

## Introduction

The PLB BRAM Interface Controller is a module that attaches to the PLB (Processor Local Bus).

This controller supports the PLB V3.4 byte enable architecture. Any access size up to the width of the PLB data bus is permitted. The PLB BRAM Interface Controller is the interface between the PLB and the bram\_block peripheral. A BRAM memory subsystem consists of the controller along with the actual BRAM components that are included in the bram\_block peripheral. If the text-based Microprocessor Hardware Specification (MHS) file is used for design entry, then the bram controller and bram\_block must both be explicitly instantiated.

## Features

- PLB V3.4 bus interface with byte\_enable support
- Used in conjunction with bram\_block peripheral to provide total BRAM memory solution
- Supports a wide range of memory sizes
- Handles Virtex™, Virtex™E, Spartan™II, Virtex™II and Virtex™II Pro BRAM
- Handles byte, half-word, word and double word single transfers
- Burst and cacheline support enabled by a design parameter

## PLB BRAM Interface Controller Parameters

To allow you to obtain an PLB BRAM Interface Controller that is uniquely tailored for your system, certain features can be parameterized in the PLB BRAM Interface Controller design. This allows you to configure a design that only utilizes the resources required by your system, and operates with the best possible performance. The features that can be parameterized in Xilinx PLB BRAM Interface Controller designs are shown in [Table 1](#).

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex™, Virtex™E, Spartan™II, Virtex™II and Virtex™II Pro	
Version of Core	opb_bram_if_cntlr	v1.00a
Resources Used		
	Min	Max
Slices	139	265
LUTs	182	347
FFs	150	261
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	5.1i or later	
Verification	N/A	
Simulation	ModelSim SE/EE 5.6e or later	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

Table 1: PLB BRAM Interface Controller Parameters

Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
PLB BRAM Base Address	C_BASEADDR	Valid Address Range <sup>(2)</sup>	None <sup>(1)</sup>	std_logic_vector
PLB BRAM HIGH Address	C_HIGHADDR	Valid Address Range <sup>(2)</sup>	None <sup>(1)</sup>	std_logic_vector
Include support for PLB burst and cacheline transfers	C_INCLUDE_BURST_CACHELN_SUPPORT	0 = don't include logic to support PLBburst and cacheline transfers 1 = include logic to support PLBburst and cacheline transfers	0	string
PLB Data Bus Width	C_PLB_DWIDTH	64	64	integer
PLB Address Bus Width	C_PLB_AWIDTH	32	32	integer
PLB Clock Period	C_PLB_CLK_PERIOD_PS	>0	40000	integer
Number of Masters	C_NUM_MASTERS	1 - 8	4	integer
Width of Master ID Bus <b>AUTOCALCULATED</b>	C_PLB_MID_WIDTH	roundup(log <sub>2</sub> (C_NUM_MASTERS))	2	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.
2. The range specified by C\_BASEADDR and C\_HIGHADDR must comprise a complete, contiguous power-of-two range, such that range = 2<sup>n</sup>, and the n least significant bits of C\_BASEADDR must be zero.
- 3.

## Programming Model

### Supported Memory Sizes

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

Table 2: Supported BRAM memory sizes for Virtex™, Virtex™E, and Spartan™II

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

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

Table 3: Supported BRAM memory sizes for Virtex™II and Virtex™II Pro

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

### 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 4: Example Address Range Specifications

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

Discontinued IP

### IPIF Interconnect, (IPIC) Interface to BRAM Interface Controller

Figure 1 illustrates all PLB slave interface input/output signals. See PLB Signals in the Processor Local Bus Architecture Specification (v3.4) for detailed functional description of the signals.

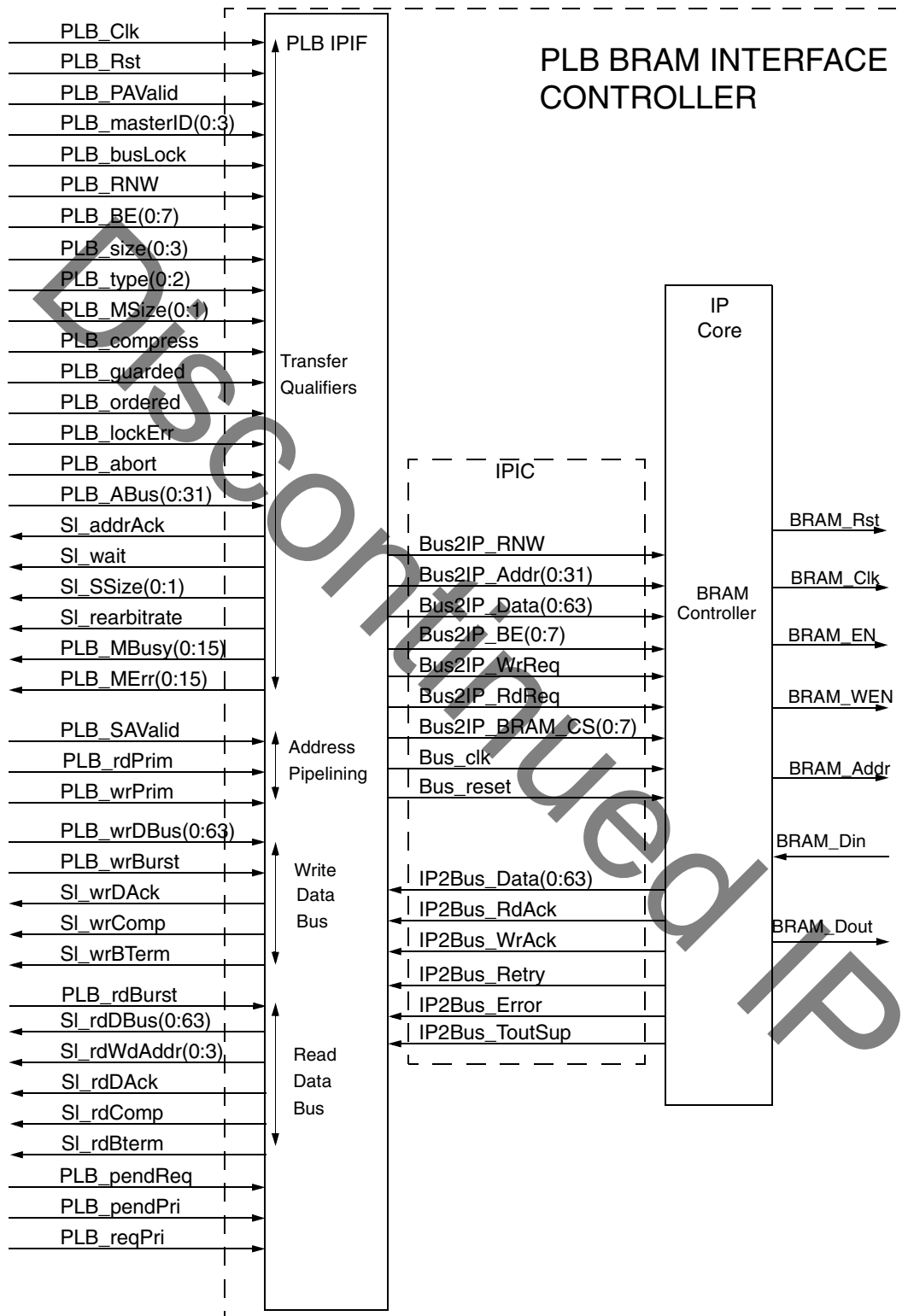


Figure 1: PLB Slave Interface

## PLB BRAM Interface Controller I/O Signals

The I/O signals for the PLB BRAM Interface Controller are listed in [Table 5](#).

**Table 5: PLB BRAM Interface Controller I/O Signals**

Signal Name	Interface	I/O	Description
PLB_abort	PLB	I	PLB abort bus request indicator
PLB_ABus(0:C_PLB_AWIDTH-1)	PLB	I	PLB address bus
PLB_BE(0:(C_PLD_DWIDTH / 8) -1)	PLB	I	PLB byte enables
PLB_busLock	PLB	I	PLB bus lock
PLB_compress	PLB	I	PLB compressed data transfer indicator
PLB_guarded	PLB	I	PLB guarded transfer indicator
PLB_lockErr	PLB	I	PLB lock error indicator
PLB_masterID(0:C_PLB_MID_WIDTH-1)	PLB	I	PLB current master indicator
PLB_ordered	PLB	I	PLB synchronize transfer indicator
PLB_PAVValid	PLB	I	PLB primary address valid indicator
PLB_rdBurst	PLB	I	PLB burst read transfer indicator
PLB_rdPrim	PLB	I	PLB secondary to primary read request indicator
PLB_RNW	PLB	I	PLB read not write
PLB_SAVValid	PLB	I	PLB secondary address valid indicator
PLB_size(0:3)	PLB	I	PLB transfer size
PLB_type(0:2)	PLB	I	PLB transfer type
PLB_wrBurst	PLB	I	PLB burst write transfer indicator
PLB_wrDBus(0:C_PLB_DWIDTH -1)	PLB	I	PLB write data bus
PLB_wrPrim	PLB	I	PLB secondary to primary write request indicator
PLB_MSize(0:1)	PLB	I	PLB master data bus size
SI_addrAck	PLB	O	Slave address acknowledge
SI_MBusy(0:C_NUM_MASTERS-1)	PLB	O	Slave busy indicator
SI_MErr(0:C_NUM_MASTERS-1)	PLB	O	Slave error indicator
SI_rdBTerm	PLB	O	Slave terminate read burst transfer
SI_rdComp	PLB	O	Slave read transfer complete indicator
SI_rdDAck	PLB	O	Slave read data acknowledge
SI_rdBBus(0:C_PLB_DWIDTH -1)	PLB	O	Slave read bus
SI_rdWdAddr(0:3)	PLB	O	Slave read word address
SI_rearbitrate	PLB	O	Slave rearbitrate bus indicator
SI_wait	PLB	O	Slave wait indicator
SI_wrBTerm	PLB	O	Slave terminate write burst transfer
SI_wrComp	PLB	O	Slave write transfer complete indicator
SI_wrDAck	PLB	O	Slave write data acknowledge
SI_SSize(0:1)	PLB	O	Slave data bus size
PLB_pendReq	PLB	I	PLB pending bus request indicator
PLB_pendPri(0:1)	PLB	I	PLB pending request priority
PLB_reqPri(0:1)	PLB	I	PLB current request priority
PLB_Clk	System	I	System C2 clock

Table 5: PLB BRAM Interface Controller I/O Signals (Continued)

Signal Name	Interface	I/O	Description
PLB_Rst	System	I	System PLB Reset
BRAM_Rst	IP Core	O	BRAM Reset
BRAM_Clk	IP Core	O	BRAM Clock
BRAM_EN	IP Core	O	BRAM Enable
BRAM_WEN	IP Core	O	BRAM Write Enable
BRAM_Addr(0:C_PLB_AWIDTH-1)	IP Core	O	BRAM Address
BRAM_Din(0:C_PLB_DWIDTH-1)	IP Core	I	BRAM Data Input
BRAM_Dout(0:C_PLB_DWIDTH-1)	IP Core	O	BRAM Data Output

## 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 busses. The dependencies between the BRAM design parameters and I/O signals are shown in Table 6.

Table 6: Parameter-Port Dependencies

Name	Affects	Depends	Relationship Description
Design Parameters			
C_NUM_MASTERS	SI_MBusy SI_MErr C_PLB_MID_WIDTH	0 to C_NUM_MASTERS -1 0 to C_NUM_MASTERS -1	Number of Signals required based on the number of PLB Masters required
C_PLB_DWIDTH	PLB_BE PLB_wrDBus SI_rdDBus	0 to C_PLB_DWIDTH/8 -1 0 to C_PLB_DWIDTH -1 0 to C_PLB_DWIDTH -1	Number of Byte Enables Decoded Width of the PLB write Data Bus Width of the Slave read Data Bus
C_PLB_AWDITH	PLB_ABUS	0 to C_PLB_AWIDTH -1	Width of the PLB Address Bus
C_PLB_MID_WIDTH	PLB_masterID	0 to C_PLB_MID_WIDTH-1	Size of the masterID is decoded as the function $\log_2(x)(C\_NUM\_MASTERS)$
I/O Signals			
PLB_ABus		C_PLB_AWIDTH	Width varies with the width of the PLB Address Bus
PLB_BE		C_PLB_DWIDTH	Width varies with the width of the PLB Data Bus
PLB_masterID		C_PLB_MID_WIDTH	Width varies with the number of bits required to encode the number of PLB Masters
PLB_wrDBus		C_PLB_DWIDTH	Width varies with the width of the PLB Data Bus
SI_MBusy		C_NUM_MASTERS	Width varies with the number of PLB masters
SI_MErr		C_NUM_MASTERS	Width varies with the number of PLB masters
SI_rdDBus		C_PLB_DWIDTH	Width varies with the width of the PLB Data Bus

Table 6: Parameter-Port Dependencies (Continued)

Name	Affects	Depends	Relationship Description
BRAM_Addr		C_PLB_AWIDTH	Width varies with the width of the BRAM Address Bus.
BRAM_Din		C_PLB_DWIDTH	Width varies with the width of the BRAM Data Bus.
BRAM_Dout		C_PLB_DWIDTH	Width varies with the width of the BRAM Data Bus.

## PLB Timing

This section describes the basic read and write timing and protocol for the PLB. For detailed descriptions refer to the IBM PLB Architecture Specification (v3.4). Figure 2 shows the operation of a single read data transfer on the PLB. The PLB\_PAValid signal, as well as the address and transfer qualifier signals, will continue to drive until the slave device asserts the SI\_AddrAck signal. The slave then asserts the SI\_rdComp signal in the same clock cycle as the data acknowledge phase to indicate that the transfer will complete and that the arbiter may arbitrate on the next read request.

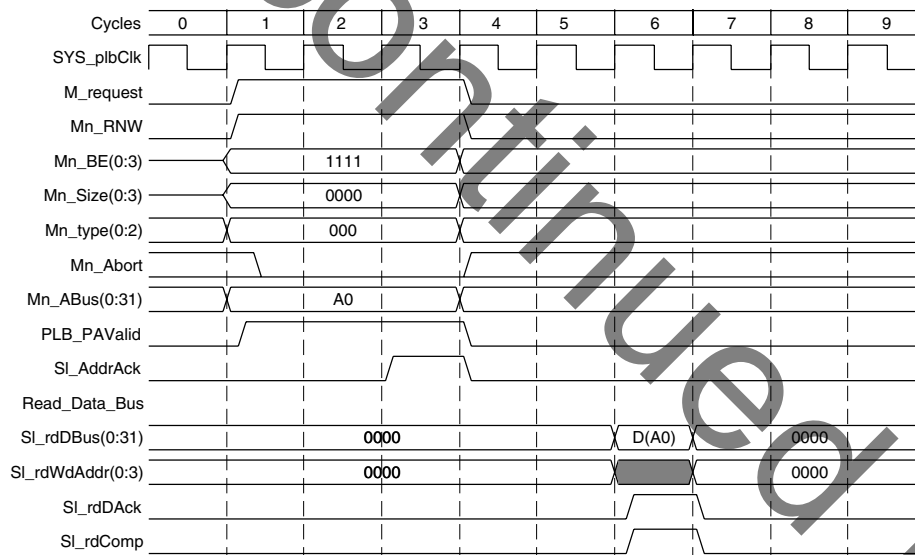


Figure 2: Single Read Transfer

Figure 3 shows the operation of a single write data transfer on the PLB. The PLB arbiter will continue to drive the PLB\_PAVValid signal as well as the address and transfer qualifier signals until the slave device asserts the SI\_AddrAck signal. The slave then asserts the SI\_wrComp and SI\_wrDack to indicate that data is valid on the bus and that the transfer is complete. Note that the write data bus must be valid at the time Mn\_request is first asserted and held until the end of the clock cycle in which SI\_wrDack signal is asserted.

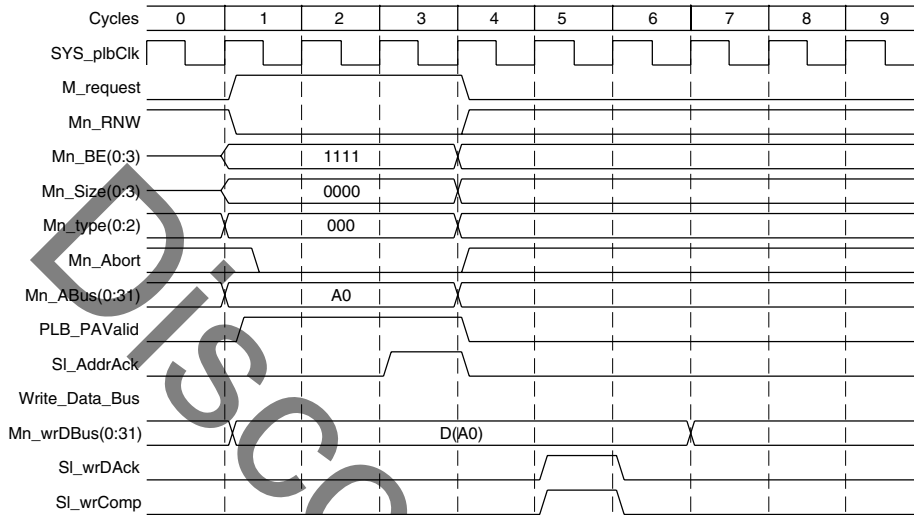


Figure 3: Single Write Transfer

Single transactions of all transaction sizes are supported by the BRAM Interface Controller. A single double-word write followed by a single double-word read is shown in Figure 4.

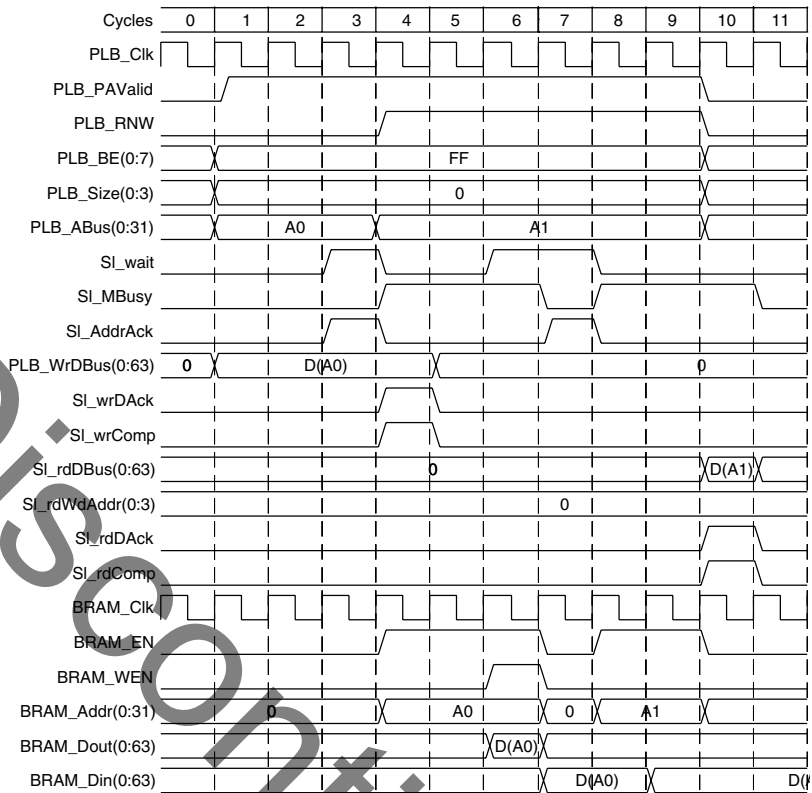


Figure 4: Single double-word write/read

Cacheline and burst transactions are supported by the BRAM Interface Controller if the parameter C\_INCLUDE\_BURST\_CACHELN\_SUPPORT=1. An 8-word cacheline write followed by an 8-word cacheline read is shown in Figure 5.

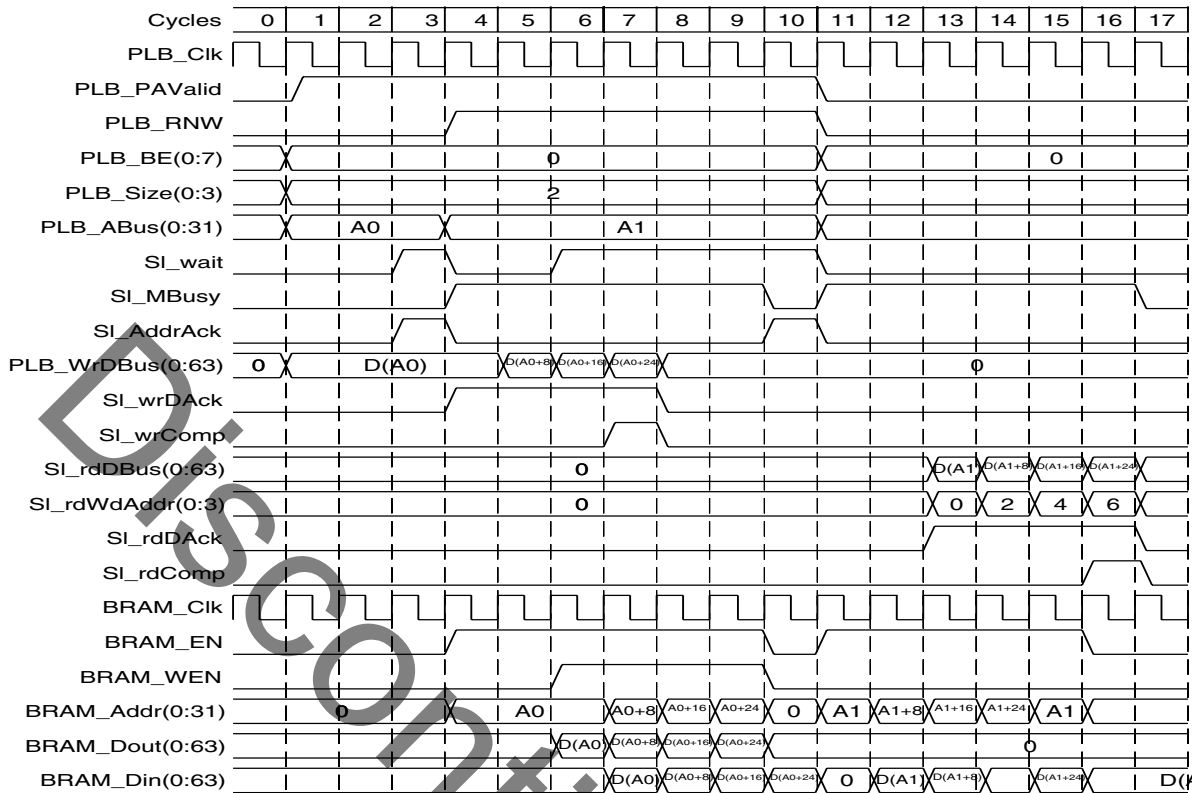


Figure 5: 8-word cacheline write/read

An 8-doubleword burst write followed by an 8-doubleword burst read is shown in **Figure 6**.

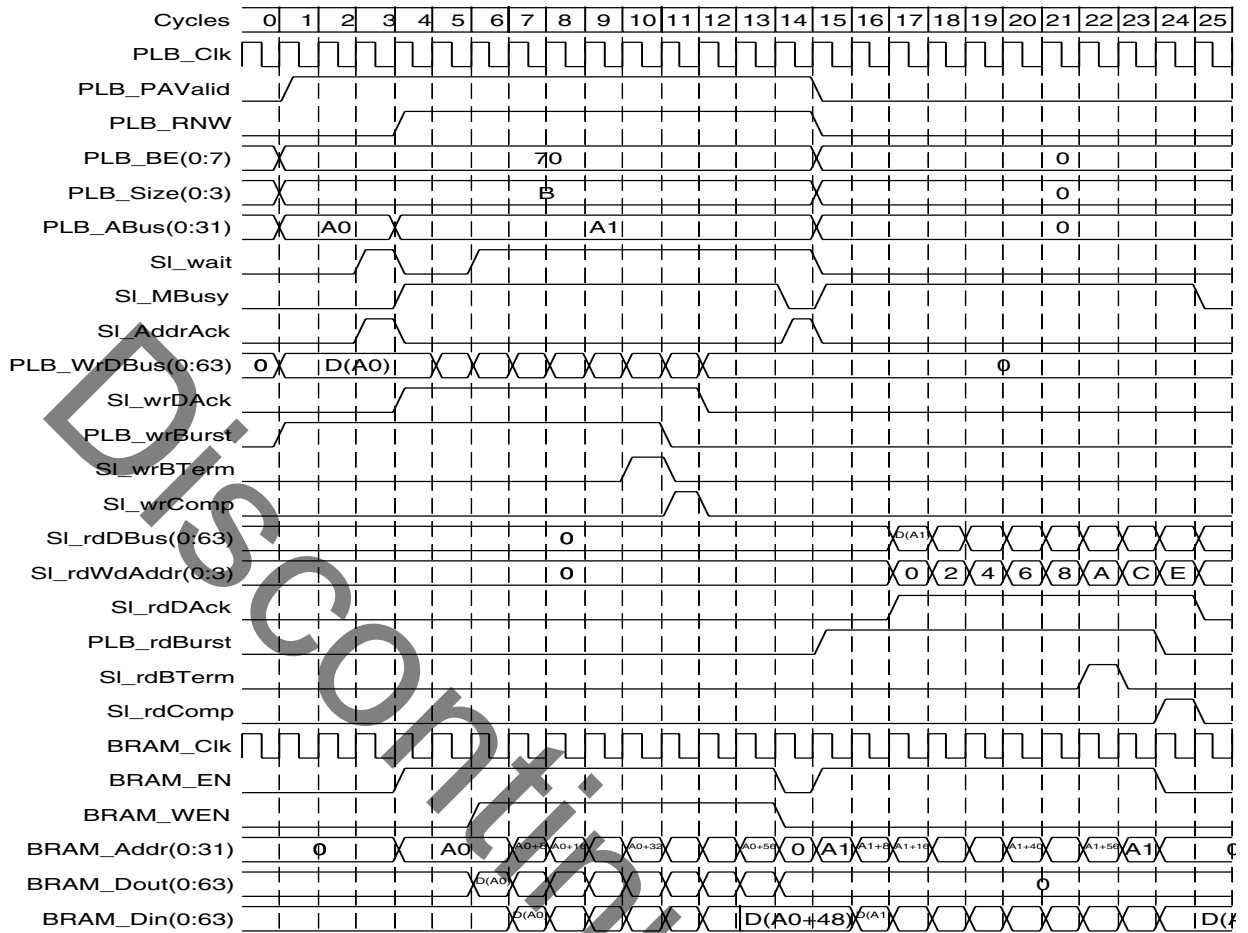


Figure 6: 8-doubleword write/read bursts

## Data Types and Organization

The BRAMs are organized as big-endian data. The bit and byte labeling for the big-endian data types is shown in **Figure 7**.

Byte address	n	n+1	n+2	...	n+7	<b>DoubleWord (PLB Only)</b>	
Byte label	0	1	2	...	7		
Byte significance	MSByte			...	LSByte		
Bit label	0						63
Bit significance	MSBit						LSBit

Byte address	n	n+1	n+2	n+3	<b>Word</b>	
Byte label	0	1	2	3		
Byte significance	MSByte			LSByte		
Bit label	0					31
Bit significance	MSBit					LSBit

Byte address	n	n+1	<b>Halfword</b>	
Byte label	0	1		
Byte significance	MSByte	LSByte		
Bit label	0			15
Bit significance	MSBit			LSBit

Byte address	n	<b>Byte</b>	
Byte label	0		
Byte significance	MSByte		
Bit label	0		7
Bit significance	MSBit		LSBit

Figure 7: Big-Endian Data Types

## Design Implementation

### Device Utilization and Performance Benchmarks

Since the PLB 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.

The PLB BRAM Interface Controller benchmarks are shown in [Table 7](#) for a Virtex™II Pro -7 FPGA.

**Table 7: PLB BRAM Interface FPGA Performance and Resource Utilization Benchmarks (Virtex™II Pro -7)**

Parameter Values	Device Resources			f <sub>MAX</sub> (MHz)
	Slices	Slice Flip- Flops	4-input LUTs	f <sub>MAX</sub>
0	139	150	182	131
1	265	261	347	116

**Notes:**

1. These benchmark designs contain the PLB BRAM IF Controller with BRAM memory blocks present in the design, but without any additional logic.. Benchmark numbers approach the performance ceiling rather than representing performance under typical user conditions.

### Target Technology

The intended target technology is a Virtex™II FPGA.

### Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/10/02	1.0	Initial release.
11/11/02	1.2	Updated supported memory sizes table
05/29/03	1.3	Added burst support and resource utilization table. Updated timing diagrams.
07/15/03	1.4	Update to new template
07/29/03	1.4.1	Change DS number because of duplication