LMB BRAM Interface Controller (v1.00b)

DS452 December 1, 2005 **Product Specification**

Introduction

This document provides the design specification for the Local Memory Bus (LMB) Block RAM (BRAM) Interface Controller.

The LMB BRAM Interface Controller connects to an lmb_v10 bus.

This controller supports the LMB v1.0 bus protocol and byte-enable architecture. Any access size up to the width of the LMB data bus is permitted. The LMB BRAM Interface Controller is the interface between the LMB and the bram_block peripheral. A BRAM memory subsystem consists of the controller along with the bram_block peripheral.

Features

- LMB v1.0 bus interfaces with byte enable support
- Used in conjunction with bram_block peripheral to provide fast BRAM memory solution for MicroBlaze™ ILMB and DLMB ports.
- Supports byte, half-word, and word transfers

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	QPro™-R Virtex™-II, QPro Virtex-II, Spartan™-II, Spartan-IIE, Spartan-3, Spartan-3E, Virtex, Virtex-II, Virtex-II Pro, Virtex-4, Virtex-E	
Version of Core	lmb_bram_if_cntlr	v1.00b
Resources Used		
	Min	Max
Slices	N/A	N/A
LUTs	6	6
FFs	2	2
Block RAMs	0	0
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	ISE 6.3i or higher	
Verification	N/A	
Simulation	ModelSim SE/PE 5.7b or later	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

Functional Description

The LMB BRAM Interface Controller is the interface between the LMB and the bram_block peripheral. A BRAM memory subsystem consists of the controller along with the bram_block peripheral.

The input/output signals of the LMB BRAM interface controller are shown in Figure 1 and are listed and described in Table 1. See the description of LMB Signals in the "MicroBlaze Bus Interfaces" chapter in the *MicroBlaze Processor Reference Guide.*

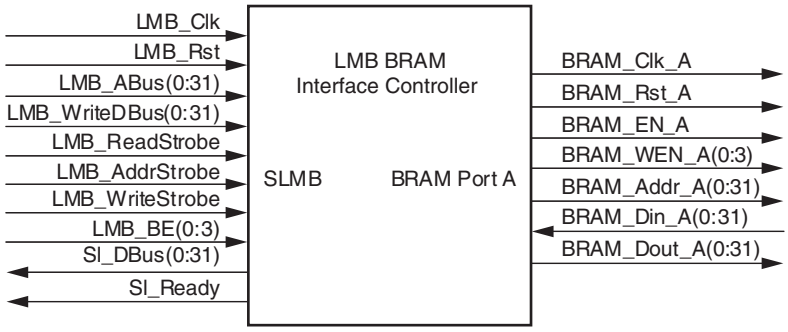

Figure 1: **Dual LMB BRAM Interface Controller Block Diagram**

LMB BRAM Interface Controller I/O Signals

The I/O ports and signals for the LMB BRAM Interface Controller are listed and described in Table 1.

Table 1: **LMB BRAM Interface Controller I/O Signals**

Port Name	MSB:LSB	I/O	Description
LMB_Clk		I	LMB Clock
LMB_Rst		I	LMB Reset (Active High)
LMB_ABus	0:C_LMB_AWIDTH-1	I	LMB Address Bus
LMB_WriteDBus	0:C_LMB_DWIDTH-1	I	LMB Write Data Bus
LMB_ReadStrobe		I	LMB Read Strobe
LMB_AddrStrobe		I	LMB Address Strobe
LMB_WriteStrobe		I	LMB Write Strobe
LMB_BE	0:C_LMB_DWIDTH/8-1	I	LMB Byte Enable Bus
SI_DBus	0:C_LMB_DWIDTH-1	O	LMB Read Data Bus
SI_Ready		O	LMB Data Ready
BRAM_Rst_A		O	BRAM Reset
BRAM_Clk_A		O	BRAM Clock
BRAM_EN_A		O	BRAM Enable
BRAM_WEN_A		O	BRAM Write Enable

Table 1: **LMB BRAM Interface Controller I/O Signals *(Contd)***

Port Name	MSB:LSB	I/O	Description
BRAM_Addr_A	0:C_LMB_AWIDTH-1	O	BRAM Address
BRAM_Din_A	0:C_LMB_DWIDTH-1	I	BRAM Data Input
BRAM_Dout_A	0:C_LMB_DWIDTH-1	O	BRAM Data Output

LMB BRAM Interface Controller Parameters

To allow the user to obtain an LMB BRAM Interface Controller that is uniquely tailored a specific system, certain features can be parameterized in the LMB BRAM Interface Controller design. This allows the user to configure a design that only utilizes the resources required by the system, and operates with the best possible performance. The features that can be parameterized in Xilinx LMB BRAM Interface Controller designs are shown in Table 2.

Table 2: LMB **BRAM Interface Controller Parameters**

Parameter Name	Feature/Description	Allowable Values	Default Value	VHDL Type
C_BASEADDR	LMB BRAM Base Address	Valid Address Range(2)	None(1)	std_logic_vector
C_HIGHADDR	LMB BRAM HIGH Address	Valid Address Range(2)	None(1)	std_logic_vector
C_MASK	LMB Decode Mask	Valid decode mask(3)	0x00800000	std_logic_vector
C_LMB_AWIDTH	LMB Address Bus Width	32	32	integer
C_LMB_DWIDTH	LMB Data Bus Width	32	32	integer

Notes:
1. No default value is specified for C_BASEADDR and C_HIGHADDR to insure that the actual value is set; if the value is not set, a compiler error is generated. These generics must be a power of 2. C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR +1.
2. The range specified by C_BASEADDR and C_HIGHADDR must comprise a complete, contiguous power-of-two range, such that range = 2^n, and the n least significant bits of C_BASEADDR must be zero.
3. The decode mask determines which bits are used by the LMB decode logic to decode a valid access to LMB.

C_BASEADDR

Base address decoded by this core.

C_HIGHADDR

High address decoded by this core.

C_MASK

If using Embedded Development Kit, this bit can be automatically set by Platform Generator tool and users do not need to setup the value. The address mask indicates which bits are used in the LMB decode to decode that a valid address is present on the LMB. Any bits that are set to '1' in the mask indicate that the address bit in that position is used to decode a valid LMD access. All other address bits are considered don't cares for the purpose of decoding LMB accesses. The platform generation tool may limit the user's choice for the address mask: the most restrictive case is that only a single bit may be set in the mask. Consult the platform generation tool documentation and informational messages for more information.

C_LMB_AWIDTH

LMB Address Bus Width. Specifies the width in bits of the LMB address buses connected to this core. The default is 32 bits.

C_LMB_DWIDTH

LMB Data Bus Width. Specifies the width in bits of the LMB data buses connected to this core. The default is 32 bits.

Allowable Parameter Combinations

There are no restrictions on parameter combinations.

Parameter - Port Dependencies

The width of many of the BRAM Interface Controller signals depends on the number of memories in the system and the width of the various data and address buses. The dependencies between the BRAM design parameters and I/O signals are shown in Table 3.

Table 3: **Parameter-Port Dependencies**

Parameter Name	Ports (Port width depends on parameter)
C_BASEADDR	none
C_HIGHADDR	none
C_MASK	none
C_LMB_AWDITH	LMB_ABus
C_LMB_DWIDTH	LMB_BE, LMB_WriteDBus, SI_DBus, BRAM_Din_A, BRAM_Dout_A

LMB BRAM Interface Controller Register Descriptions

The BRAMs are organized as big-endian data. The bit and byte labeling for the big-endian data types is shown in Figure 1.

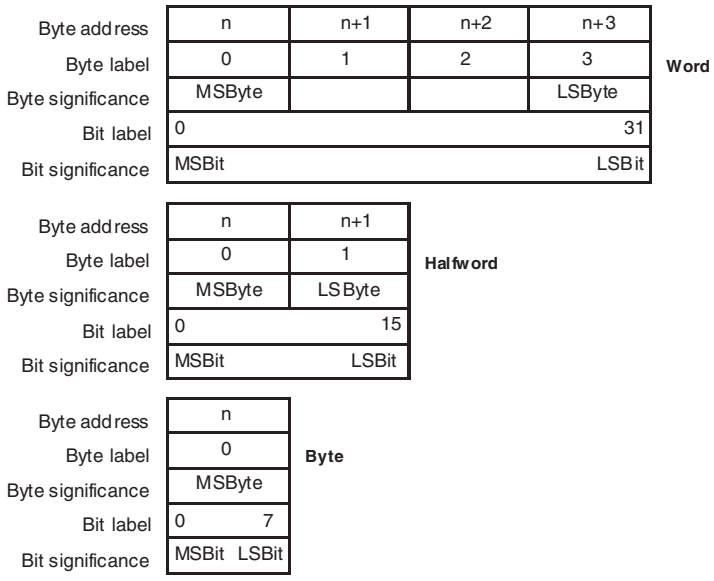

Figure 2: **Big-Endian Data Types**

LMB BRAM Interface Controller Interrupt Descriptions

Not applicable.

Design Implementation

Design Tools

The LMB BRAM interface controller design is hand written.

XST is the synthesis tool used for synthesizing the LMB BRAM interface controller. The NGC netlist output from XST is then input to the Xilinx Alliance tool suite for actual device implementation.

Target Technology

The intended target technology is an FPGA in one of the following families: Virtex, Virtex-E, Spartan-II, Spartan-IIE, Spartan--3, Virtex-II, QPro Virtex-II, QPro-R Virtex-II, Virtex-II Pro, or Virtex-4.

Device Utilization and Performance Benchmarks

Since the BRAM interface controller is a module that will be used with other design pieces in the FPGA, the utilization and timing numbers reported in this section are just estimates. As the BRAM interface controller is combined with other pieces of the FPGA design, the utilization of FPGA resources and timing of the BRAM interface controller design will vary from the results reported here. These numbers do not reflect any BRAM resources used.)

Table 4: **BRAM Interface Controller FPGA Performance and Resource Utilization Benchmarks (Virtex-II Pro -6**

Parameter Values		Device Resources	
C_LMB_DWIDTH	C_LMB_AWIDTH	Slice Flip- Flops	4-input LUTs
32	32	1	5

Notes:
1. These benchmark designs contain only the BRAM interface controller with registered inputs/outputs without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user conditions.

Programming Model

Supported Memory Sizes

The following sizes are supported for Virtex, Virtex-E, and Spartan-II:

Table 5: **Supported BRAM memory sizes for Virtex, Virtex-E, and Spartan-II**

Host Bus Size (bits)	Supported Memory Sizes (Bytes)
32	2 KB, 4 KB, 8 KB, 16 KB
64	4 KB, 8 KB, 16 KB, 32 KB

The following sizes are supported for Virtex-II and Virtex-II Pro:

Table 6: **Supported BRAM memory sizes for Virtex-II and Virtex-II Pro**

Host Bus Size (bits)	Supported Memory Sizes (Bytes)
32	8 KB, 16 KB, 32 KB, 64 KB
64	16 KB, 32 KB, 64 KB, 128 KB

Example Base Address, High Address Specifications

The base address (C_BASEADDR) and high address (C_HIGHADDR) must specify a valid range for the BRAM that is attached to the BRAM Controller. The range (C_HIGHADDR – C_BASEADDR) specified by the high address and base address must be equal to 2^n bytes, where n is a positive integer and 2^n is a valid memory size as shown above. In addition, the n least significant bits of C_BASEADDR must be equal to 0.:

Table 7: **Example Address Range Specifications**

Memory Size (Bytes)	C_BASEADDR	C_HIGHADDR
2 K	0x10000000	0x100007FF
4 K	0x50000000	0x50000FFF
8 K	0x24000000	0x24001FFF
16 K	0xE0000000	0xE0003FFF
32 K	0x3FF00000	0x3FF07FFF
64 K	0x82000000	0x8200FFFF
128 K	0xB0000000	0xB001FFFF

LMB Timing

This section describes the basic read and write timing for the LMB. For more information refer to the "MicroBlaze Bus Interfaces" chapter in the *MicroBlaze Processor Reference Guide*.

Back-to-Back Write Operation (Typical LMB access - 2 clocks per write)

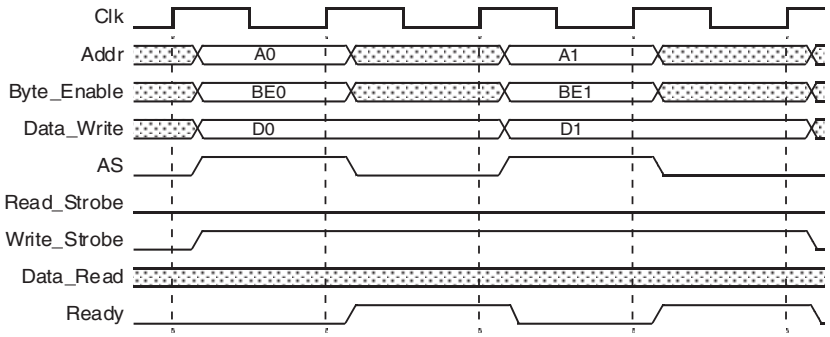

Figure 3: **LMB Back-to-Back Write Operation**

Single Cycle Back-to-Back Read Operation (Typical I-side access - 1 clock per read)

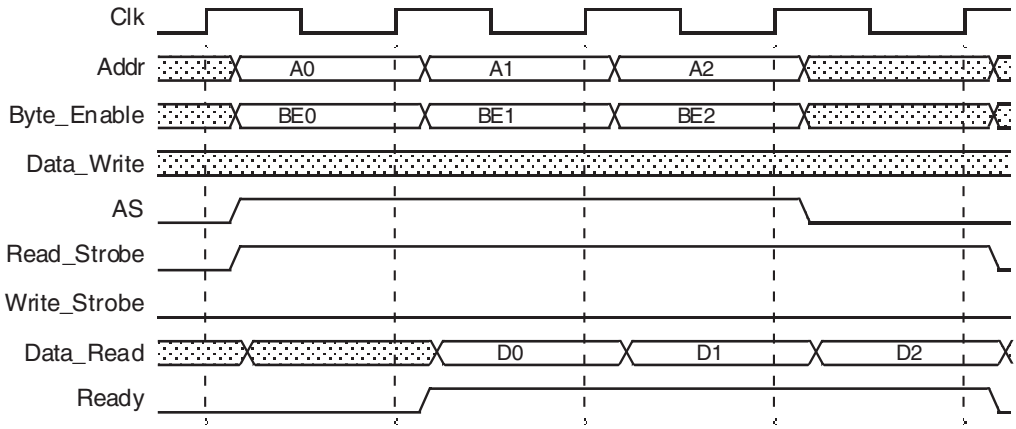

Figure 4: **LMB Single Cycle Back-to-Back Read Operation**

Back-to-Back Mixed Read/Write Operation (Typical D-side timing)

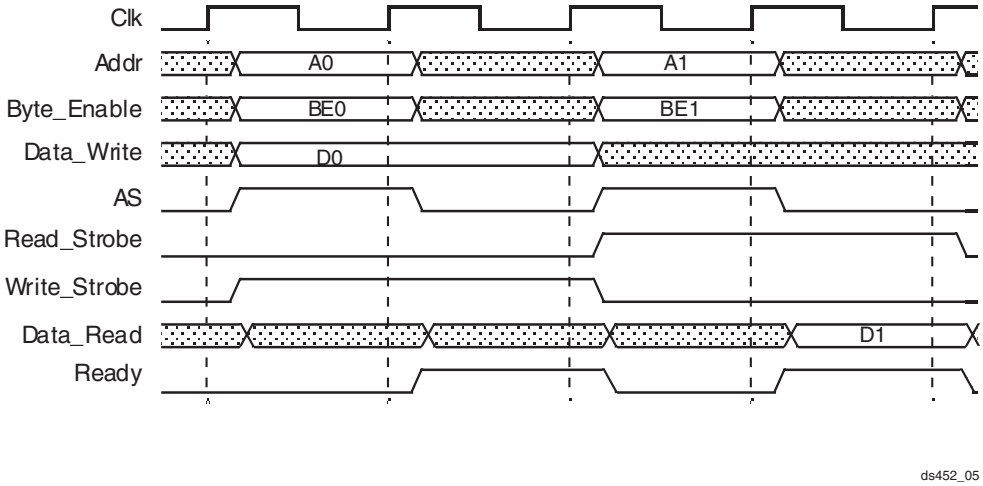

Figure 5: **Back-to-Back Mixed Read/Write Operation**

Specification Exceptions

Not applicable.

Reference Documents

None.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/10/02	1.0	Initial release.
11/11/02	1.1	Modified supported memory sizes table
01/01/03	1.2	Added device utilization
01/07/03	1.3	Update for EDK SP3
07/10/03	1.4	Update to new template
09/17/03	1.5	Correct trademarks, add summary to pdf
11/17/03	1.6	Add parameter definitions
01/26/04	1.7	Updated trademarks and copyright
08/13/04	1.8	Update for Gmm; updated content format, reviewed and corrected trademark usage.
4/4/05	1.9	Update for EDK 7.1.1 SP1 release; updated trademarks and supported device listing.
8/17/05	2.0	Converted to new DS template; ; incorporated CR206184; updated figures to graphic standards; reformatted tables.
12/1/05	2.1	Added Spartan-3E to supported device listing.