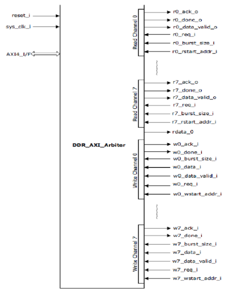
Manuals+ — User Manuals Simplified.

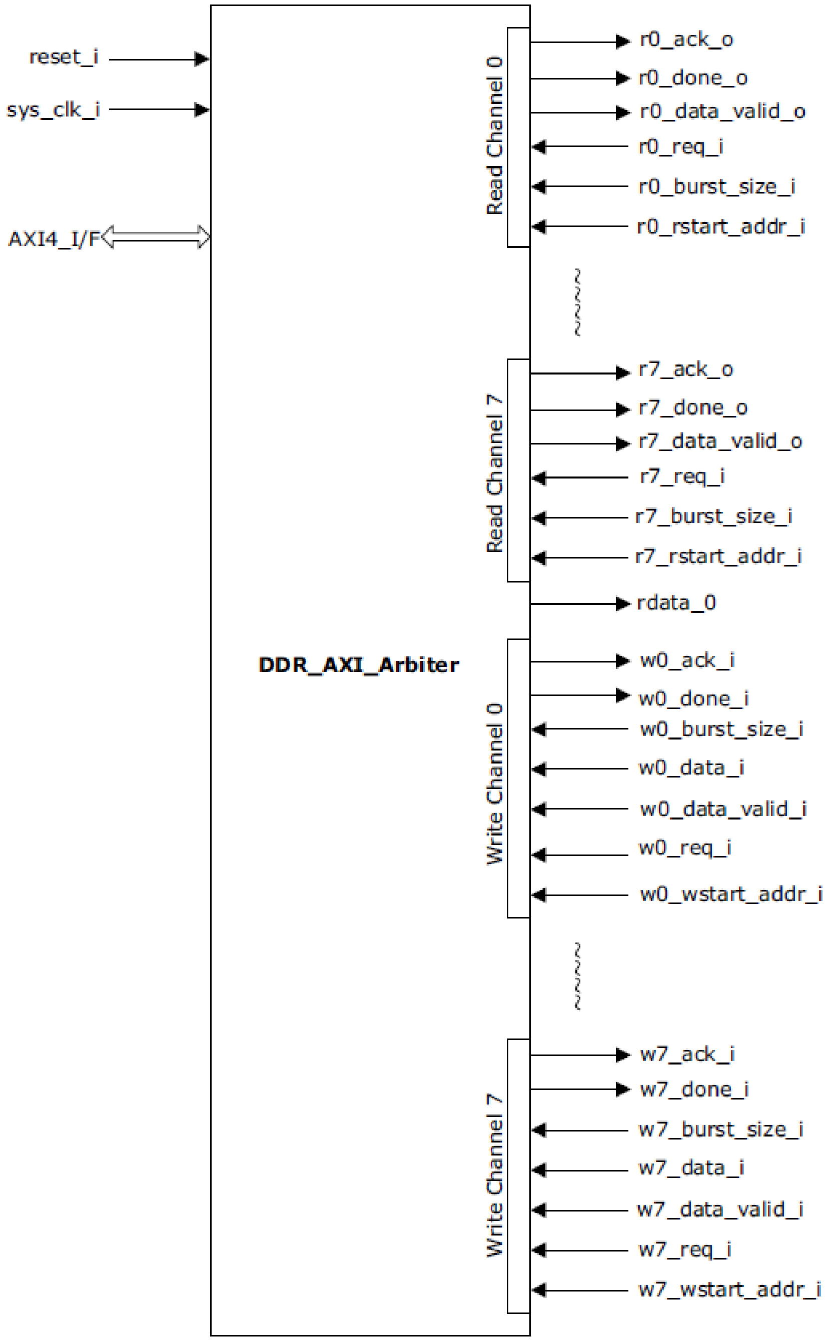


Microsemi UG0950 DDR AXI4 Arbiter IP User Guide

Home » Microsemi » Microsemi UG0950 DDR AXI4 Arbiter IP User Guide

Contents

Microsemi UG0950 DDR AXI4 Arbiter IP



Product Information

The Microsemi DDR_AXI4_Arbiter is a hardware implementation device that is commonly used in video and graphics applications. It is designed to support Double Data Rate (DDR) Synchronous Dynamic Random Access Memory (SDRAM) for fast processing in video systems.
The device is equipped with key features such as design description, inputs and outputs, configuration parameters, and timing diagram for efficient functionality.

Key Features

- Supports DDR SDRAM
- Efficient design description
- Multiple inputs and outputs
- Configurable parameters for customization
- Timing diagram for accurate performance evaluation

Supported Families

The DDR_AXI4_Arbiter is designed to support a wide range of families for video and graphics applications.

Product Usage Instructions

To use the Microsemi DDR_AXI4_Arbiter device, follow the installation instructions provided in the user manual. The device should be installed by a qualified technician to ensure proper functionality. Once installed, the device can be configured using the configuration parameters provided in the user manual. The timing diagram should be used to evaluate the performance of the device. In case of any issues or queries regarding the device, contact Microsemi sales support through the provided contact information.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center 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; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at **www.microsemi.com**.

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113 Outside the USA: +1 (949) 380-6100 Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: **sales.support@microsemi.com**
www.microsemi.com

Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision 1.0

The first publication of this document.

Introduction

Memories are an integral part of any typical video and graphics application. They are used for buffering video pixel data. One common buffering example displays frame buffers in which the complete video pixel data for a frame is buffered in the memory.

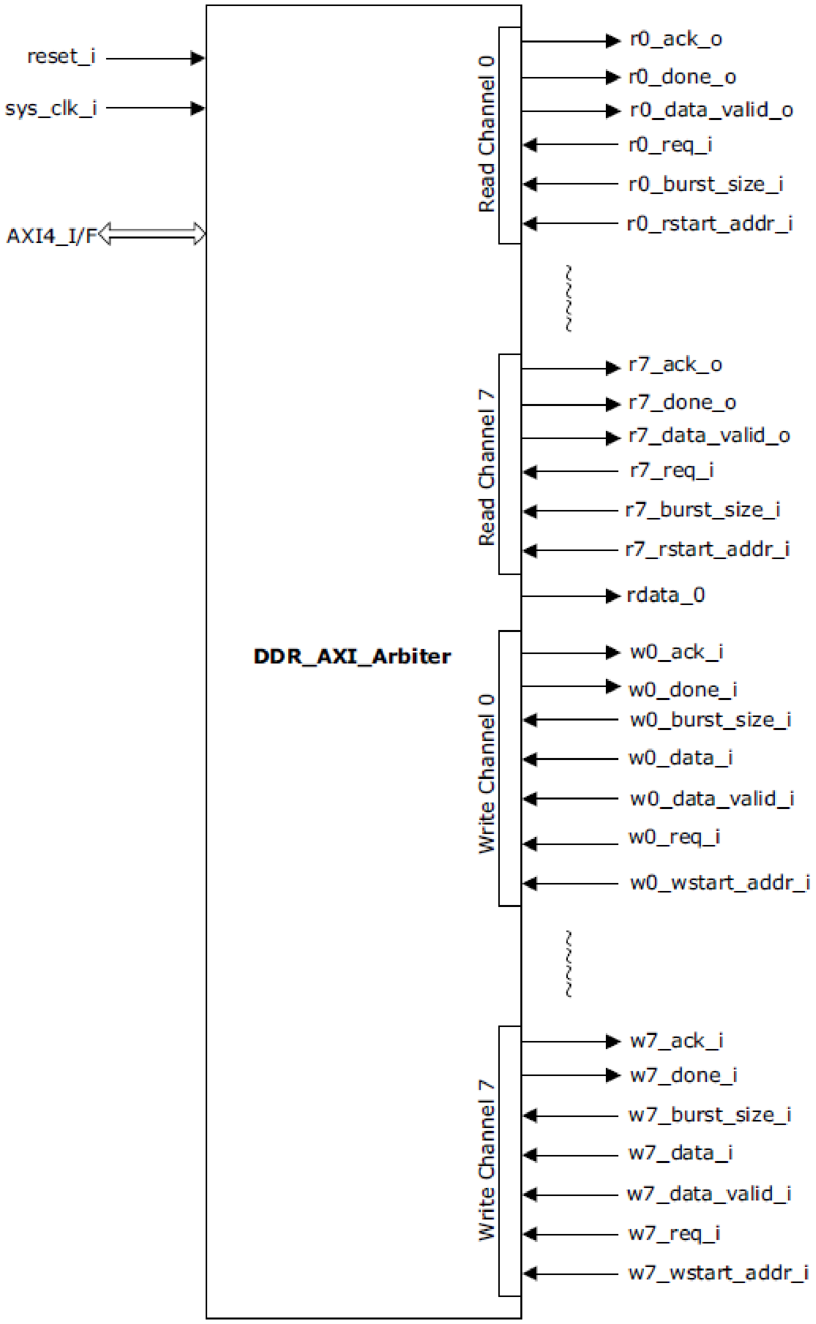
Double Data Rate (DDR) Synchronous Dynamic Random Access Memory (SDRAM) is one of the commonly used memories in video applications for buffering. SDRAM is used because of its speed which is required for fast processing in video systems.

Hardware Implementation

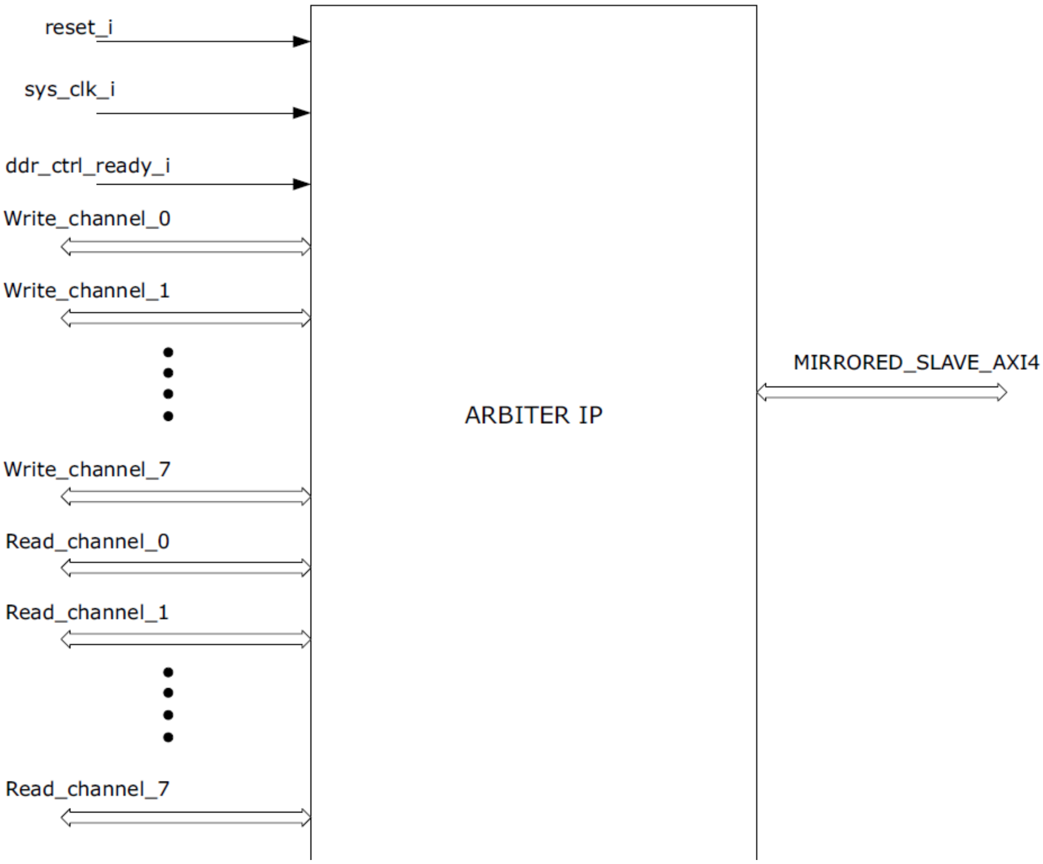
Design Description

The DDR AXI4 Arbiter provides an AXI4 master interface to the DDR on-chip controllers. The arbiter supports up to eight write channels and eight read channels. The block arbitrates between eight read channels to provide access to the AXI read channel in a first-come first-serve manner. The same way the block arbitrates between eight write channels to provide access to the AXI write channel in a first-come first-serve manner. All the eight read and write channels have equal priority. The AXI4 master interface of the Arbiter IP can be configured for various data widths ranging from 32 bits to 512 bits.
The following figure shows the top-level pin-out diagram of the DDR AXI4 Arbiter.

Top-level pin-out block diagram for Native Arbiter Interface



Top-level block diagram for Arbiter Bus Interface



A read transaction is triggered by setting the input signal r(x)_req_i high on a particular read channel. The arbiter responds by acknowledgment when it is ready to service the read request. Then it samples the starting AXI address and read burst size which are input from the external master. The channel processes the inputs and generates the required AXI transactions to read data from the DDR memory. The read data output from arbiter is common to all the read channels. During data read out, the read data valid of the corresponding channel goes high. The end of the read transaction is denoted by a read done signal when all the requested bytes are sent out. Similar to a read transaction, a write transaction is triggered by setting the input signal w(x)_req_i high. Along with the request signal, the write start address and the burst length must be provided during the request. When the arbiter is available to service the write request, it responds by sending an acknowledgment signal on corresponding channel. Then the user has to provide the write data along with the data valid signal on the channel. The number of clocks the data valid high period should match the burst length. The arbiter completes the write operation and sets the write done signal high denoting the completion of write transaction.

Inputs and Outputs

The following table lists the inputs and output ports of the DDR AXI4 Arbiter for Bus interface.

Input and Output Ports for Arbiter Bus Interface

SIGNAL NAME	DIRECTION	WIDTH	DESCRIPTION
reset_i	Input		Active Low asynchronous reset signal to design
sys_ckl_i	Input		System clock
ddr_ctrl_ready_i	Input		Receives the ready Input signal from the DDR controller
ARVALID_I_0	Input		Read request from read channel 0
ARSIZE_I_0	Input	8 bits	read burst size from read channel 0

SIGNAL NAME	DIRECTION	WIDTH	DESCRIPTION
ARADDR_I_0	Input	[31:0]	DDR address from where read has to be started for read channel 0
ARREADY_O_0	OutputOutput		Arbiter acknowledgment to read request from read channel 0
RVALID_O_0	Output		Read data valid from read channel 0
RDATA_O_0	Output	[AXI_DATA_WIDTH-1 : 0]	Read data from read channel 0
RLAST_O_0	Output		Read end of frame signal from read channel 0
BUSER_O_r0	Output		Read completion to read channel 0
ARVALID_I_1	Input		Read request from read channel 1
ARSIZE_I_1	Input	8 bits	Read burst size from read channel 1
ARADDR_I_1	Input	[31:0]	DDR address from where read has to be started for read channel 1
ARREADY_O_1	Output		Arbiter acknowledgment to read request from read channel 1
RVALID_O_1	Output		Read data valid from read channel 1
RDATA_O_1	Output	[AXI_DATA_WIDTH-1 : 0]	Read data from read channel 1
RLAST_O_1	Output		Read end of frame signal from read channel 1
BUSER_O_r1	Output		Read completion to read channel 1
ARVALID_I_2	Input		Read request from read channel 2
ARSIZE_I_2	Input	8 bits	Read burst size from read channel 2
ARADDR_I_2	Input	[31:0]	DDR address from where read has to be started for read channel 2
ARREADY_O_2	Output		Arbiter acknowledgment to read request from read channel 2
RVALID_O_2	Output		Read data valid from read channel 2
RDATA_O_2	Output	[AXI_DATA_WIDTH-1 : 0]	Read data from read channel 2
RLAST_O_2	Output		Read end of frame signal from read channel 2
BUSER_O_r2	Output		Read completion to read channel 2
ARVALID_I_3	Input		Read request from read channel 3
ARSIZE_I_3	Input	8 bits	Read burst size from read channel 3
ARADDR_I_3	Input	[31:0]	DDR address from where read has to be started for read channel 3
ARREADY_O_3	Output		Arbiter acknowledgment to read request from read channel 3
RVALID_O_3	Output		Read data valid from read channel 3
RDATA_O_3	Output	[AXI_DATA_WIDTH-1 : 0]	Read data from read channel 3
RLAST_O_3	Output		Read end of frame signal from read channel 3

SIGNAL NAME	DIRECTION	WIDTH	DESCRIPTION
ARADDR_I_4	Input	[31:0]	DDR address from where read has to be started for read channel 4
ARREADY_O_4	Output		Arbiter acknowledgment to read request from read channel 4
RVALID_O_4	Output		Read data valid from read channel 4
RDATA_O_4	Output	[AXI_DATA_WIDTH-1 : 0]	Read data from read channel 4
RLAST_O_4	Output		Read end of frame signal from read channel 4
BUSER_O_r4	Output		Read completion to read channel 4
ARVALID_I_5	Input		Read request from read channel 5
ARSIZE_I_5	Input	8 bits	Read burst size from read channel 5
ARADDR_I_5	Input	[31:0]	DDR address from where read has to be started for read channel 5
ARREADY_O_5	Output		Arbiter acknowledgment to read request from read channel 5
RVALID_O_5	Output		Read data valid from read channel 5
RDATA_O_5	Output	[AXI_DATA_WIDTH-1 : 0]	Read data from read channel 5
RLAST_O_5	Output		Read end of frame signal from read channel 5
BUSER_O_r5	Output		Read completion to read channel 5
ARVALID_I_6	Input		Read request from read channel 6
ARSIZE_I_6	Input	8 bits	Read burst size from read channel 6
ARADDR_I_6	Input	[31:0]	DDR address from where read has to be started for read channel 6
ARREADY_O_6	Output		Arbiter acknowledgment to read request from read channel 6
RVALID_O_6	Output		Read data valid from read channel 6
RDATA_O_6	Output	[AXI_DATA_WIDTH-1 : 0]	Read data from read channel 6
RLAST_O_6	Output		Read end of frame signal from read channel 6
BUSER_O_r6	Output		Read completion to read channel 6
ARVALID_I_7	Input		Read request from read channel 7
ARSIZE_I_7	Input	8 bits	Read burst size from read channel 7
ARADDR_I_7	Input	[31:0]	DDR address from where read has to be started for read channel 7
ARREADY_O_7	Output		Arbiter acknowledgment to read request from read channel 7
RVALID_O_7	Output		Read data valid from read channel 7
RDATA_O_7	Output	[AXI_DATA_WIDTH-1 : 0]	Read data from read channel 7
RLAST_O_7	Output		Read end of frame signal from read channel 7

SIGNAL NAME	DIRECTION	WIDTH	DESCRIPTION
WVALID_I_0	Input		Write data valid to write channel 0
AWVALID_I_0	Input		Write request from write channel 0
AWADDR_I_0	Input	[31:0]	DDR address to which write has to be happen from write channel 0
AWREADY_O_0	Output		Arbiter acknowledgment to write request from write channel 0
BUSER_O_0	Output		Write completion to write channel 0
AWSIZE_I_1	Input	8 bits	Write burst size for write channel 1
WDATA_I_1	Input	[AXI_DATA_WIDTH-1:0]	Video data Input to write channel 1
WVALID_I_1	Input		Write data valid to write channel 1
AWVALID_I_1	Input		Write request from write channel 1
AWADDR_I_1	Input	[31:0]	DDR address to which write has to be happen from write channel 1
AWREADY_O_1	Output		Arbiter acknowledgment to write request from write channel 1
BUSER_O_1	Output		Write completion to write channel 1
AWSIZE_I_2	Input	8 bits	Write burst size for write channel 2
WDATA_I_2	Input	[AXI_DATA_WIDTH-1:0]	Video data Input to write channel 2
WVALID_I_2	Input		Write data valid to write channel 2
AWVALID_I_2	Input		Write request from write channel 2
AWADDR_I_2	Input	[31:0]	DDR address to which write has to be happen from write channel 2
AWREADY_O_2	Output		Arbiter acknowledgment to write request from write channel 2
BUSER_O_2	Output		Write completion to write channel 2
AWSIZE_I_3	Input	8 bits	Write burst size for write channel 3
WDATA_I_3	Input	[AXI_DATA_WIDTH-1:0]	Video data Input to write channel 3
WVALID_I_3	Input		Write data valid to write channel 3
AWVALID_I_3	Input		Write request from write channel 3
AWADDR_I_3	Input	[31:0]	DDR address to which write has to be happen from write channel 3
AWREADY_O_3	Output		Arbiter acknowledgment to write request from write channel 3
BUSER_O_3	Output		Write completion to write channel 3
AWSIZE_I_4	Input	8 bits	Write burst size for write channel 4
WDATA_I_4	Input	[AXI_DATA_WIDTH-1:0]	Video data Input to write channel 4
WDATA_I_4	Input	[AXI_DATA_WIDTH-1:0]	Video data Input to write channel 4
WVALID_I_4	Input		Write data valid to write channel 4
AWVALID_I_4	Input		Write request from write channel 4
AWADDR_I_4	Input	[31:0]	DDR address to which write has to be happen from write channel 4

SIGNAL NAME	DIRECTION	WIDTH	DESCRIPTION
AWREADY_O_4	Output		Arbiter acknowledgment to write request from write channel 4
BUSER_O_4	Output		Write completion to write channel 4
AWSIZE_I_5	Input	8 bits	Write burst size for write channel 5
WDATA_I_5	Input	[AXI_DATA_WIDTH-1:0]	Video data Input to write channel 5
WVALID_I_5	Input		Write data valid to write channel 5
AWVALID_I_5	Input		Write request from write channel 5
AWADDR_I_5	Input	[31:0]	DDR address to which write has to be happen from write channel 5
AWREADY_O_5	Output		Arbiter acknowledgment to write request from write channel 5
BUSER_O_5	Output		Write completion to write channel 5
AWSIZE_I_6	Input	8 bits	Write burst size for write channel 6
WDATA_I_6	Input	[AXI_DATA_WIDTH-1:0]	Video data Input to write channel 6
WVALID_I_6	Input		Write data valid to write channel 6
AWVALID_I_6	Input		Write request from write channel 6
AWADDR_I_6	Input	[31:0]	DDR address to which write has to be happen from write channel 6
AWREADY_O_6	Output		Arbiter acknowledgment to write request from write channel 6
BUSER_O_6	Output		Write completion to write channel 6
AWSIZE_I_7	Input	8 bits	Write burst size from write channel 7
WDATA_I_7	Input	[AXI_DATA_WIDTH-1:0]	Video data Input to write channel 7
WVALID_I_7	Input		Write data valid to write channel 7
AWVALID_I_7	Input		Write request from write channel 7
AWADDR_I_7	Input	[31:0]	DDR address to which write has to be happen from write channel 7
AWREADY_O_7	Output		Arbiter acknowledgment to write request from write channel 7
BUSER_O_7	Output		Write completion to write channel 7

Input and Output ports for Native Arbiter Interface

Signal Name	Direction	Width	Description
reset_i	Input		Active low asynchronous reset signal to design
sys_clk_i	Input		System clock
ddr_ctrl_ready_i	Input		Receives the ready input signal from the DDR controller
r0_req_i	Input		Read request from master 0
r0_burst_size_i	Input	8 bits	Read burst size

Signal Name	Direction	Width	Description
r0_rstart_addr_i	Input	32 bits	DDR address from where read has to be started for read channel 0
r0_ack_o	Output		Arbiter acknowledgment to read request from master 0
r0_data_valid_o	Output		Read data valid from read channel 0
r0_done_o	Output		Read completion to master 0
r1_req_i	Input		Read request from master 1
r1_burst_size_i	Input	8 bits	Read burst size
r1_rstart_addr_i	Input	32 bits	DDR address from where read has to be started for read channel 1
r1_ack_o	Output		Arbiter acknowledgment to read request from master 1
r1_data_valid_o	Output		Read data valid from read channel 1
r1_done_o	Output		Read completion to master 1
r2_req_i	Input		Read request from master 2
r2_burst_size_i	Input	8 bits	Read burst size
r2_rstart_addr_i	Input	32 bits	DDR address from where read has to be started for read channel 2
r2_ack_o	Output		Arbiter acknowledgment to read request from master 2
r2_data_valid_o	Output		Read data valid from read channel 2
r2_done_o	Output		Read completion to master 2
r3_req_i	Input		Read request from master 3
r3_burst_size_i	Input	8 bits	Read burst size
r3_rstart_addr_i	Input	32 bits	DDR address from where read has to be started for read channel 3
r3_ack_o	Output		Arbiter acknowledgment to read request from master 3
r3_data_valid_o	Output		Read data valid from read channel 3
r3_done_o	Output		Read completion to master 3
r4_req_i	Input		Read request from master 4
r4_burst_size_i	Input	8 bits	Read burst size
r4_rstart_addr_i	Input	32 bits	DDR address from where read has to be started for read channel 4
r4_ack_o	Output		Arbiter acknowledgment to read request from master 4
r4_data_valid_o	Output		Read data valid from read channel 4
r4_done_o	Output		Read completion to master 4
r5_req_i	Input		Read request from master 5
r5_burst_size_i	Input	8 bits	Read burst size

Signal Name	Direction	Width	Description
r5_rstart_addr_i	Input	32 bits	DDR address from where read has to be started for read channel 5
r5_ack_o	Output		Arbiter acknowledgment to read request from master 5
r5_data_valid_o	Output		Read data valid from read channel 5
r5_done_o	Output		Read completion to master 5
r6_req_i	Input		Read request from master 6
r6_burst_size_i	Input	8 bits	Read burst size
r6_rstart_addr_i	Input	32 bits	DDR address from where read has to be started for read channel 6
r6_ack_o	Output		Arbiter acknowledgment to read request from master 6
r6_data_valid_o	Output		Read data valid from read channel 6
r6_done_o	Output		Read completion to master 6
r7_req_i	Input		Read request from master 7
r7_burst_size_i	Input	8 bits	Read burst size
r7_rstart_addr_i	Input	32 bits	DDR address from where read has to be started for read channel 7
r7_ack_o	Output		Arbiter acknowledgment to read request from master 7
r7_data_valid_o	Output		Read data valid from read channel 7
r7_done_o	Output		Read completion to master 7
rdata_o	Output	[AXI_DATA_WIDTH - 1:0]	Video data output from read channel
w0_burst_size_i	Input	8 bits	Write burst size
w0_data_i	Input	[AXI_DATA_WIDTH - 1:0]	Video data input to write channel 0
w0_data_valid_i	Input		Write data valid to write channel 0
w0_req_i	Input		Write request from master 0
w0_wstart_addr_i	Input	32 bits	DDR address to which write has to be happen from write channel 0
w0_ack_o	Output		Arbiter acknowledgment to write request from master 0
w0_done_o	Output		Write completion to master 0
w1_burst_size_i	Input	8 bits	Write burst size
w1_data_i	Input	[AXI_DATA_WIDTH - 1:0]	Video data input to write channel 1
w1_data_valid_i	Input		Write data valid to write channel 1
w1_req_i	Input		Write request from master 1
w1_wstart_addr_i	Input	32 bits	DDR address to which write has to be happen from write channel 1
w1_ack_o	Output		Arbiter acknowledgment to write request from master 1
w1_done_o	Output		Write completion to master 1

Signal Name	Direction	Width	Description
w2_burst_size_i	Input	8 bits	Write burst size
w2_data_i	Input	[AXI_DATA_WIDTH - 1:0]	Video data input to write channel 2
w2_data_valid_i	Input		Write data valid to write channel 2
w2_req_i	Input		Write request from master 2
w2_wstart_addr_i	Input	32 bits	DDR address to which write has to be happen from write channel 2
w2_ack_o	Output		Arbiter acknowledgment to write request from master 2
w2_done_o	Output		Write completion to master 2
w3_burst_size_i	Input	8 bits	Write burst size
w3_data_i	Input	[AXI_DATA_WIDTH - 1:0]	Video data input to write channel 3
w3_data_valid_i	Input		Write data valid to write channel 3
w3_req_i	Input		Write request from master 3
w3_wstart_addr_i	Input	32 bits	DDR address to which write has to be happen from write channel 3
w3_ack_o	Output		Arbiter acknowledgment to write request from master 3
w3_done_o	Output		Write completion to master 3
w4_burst_size_i	Input	8 bits	Write burst size
w4_data_i	Input	[AXI_DATA_WIDTH - 1:0]	Video data input to write channel 4
w4_data_valid_i	Input		Write data valid to write channel 4
w4_req_i	Input		Write request from master 4
w4_wstart_addr_i	Input	32 bits	DDR address to which write has to be happen from write channel 4
w4_ack_o	Output		Arbiter acknowledgment to write request from master 4
w4_done_o	Output		Write completion to master 4
w5_burst_size_i	Input	8 bits	Write burst size
w5_data_i	Input	[AXI_DATA_WIDTH - 1:0]	Video data input to write channel 5
w5_data_valid_i	Input		Write data valid to write channel 5
w5_req_i	Input		Write request from master 5
w5_wstart_addr_i	Input	32 bits	DDR address to which write has to be happen from write channel 5
w5_ack_o	Output		Arbiter acknowledgment to write request from master 5
w5_done_o	Output		Write completion to master 5
w6_burst_size_i	Input	8 bits	Write burst size
w6_data_i	Input	[AXI_DATA_WIDTH - 1:0]	Video data input to write channel 6
w6_data_valid_i	Input		Write data valid to write channel 6
w6_req_i	Input		Write request from master 6

Signal Name	Direction	Width	Description
w6_wstart_addr_i	Input	32 bits	DDR address to which write has to be happen from write channel 6
w6_ack_o	Output		Arbiter acknowledgment to write request from master 6
w6_done_o	Output		Write completion to master 6
w7_burst_size_i	Input	8 bits	Write burst size
w7_data_i	Input	[AXI_DATA_WIDTH - 1:0]	Video data input to write channel 7
w7_data_valid_i	Input		Write data valid to write channel 7
w7_req_i	Input		Write request from master 7
w7_wstart_addr_i	Input	32 bits	DDR address to which write has to be happen from write channel 7
w7_ack_o	Output		Arbiter acknowledgment to write request from master 7
w7_done_o	Output		Write completion to master 7

AXI I/F Signals			
Read Address Channel			
arid_o	Output	[AXI_ID_WIDTH - 1:0]	Read address ID. Identification tag for the read address group of signals.
araddr_o	Output	[31:0]	Read address. Provides the initial address of a read burst transaction. Only the start address of the burst is provided.
arlen_o	Output	[7:0]	Burst length. Provides the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
arsize_o	Output	[2:0]	Burst size. Size of each transfer in the burst.
arburst_o	Output	[1:0]	Burst type. Coupled with the size information, details how the address for each transfer within the burst is calculated. Fixed to 2’b01 à Incremental address burst.
arlock_o	Output	[1:0]	Lock type. Provides additional information about the atomic characteristics of the transfer. Fixed to 2’b00 à Normal Access.
arcache_o	Output	[3:0]	Cache type. Provides additional information about the cacheable characteristics of the transfer. Fixed to 4’b0000 à Non-cacheable and non-bufferable.

Signal Name	Direction	Width	Description
arready_o	Input		Read address ready. The slave is ready to accept an address and associated control signals. 1 = slave ready 0 = slave not ready
Read Data Channel			
rid	Input	[AXI_ID_WIDTH - 1:0]	Read ID tag. ID tag of the read data group of signals. The rid value is generated by the Slave and must match the arid value of the read transaction to which it is responding.
rdata	Input	[AXI_DATA_WIDTH - 1:0]	Read data
rresp	Input	[1:0]	Read response. The status of the read transfer. Allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
rlast	Input		Read last. Last transfer in a read burst.
rvalid	Input		Read valid. Required read data is available and the read transfer can complete. 1 = read data available 0 = read data not available
rready	Output		Read ready. Master can accept the read data and response information. 1= master ready 0 = master not ready
Write Address Channel			
awid	Output	[AXI_ID_WIDTH - 1:0]	Write address ID. Identification tag for the write address group of signals.
awaddr	Output	[31:0]	Write address. Provides the address of the first transfer in a write burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst.
awlen	Output	[7:0]	Burst length. Provides the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
awsize	Output	[2:0]	Burst size. Size of each transfer in the burst. Byte lane strobes indicate exactly which byte lanes to update.

Signal Name	Direction	Width	Description
awcache	Output	[3:0]	Cache type. Indicates the bufferable, cacheable, write-through, write-back, and allocate attributes of the transaction. Fixed to 4'b0000 à Non-cacheable and non-bufferable.
awprot	Output	[2:0]	Protection type. Indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access. Fixed to 3'b000 à Normal, secure data access.
awvalid	Output		Write address valid. Indicates that valid write address and control information are available. 1 = address and control information available 0 = address and control information not available. The address and control information remain stable until the address acknowledge signal, awready, goes HIGH.
awready	Input		Write address ready. Indicates that the slave is ready to accept an address and associated control signals. 1 = slave ready 0 = slave not ready
Write Data Channel			
wdata	Output	[AXI_DATA_WIDTH - 1:0]	Write data
wstrb	Output	[AXI_DATA_WIDTH - 8:0]	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus.
wlast	Output		Write last. Last transfer in a write burst.
wvalid	Output		Write valid. Valid write data and strobes are available. 1 = write data and strobes available 0 = write data and strobes not available
wready	Input		Write ready. Slave can accept the write data. 1 = slave ready 0 = slave not ready
Write Response Channel			
bid	Input	[AXI_ID_WIDTH - 1:0]	Response ID. The identification tag of the write response. The bid value must match the awid value of the write transaction to which the slave is responding.
bresp	Input	[1:0]	Write response. Status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
bvalid	Input		Write response valid. Valid write response is available. 1 = write response available 0 = write response not available

Signal Name	Direction	Width	Description
bready	Output		Response ready. Master can accept the response information. 1 = master ready 0 = master not ready

Configuration Parameters

The following table lists the configuration parameters used in the hardware implementation of the DDR AXI4

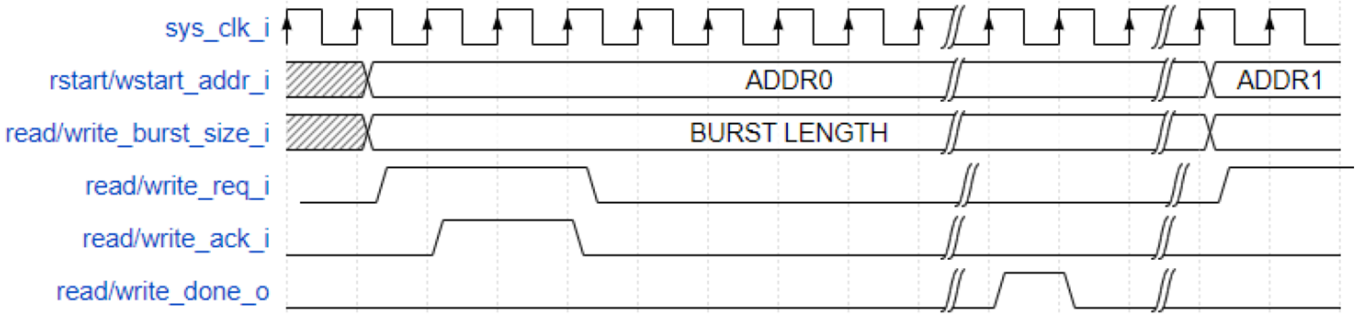
Arbiter. These are generic parameters and can be varied based on the application requirements.

Signal Name	Description
AXI ID Width	Defines the AXI ID width.
AXI Data Width	Defines the AXI data width.
Number of Read channels	Options to select the required no of write channels from the dropdown menu ranging from one channel to eight write channels.
Number of Write channels	Options to select the required no of read channels from the dropdown menu ranging from one channel to eight read channels.
AXI4_SELECTION	Options to select between AXI4_MASTER and AXI4_MIRRORED_SLAVE.
Arbiter Interface	Option to select the bus interface.

Timing Diagram

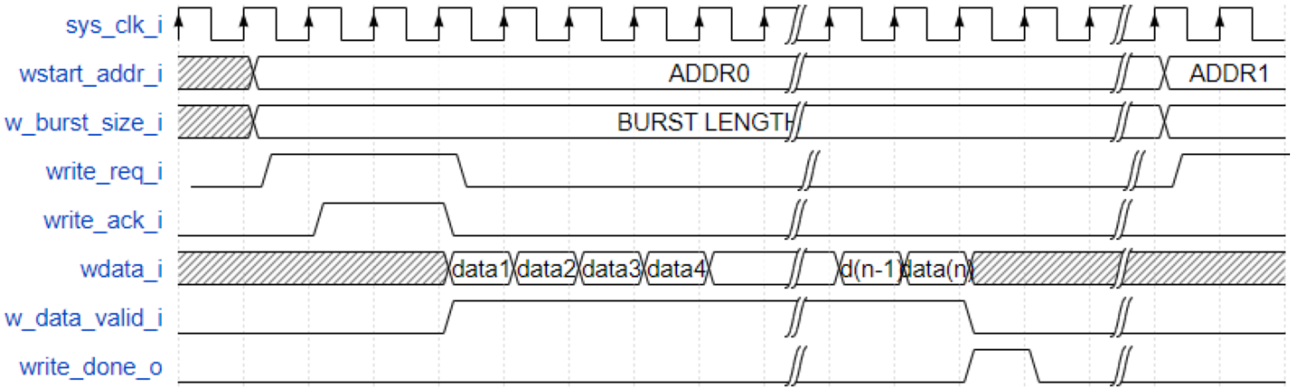
The following figure shows the connection of the read and write request inputs, starting memory address, write inputs from the external master, read or write acknowledgment, and read or write completion inputs given by arbiter.

Timing Diagram for Signals used in Writing/Reading through AXI4 Interface



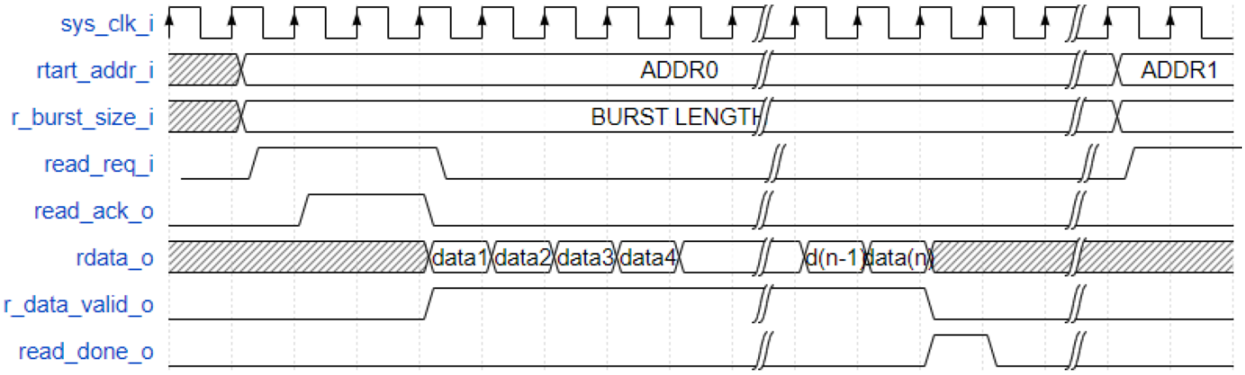
The following figure shows the connection between the write data input from the external master along with the data input valid. This is same for eight write channels.

Timing Diagram for Writing into Internal Memory



The following figure shows the connection between the read data output towards the external master along with the data output valid for all the eight read channels.

Timing Diagram for Data Received through DDR AXI4 Arbiter for Read Channels



License

The IP can be used in RTL mode without any license.

Installation Instructions

The core must be installed into Libero software. It is done automatically through the Catalog update function in Libero, or the CPZ file can be manually added using the Add Core catalog feature. Once the CPZ file is installed in Libero, the core can be configured, generated, and instantiated within SmartDesign for inclusion in the Libero project.
For further instructions on core installation, licensing, and general use, refer to the Libero SoC Online Help.

Resource Utilization

DDR AXI4 Arbiter block is implemented on a PolarFire® FPGA (MPF300T -1FCG1152E package) for four write channels and four read channels configuration.

Resource	Usage
DFFs	2822
4 input LUTs	2999
MACC	0
LSRAM 18K	13
uSRAM 1K	1

Documents / Resources

UG0950 User Guide DDR_AXI4_Arbiter	[Microsemi UG0950 DDR AXI4 Arbiter IP](#) [pdf] User Guide UG0950 DDR AXI4 Arbiter IP, UG0950, DDR AXI4 Arbiter IP, AXI4 Arbiter IP, Arbiter IP

References

- Microsemi \| Semiconductor & System Solutions \| Power Matters
- microsemi.com/index.php?option=com_docman&task=doc_download&gid=132044