HB0853
Handbook
CoreDDR_LiteAXI v2.0

Microsemi Corporate 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
E-mail: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) 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. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

Contents

Figures

Tables

1 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.

1.1 Revision 1.0

Revision 1.0 is the first publication of this document. Created for CoreDDR_LiteAXI v2.0.

2 Introduction

CoreDDR_LiteAXI IP converts AXI4 transactions to PolarFire DDR Native interface transactions that access DDR memory through AXI4 interface. This IP is a lighter version of AXI to DDR Native interface, which supports only the AXI INCR transactions.

2.1 Features

CoreDDR_LiteAXI supports the following features:

- AXI4 protocol
- 1:1 synchronous clock
- Interface data widths: 128, 256, and 512-bits
- 32 to 40-bit AXI address bus
- Single or burst transfers
- Only AXI4 increment transfers
- Maximum of 16-bit ID width

2.2 Core Version

This handbook is for CoreDDR_LiteAXI version 2.0.

2.3 Supported Families

- PolarFire®

2.4 Device Utilization and Performance

Utilization and performance data is as shown in Table 1, for PolarFire (MPF300T) device family. The data provided in these tables are indicative only. The overall device utilization and performance of the core is system dependent.

Table 1 • **CoreDDR_LiteAXI Utilization and Performance**

Data Bus Width (bits)	Logic Elements				Memory Blocks		Performance in MHz
	Combinational	%	Sequential	%	uSRAM	LSRAM	
128	2949	0.98	2554	0.85	3	9	243
256	4624	1.54	4331	1.45	4	16	223
512	7937	2.65	7834	2.62	4	29	232

Note: The data in this table was achieved using default synthesis and layout settings. Frequency (in MHz) was set to 200 and speed grade set to -1.

2.5 References

AMBA AXI protocol specification:

UG0676: PolarFire FPGA Memory Controller User Guide

TU0775: PolarFire FPGA: Building a RISC-V Processor Subsystem Tutorial

3 Functional Description

CoreDDR_LiteAXI consists of an AXI4 slave interface, read and write data FIFOs, read ID FIFO, read and write response generators, and Native Interface transaction generator. The following figure shows the high level block diagram of CoreDDR_LiteAXI.

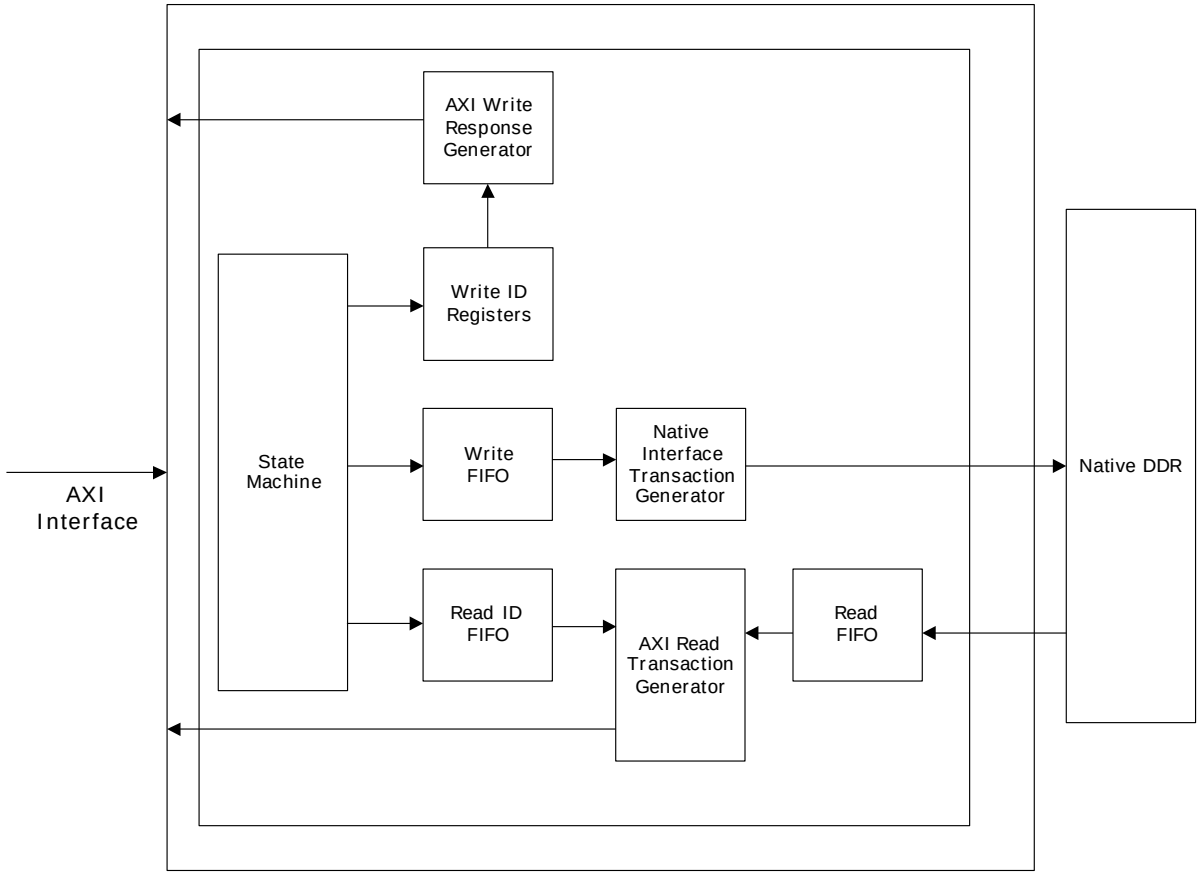
The state machine analyzes the AXI4 transactions and efficiently sends those transactions to DDR Native Interface. This IP handles the AXI handshaking, and monitors the FIFO levels to make sure they do not overflow.

The read and write FIFOs are used to allow posting of transactions and to hold data that may be needed for read and write transactions.

The Native Interface transaction generator initiates the transactions on the DDR native interface depending on the AXI transaction size. In case of write, the number of bytes are calculated based on the sum of the bytes in each beat (as per the number of ones in WSTRB). CoreDDR_LiteAXI uses the DDR Native interface data mask signal (I_dm_in) according to the WSTRB signal (that is, the DDR subsystem must be configured to use the data mask signal).

AXI read transaction generator block provides data to AXI read channel when read FIFO is not empty.

Figure 1 • **High Level Block Diagram of CoreDDR_LiteAXI**



The CoreDDR_Lite AXI IP handles the simultaneous read and write transactions to the same address by holding the read transaction until the completion of write transaction.

Limitations:

1. FIXED and WRAP type bursts are not supported.
2. AXI transactions with data before address are not supported.

4 Interface

4.1 Ports

The following table describes the CoreDDR_LiteAXI I/O signals.

Table 2 • **CoreDDR_LiteAXI I/O Signals**

Port Name	Type	Description
Clocks and Reset		
ACLK	Input	Clock for user logic generated by the DDR subsystem (SYS_CLK). All native interface signals are synchronous to this clock.
ARESET_N	Input	Active-low asynchronous reset.
AXI interface signals		
AWID[AXI_ID_WIDTH - 1:0]	Input	Write address ID.
AWADDR[AXI_ADDR_WIDTH - 1:0]	Input	Write address.
AWLEN[7:0]	Input	Burst length.
AWSIZE[2:0]	Input	Burst size.
AWBURST[1:0]	Input	Burst type. Only INCR burst type is supported.
AWVALID	Input	Write address valid.
AWREADY	Output	Write address ready.
WDATA[AXI_DATA_WIDTH – 1:0]	Input	Write data.
WSTRB[AXI_DATA_WIDTH/8 -1:0]	Input	Write strobes.
WLAST	Input	Write last.
WVALID	Input	Write valid.
WREADY	Output	Write ready.
BID[AXI_ID_WIDTH – 1:0]	Output	Response ID tag.
BRESP[1:0]	Output	Write response.
BVALID	Output	Write response valid.
BREADY	Input	Response ready.
ARID[AXI_ID_WIDTH – 1:0]	Input	Read address ID.
ARADDR[AXI_ADDR_WIDTH – 1:0]	Input	Read address.
ARLEN[7:0]	Input	Burst length.
ARSIZE[2:0]	Input	Burst size.
ARBURST[1:0]	Input	Burst type.
ARVALID	Input	Read address valid.
ARREADY	Output	Read address ready.
RID[AXI_ID_WIDTH – 1:0]	Output	Read ID tag.
RDATA[AXI_DATA_WIDTH – 1:0]	Output	Read data.
RRESP[1:0]	Output	Read response.
RLAST	Output	Read last.

Table 2 • **CoreDDR_LiteAXI I/O Signals**

RVALID	Output	Read valid.
RREADY	Input	Read ready.
Native Interface Signals		
L_ADDR[LOCAL_BUS_ASIZE – 1:0]	Input	Native interface address sizes: DDR4 = 39 bits, DDR3 = 36 bits, and LPDDR3 = 36 bits.
L_B_SIZE[BURST_SIZE_WIDTH – 1:0]	Input	Native interface burst length in terms of bytes. It must be in multiples of the native interface bus width.
L_R_REQ	Input	Native interface read request.
L_W_REQ	Input	Native interface write request.
L_BUSY	Output	Specifies that the subsystem is busy and is not accepting new requests. A command is accepted on any clock cycle where L_R_REQ or L_W_REQ is set, and L_BUSY is low. If L_BUSY is high when L_R_REQ or L_W_REQ is set, the request may be kept asserted (along with the desired L_ADDR, L_B_SIZE and L_AUTO_PCH values) until L_BUSY goes low.
L_D_REQ	Output	Requests data on the native interface write data bus (L_DATAIN) during a write transaction. Asserts one clock cycle prior to when data is required.
L_R_VALID	Output	Data-valid indication for data on the native interface read data bus (L_DATAOUT).
L_DATAIN[AXI_DATA_WIDTH – 1:0]	Input	Input data bus. This data bus is eight times the width of the SDRAM device data bus. Memory width (bits): 16, 32, 64 Input data bus (bits): 128, 256, 512
L_DATAOUT[AXI_DATA_WIDTH – 1:0]	Output	Output data bus. This data bus width is same as L_DATAIN width.
L_DM_IN[AXI_DATA_WIDTH/8 – 1:0]	Input	Individual byte masks during data write.
Note: All signals are Active High (logic'1') unless noted.		

4.2 Configuration Parameters

The following table describes the configurable parameters for CoreDDR_LiteAXI. If a setting other than the default is required, use the configuration dialog box in SmartDesign to select appropriate values for the configurable options.

Table 3 • **CoreDDR_LiteAXI Configuration Options**

Parameter Name	Valid Range	Default	Description
AXI_ID_WIDTH	4-16	4	AXI4 slave IF ID width.
AXI_ADDR_WIDTH	32-40	32	AXI4 address width. Must be less or equal to LOCAL_BUS_ASIZE.
AXI_DATA_WIDTH	128, 256, 512	128	AXI4 data width. It should be same as Native IF data bus width of DDR subsystem.
LOCAL_BUS_ASIZE	32-40	36	Native IF address width.

5 Timing Diagrams

CoreDDR_LiteAXI IP complies with the AMBA® AXI4 protocol specifications timings.

6 Tool Flow

6.1 License

CoreDDR_LiteAXI does not require a license.

6.2 RTL

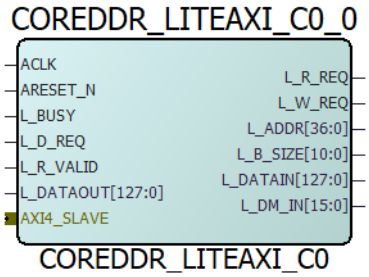
The complete RTL source code is provided for the core and testbenches.

6.3 SmartDesign

CoreDDR_LiteAXI is pre-installed in the SmartDesign IP Deployment design environment. An example instantiated view is as shown in Figure 2. The core can be configured using the configuration GUI within SmartDesign, as shown in Figure 3.

For information on using SmartDesign to instantiate and generate cores, refer to the Using DirectCore in Libero® SoC user guide.

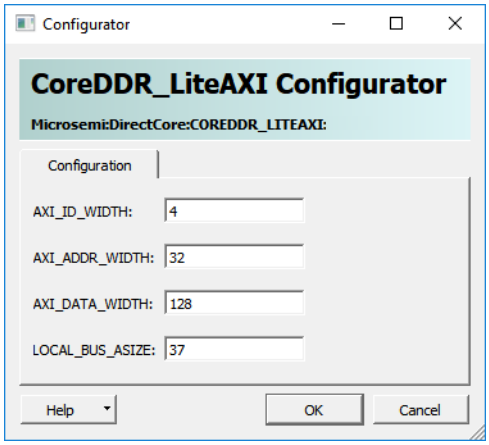
Figure 2 • **SmartDesign CoreDDR_LiteAXI Instance View**



6.4 Configuring CoreDDR_LiteAXI in SmartDesign

The following figure shows the CoreDDR_LiteAXI Configurator in SmartDesign.

Figure 3 • **SmartDesign CoreDDR_LiteAXI Configurator**



6.5 Simulation Flows

The User Testbench for CoreDDR_LiteAXI is not included.

6.6 Synthesis in Libero SoC

Click the **Synthesis** icon in Libero SoC. The Synthesis window appears, displaying the Synplicity® project. Set synplicity to use the Verilog 2001 standard, if Verilog is being used. To run synthesis, select the **Run** icon.

6.7 Place-and-Route in Libero SoC

Click the **Layout** icon in the Libero SoC to invoke Designer. CoreDDR_LiteAXI requires no special place-and-route settings.

7 Register Map and Descriptions

CoreDDR_LiteAXI does not contain any registers.

8 System Integration

The following is an example of the system integration diagram for CoreDDR_LiteAXI.

This design example contains the CoreAXI_LiteAXI (axi2ni_0) which is interfaced with Mi-V soft processor and DDR3 subsystem. The soft processor can perform read/writes or execute code from DDR3 memory. This design example is similar to the *TU0775: PolarFire FPGA: Building a RISC-V Processor Subsystem Tutorial* design and uses the DDR3 Native interface along with the CoreDDR_LiteAXI IP.

Figure 4 • **CoreDDR_LiteAXI System Integration Diagram**