrl Timings with Transfer Error Operation

Figure 5 shows the timings for generating the valid (HPD_VALID) corresponding to descriptor 0. It also shows the timings for abort (HPD_ABORT) generation. The timings shown for HPD_VALID are same for all the descriptors (descriptor 0 to descriptor 3).

Figure 5 shows the HPD_VALID for descriptor 0 generated along with HPD_ABORT. The HPD_VALID input is asserted only for single clock cycle. The corresponding HPD_START for the descriptor 0 is asserted on the clock after the HPD_VALID input is initiated indicating that the transfer is active for the selected descriptor. It remains asserted till the current transfer is complete. The abort (HPD_ABORT) when asserted terminates the current transaction. The HPD_ABORT for the currently-active-descriptor must be de-asserted by the user design logic when the HPD_START bit for the descriptor 0 goes low as shown in Figure 5. The user design logic is ready for another transfer when the HPD_START is low.

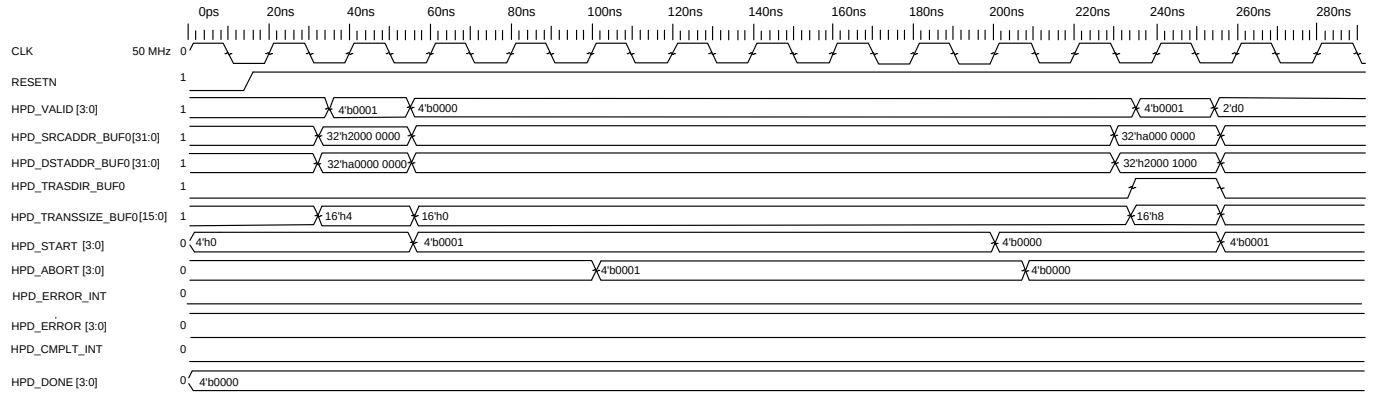

Figure 5 HPD_VALID for Descriptor 0 Generated along with HPD_ABORT Operation

Figure 6 shows the timings for generating the valid (HPD_VALID) corresponding to descriptor 0. It also shows the timings for pause/resume (HPD_PAU_RES) operation. The timings shown for HPD_VALID are same for all the descriptors (descriptor 0 to descriptor 3).

Figure 6 shows the HPD_VALID for descriptor 0 generated along with HPD_PAU_RES. The HPD_VALID input is asserted only for single clock cycle. The corresponding HPD_START for the descriptor 0 is asserted on the clock after the HPD_VALID input is initiated indicating that the transfer is active for the selected descriptor. It remains asserted till the current transfer is complete. The pause (HPD_PAU_RES) when asserted pauses/suspends the current transaction. The current transaction will resume when HPD_PAU_RES is de-asserted by the user design logic as shown in Figure 6. The user design logic is ready for another transfer when the HPD_START is low.

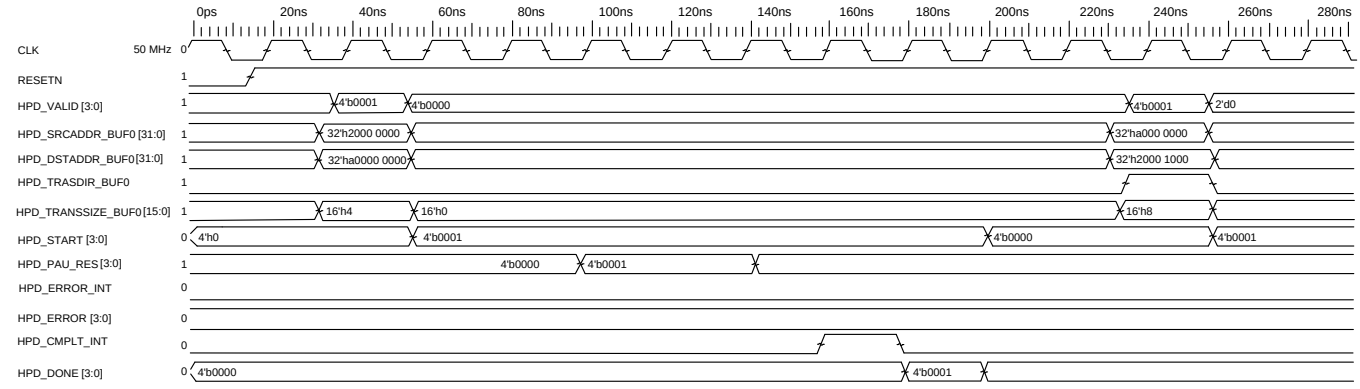

Figure 6 HPD_VALID for Descriptor 0 Generated along with HPD_ PAU_RES Operation

Licensing

CoreHPDMACtrl is licensed in one way: Register transfer level (RTL).

RTL

Complete RTL source code is provided for the core.

SmartDesign

CoreHPDMACtrl is preinstalled in the SmartDesign IP Deployment design environment. An example instantiated view is shown in Figure 7.

The core can be configured using the configuration GUI within SmartDesign. Figure 8 on page 14 shows the **SmartDesign CoreHPDMACtrl Configuration** window.

For more information on using SmartDesign to instantiate and generate cores, refer to the *Using DirectCore in Libero® System-on-Chip (SoC) User Guide* or consult the Libero SoC online help.

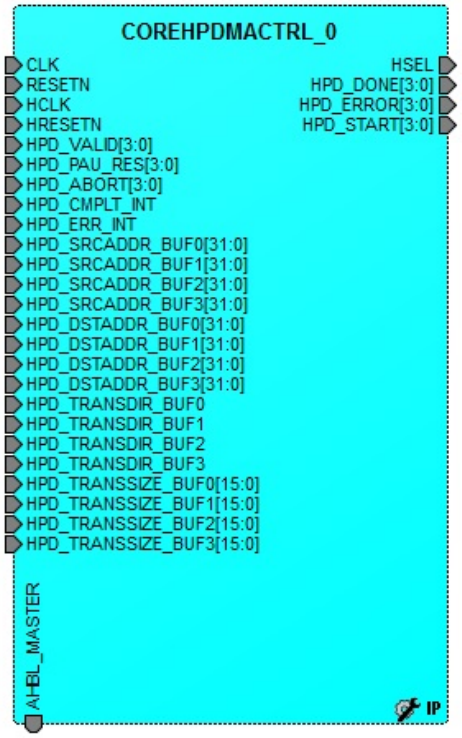

Figure 7 SmartDesign CoreHPDMACtrl Instance View

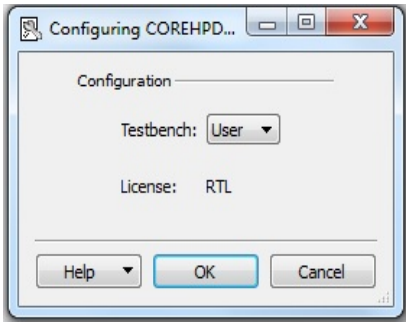

Figure 8 SmartDesign CoreHPDMACtrl Configuration Window

Simulation Flows

The User Testbench for CoreHPDMACtrl is included in all releases.

To run simulations, select the **User Testbench** flow within SmartDesign and click **Save** and **Generate** on the **Generate** pane. The **User Testbench** is selected through the core Testbench Configuration GUI.

When SmartDesign generates the Libero SoC project, it installs the user testbench files.

To run the **User Testbench**, set the design root to CoreHPDMACtrl instantiation in the Libero SoC design hierarchy pane and click **Simulation** in the Libero SoC design flow window. This invokes ModelSim® and automatically runs the simulation.

Synthesis in Libero SoC

Clicking **Synthesis** in Libero SoC software, the **Synthesis** window displays the Synplicity® project. Synplicity should be set for using the Verilog 2001 standard if Verilog is being used. Select **Run** to run Synthesis.

Place-and-Route in Libero SoC

To invoke designer, **Layout** should be clicked in the Libero SoC software. CoreHPDMACtrl does not require any special place-and-route settings.

Core Interfaces

I/O Signals

I/O signal descriptions for CoreHPDMACtrl are listed in Table 2.

Table 2 CoreHPDMACtrl I/O Signals

Portname	Type	Description
Clocks and Resets		
CLK	In	System Clock.
RESETN	In	System reset. Active low asynchronous reset.
HCLK	In	AHB Clock.
HRESET	In	AHB reset. Active low asynchronous reset.
User Interface Signal		
HPD_VALID[3:0]	In	Active-high input which selects which of the four buffer descriptors are valid.
HPD_PAU_RES[3:0]	In	Active-high input which selects which of the four buffer descriptors are to be paused. Active-low input which selects which of the four buffer descriptors are to be resumed from where they have stopped.
HPD_ABORT[3:0]	In	Active-high input which selects which of the four buffer descriptors are to be cleared. This clears the corresponding descriptor fields. The HPDMA then terminates the current transfer and resets the descriptor status and control registers.
HPD_START[3:0]	Out	Active-high indicates which buffer descriptor is running.
HPD_DONE [3:0]	Out	Active-high pulse indicates when a buffer descriptor transfer is complete.
HPD_ERROR [3:0]	Out	Active-high pulse with done indicates if the transfer completed in error.
HPD_SRCADDR_BUF0	In	Source end memory address register for buffer descriptor 0.
HPD_DSTADDR_BUF0	In	Destination end memory address register for buffer descriptor 0.
HPD_TRANSSIZE_BUF0	In	Transfer size in bytes for buffer descriptor 0. Defines number of bytes to be transferred in a descriptor 0 transfer.
HPD_TRANSDIR_BUF0	In	Buffer descriptor 0 transfer direction. 0: AHB bus matrix to MSS DDR bridge 1: MSS DDR bridge to AHB bus matrix
HPD_SRCADDR_BUF1	In	Source end memory address register for buffer descriptor 1.
HPD_DSTADDR_BUF1	In	Destination end memory address register for buffer descriptor 1.
HPD_TRANSSIZE_BUF1	In	Transfer size in bytes for buffer descriptor 1. Defines number of bytes to be transferred in a descriptor 1 transfer.
HPD_TRANSDIR_BUF1	In	Buffer descriptor 1 transfer direction. 0: AHB bus matrix to MSS DDR bridge 1: MSS DDR bridge to AHB bus matrix

Portname	Type	Description
HPD_SRCADDR_BUF2	In	Source end memory address register for buffer descriptor 2
HPD_DSTADDR_BUF2	In	Destination end memory address register for buffer descriptor 2
HPD_TRANSSIZE_BUF2	In	Transfer size in bytes for buffer descriptor 2. Defines number of bytes to be transferred in a descriptor 2 transfer.
HPD_TRANSDIR_BUF2	In	Buffer descriptor 2 transfer direction. 0: AHB bus matrix to MSS DDR bridge 1: MSS DDR bridge to AHB bus matrix
HPD_SRCADDR_BUF3	In	Source end memory address register for buffer descriptor 3
HPD_DSTADDR_BUF3	In	Destination end memory address register for buffer descriptor 3
HPD_TRANSSIZE_BUF3	In	Transfer size in bytes for buffer descriptor 3. Defines number of bytes to be transferred in a descriptor 3 transfer.
HPD_TRANSDIR_BUF3	In	Buffer descriptor 3 transfer direction. 0: AHB bus matrix to MSS DDR bridge 1: MSS DDR bridge to AHB bus matrix
Interrupts		
HPD_CMPLT_INT	In	It is asserted when HPDMA completes at least one Descriptor transfer. Once asserted, it remains asserted until cleared by means of writing 1 to the HPDMAICR_CLR_XFR_INT bit Interrupt Clear register of the Descriptor- 0, 1, 2, and 3. If HPDMA completes more than one Descriptor transfers before the interrupt is serviced, then this bit remains asserted until all the descriptors have had HPDMAICR_CLR_XFR_INT written to one.
HPD_ERROR_INT	In	It is asserted when any error occurs during any of the four Descriptors transfer. The logical OR of individual Descriptor transfer error status is asserted as transfer error interrupt. Once asserted, it remains asserted until cleared by means of writing 1 HPDMAICR_CLR_XFR_INT bit Interrupt Clear Register of the Descriptor- 0, 1, 2, and 3. If HPDMA completes more than one Descriptor with errors before the interrupt is serviced, then this bit remains asserted until all the descriptors have had HPDMAICR_CLR_XFR_INTwritten to one.
AHB-Lite Master Interface Signals		
HSEL	Out	AHBL slave select – this signal indicates that the current transfer is intended for the selected slave.
HADDR[31:0]	Out	AHBL address – 32-bit address on the AHBL interface.
HWRITE	Out	AHBL write – When HIGH,this signal indicates that the current transaction is a write. When low, this signal indicates that the current transaction is a read.
HTRANS[1:0]	Out	AHBL transfer type – Indicates the transfer type of the current transaction. b00: IDLE b01: BUSY b10: NONSEQUENTIAL b11: SEQUENTIAL

Portname	Type	Description
HSIZE[1:0]	Out	AHBL transfer size – Indicates the size of the current transfer (8/16/32/64 bit transactions only). bx00: 8-bit (byte) transaction bx01: 16-bit (half word) transaction bx10: 32-bit (word) transaction bx11: 64-bit(double word) transaction
HBURST[2:0]	Out	AHBL Burst type.
HWDATA[31:0]	Out	AHBL write data – Writes data from the AHBL master to the AHBL slave
HREADY	In	When HIGH, this signal indicates to the master that the previous transfer is complete.
HRESP	In	AHBL response status – When driven HIGH at the end of a transaction, this signal indicates that the transaction has completed with errors. When driven low at the end of a transaction, this signal indicates that the transaction has completed successfully.
HRDATA[31:0]	In	AHBL read data – Reads data from the AHBL slave to the AHBL master.

Core Parameters

There are no parameters for CoreHPDMACtrl.

Register Map and Descriptions

CoreHPDMACtrl does not contain any registers.

References

- SmartFusion2 System Controller User Guide
- ARM Cortex-M3 Processor and Subsystem in SmartFusion2 Devices User Guide
- IGLOO2 System Controller User Guide
- IGLOO2 High Performance Memory Subsystem User Guide

Ordering Information

Ordering Codes

CoreHPDMACtrl can be ordered through the local sales representative. It should be ordered using the following number scheme: CoreHPDMACtrl -XX, where XX is listed in Table 3.

Table 3 Ordering Codes

XX	Description
RM	RTL for RTL source—multiple use license.

List of Changes

The following table lists critical changes that were made in each revision of the document.

Table 4 List of changes

Date	Change	Page
March 2014	CoreHPDMACtrl v2.1 release.	N/A
	Added the new ports for source address, destination address, transfer size, and transfer direction as mentioned in Table 2.	15
	Updated Core Parameters section.	17
November 2013	CoreHPDMACtrl v2.0 release.	N/A

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world **650. 318.8044**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1(949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and security, aerospace, and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices, and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

50200500 -1/04.14