LogiCORE IP PLBv46 to PLBv46 Bridge (v1.03a)

DS618 July 23, 2010 **Product Specification**

Introduction

The PLBv46 (Processor Local Bus Version 4.6 with Xilinx simplification) to PLBv46 Bridge allows the designer to connect two PLB buses.

The PLBv46 to PLBv46 Bridge can be used to isolate the slow PLB peripherals from the primary PLB and improve the system performance.

The PLBv46 to PLBv46 Bridge is a slave on the primary PLB and is a master on the secondary PLB.

The Xilinx PLBv46 to PLBv46 Bridge design allows customers to tailor the bridge to suit their application by setting certain parameters to enable/disable features. Parameterizable features of the design are discussed in Design Parameters section.

Features

- Supports prefetching for read operations
- Supports posted writes to the slave
- Communicates with 32, 64, and 128 bit masters on primary PLB
- Communicates with 32, 64 and 128 bit slaves on Secondary PLB
- Supports 1:1, 2:1, and 4:1 clock frequency ratios from primary PLB to secondary PLB
- Interrupt service to identify abnormal terminations
- Exclusive resets to primary PLB and secondary PLB
- Supports burst of maximum 16 data beats

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Virtex®-4, Virtex-4Q, Virtex-4QV, Virtex-5, Virtex-6, Spartan®-3E, Automotive Spartan-3E, Spartan-3, Automotive Spartan-3, Spartan-3A, Automotive Spartan-3A, Spartan-3A DSP, Automotive Spartan-3A DSP, Spartan-6
Supported User Interfaces	PLBv46
Resources	
See Table 8, Table 9, Table 10, Table 11, and Table 12.	
Provided with Core	
Documentation	Product Specification
Design Files	VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	Not Provided
Tested Design Tools	
Design Entry Tools	Platform Studio, XPS
Simulation	Mentor Graphic ModelSim v6.5c and above
Synthesis Tools	XST
Support	
Provided by Xilinx, Inc.	

Notes:
1. For a complete listing of supported devices, see the release notes for this core.

Functional Description

Overview

The PLBv46 to PLBv46 Bridge passes the PLB transactions from the primary PLB to the secondary PLB. The bridge functions as a slave on the primary PLB and as a master on the secondary PLB.

The PLB transactions received by PLBv46 Slave are decoded in the primary side of the bridge. The secondary side interface logic then generates the necessary sequence of PLB signals (including address, byte enables, and data) to perform the transaction on PLB slaves.

The PLBv46 to PLBv46 Bridge is a synchronous design, always the rising edge of the primary PLB clock signal shall be coincident with the rising edge of the secondary PLB clock signal.

The block diagram of PLBv46 to PLBv46 Bridge is shown in Figure 1 and described in the following sections.

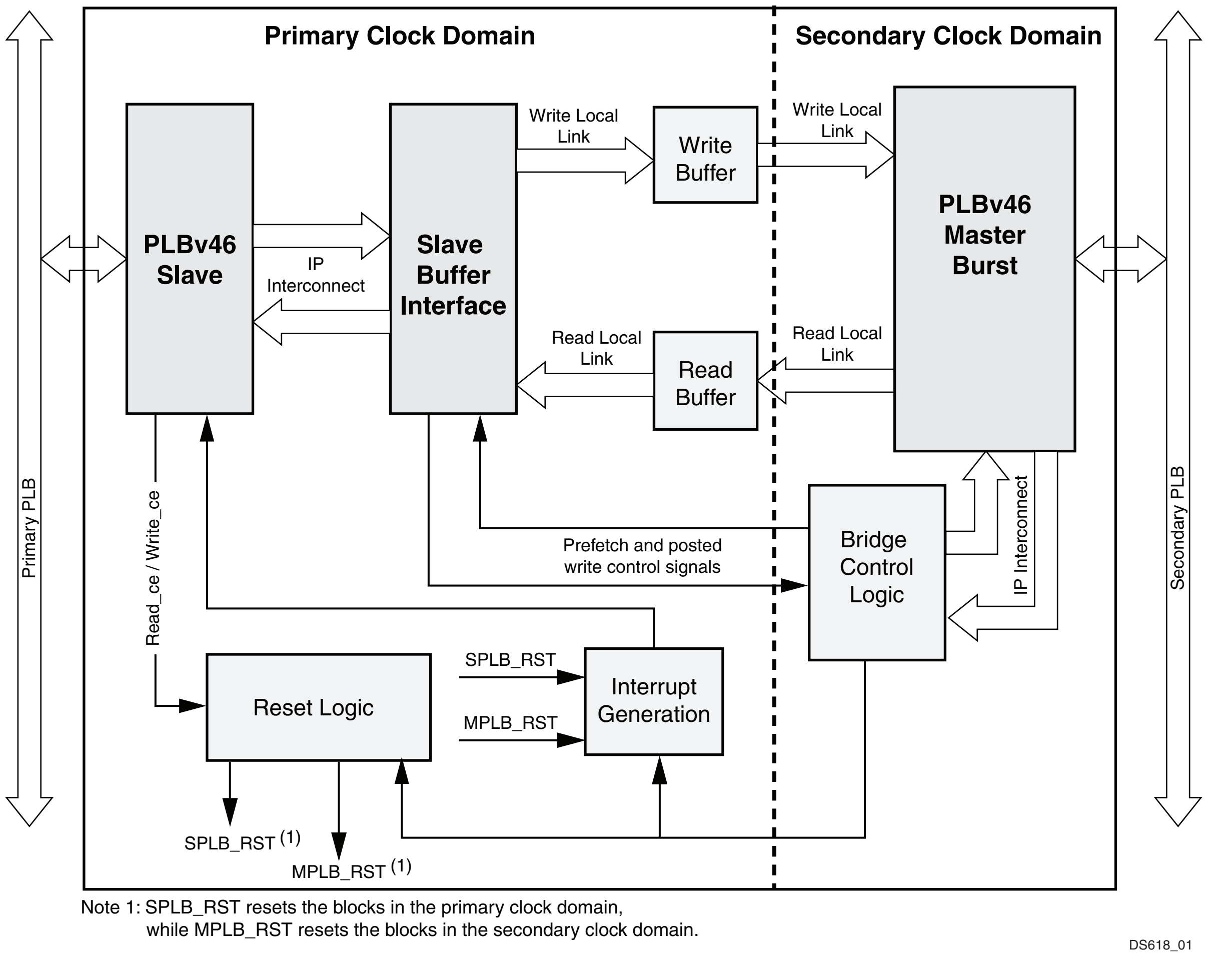

Figure 1: **Block Diagram for the PLBv46 to PLBv46 Bridge**

PLBv46 Slave

The PLBv46 Slave provides a bi-directional slave interface to the primary PLB. The primary PLB data bus width and the address bus width can be configured by setting the parameters as shown in Table 2. This module provides the following functions.

- Automatic byte steering
  - The automatic byte steering comes into picture when the primary PLB data width is different from the bridge native data width that is fixed to 32.
- Address decoding
  - Address decoding is done for the registers inside the bridge and for the slaves on the secondary PLB
- Interrupt service
  - The PLB Slave collects the interrupts from the other part of the bridge and combines in to a single interrupt that is sent to the processor

This module is an instantiated core. Refer to PLBV46_SLAVE Product Specification, DS513 for more information.

Slave Buffer Interface

The Slave Buffer Interface provides the conditional read/write access to the PLBv46 Slave. This module decodes the request from PLBv46 Slave and passes the request to the PLBv46 Master Burst through Control Logic and the data goes through the Write Buffer. This module also consists of Xilinx Local Link Interface to communicate with the read buffer and write buffer.

Write Buffer

The Write Buffer stores the data from the Slave Buffer Interface during the posted write transaction. The Write buffer is a 32-bit FIFO with maximum depth of 64. The Write Buffer communicates with the Slave Buffer Interface and the PLBv46 Master Burst using Xilinx Local Link Interface. The depth of this FIFO can be configured using the parameter C_SPLB_BIGGEST_MASTER as

DEPTH = 16 * (C_SPLB_BIGGEST_MASTER/32).

Read Buffer

The Read Buffer stores the data from the PLBv46 Master Burst during prefetch read transaction. The Read buffer is a 32-bit FIFO with maximum depth of 64. The Read Buffer communicates with the Slave Buffer Interface and the PLBv46 Master Burst using Xilinx Local Link Interface. The depth of this FIFO can be configured using the parameter C_SPLB_BIGGEST_MASTER as

DEPTH = 16 * (C_SPLB_BIGGEST_MASTER/32).

Bridge Control Logic

The Bridge Control Logic module controls the bridge and provides the read/write requests to the PLBv46 Master Burst. The Bridge Control Logic generates the status and error information to the reset and interrupt generation modules.

Reset Logic

The Reset Logic provides the resets on primary side or secondary side of the bridge independently. The inputs SPLB_Rst & MPLB_Rst and the status info from the Bridge Control Logic drives the Reset Logic to generate the resets.

There are some operations to be performed before resetting the primary or secondary side depending on the status of the transactions, as described below.

- Reset on primary
  - No transaction pending on either side - no affect (reset bridge registers)
  - Read being completed on primary - discard the data then resets the primary side

  - Prefetch read is in progress on secondary - allow read to complete prior to applying reset then discard the data
  - Posted writes into write buffer - complete writes on secondary then resets the primary side
- Reset on secondary
  - No transaction pending on either bus - no affect (resets bridge registers)
  - Prefetch read is in progress on secondary - reset the secondary and retry the read on primary
  - Posted writes - discard writes and resets secondary

Interrupt Generation

The Interrupt Generation logic generates the interrupts when the Bridge Control Logic observes the abnormal situations. The following are the scenarios that cause the interrupt.

- Secondary PLB address phase time-out
- Secondary PLB master error
- Primary PLB never reissues the read request when prefetching has been started
- Primary PLB over runs the address range of the bridge
- Primary PLB reset
- Secondary PLB reset

PLBv46 Master Burst

The PLBv46 Master Burst provides the PLB master interface on the secondary PLB. The secondary PLB data bus width can be configured by setting the parameter C_MPLB_DWIDTH. The PLBv46 Master Burst has automatic byte steering to support the bridge side data bus width that is fixed to 32-bit.

This module receives data from and transmits data to the read & write buffers via the Xilinx Local Link Interface protocol. The Xilinx Local Link Interface is a point-to-point, synchronous interface intended for high data rate applications. Because data flow is unidirectional, the PLBv46 Master Burst employs two Local Link Interfaces, one for data read operations and one for data write operations.

This module is an instantiated core. Refer to PLBV46 Master Burst specification, DS565 for more information.

I/O Signals

The PLBv46 to PLBv46 Bridge I/O signals are listed and described in Table 1.

Table 1: **Bridge I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
Primary PLB I/O Signals					
P1	SPLB_Clk	PLB	I	-	PLB clock to the primary side of the bridge
P2	SPLB_Rst	PLB	I	-	PLB reset
P3	PLB_ABus[0:(C_SPLB_AWIDTH - 1)]	PLB	I	-	PLB address bus
P4	PLB_PAValid	PLB	I	-	PLB primary address valid indicator
P5	PLB_SAValid	PLB	I	-	PLB secondary address valid indicator
P6	PLB_masterID[0:C_SPLB_MID_WIDTH - 1]	PLB	I	-	PLB current master identifier

Table 1: **Bridge I/O Signals** ***(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P7	PLB_RNW	PLB	I	-	PLB read not write
P8	PLB_BE[0:(C_SPLB_DWIDTH/8) - 1]	PLB	I	-	PLB byte enables
P9	PLB_MSize[0:1]	PLB	I	-	PLB master data bus size
P10	PLB_size[0:3]	PLB	I	-	PLB transfer size
P11	PLB_type[0:2]	PLB	I	-	PLB transfer type
P12	PLB_wrDBus[0: (C_SPLB_DWIDTH - 1)]	PLB	I	-	PLB Write Data Bus
P13	PLB_wrBurst	PLB	I	-	PLB burst write transfer indicator
P14	PLB_rdBurst	PLB	I	-	PLB burst read transfer indicator
P15	Sl_addrAck	PLB	O	0	Slave address acknowledge
P16	Sl_SSize[0:1]	PLB	O	0	Slave data bus size
P17	Sl_wait	PLB	O	0	Slave wait indicator
P18	Sl_rearbitrate	PLB	O	0	Slave rearbitrate bus indicator
P19	Sl_wrDAck	PLB	O	0	Slave read data acknowledge
P20	Sl_wrComp	PLB	O	0	Slave write transfer complete indicator
P21	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P22	Sl_rdDBus[0:(C_SPLB_ DWIDTH - 1)]	PLB	O	0	Slave read data bus
P23	Sl_rdWdAddr[0:3]	PLB	O	0	Slave read word address
P24	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
P25	Sl_rdComp	PLB	O	0	Slave read transfer complete indicator
P26	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
P27	Sl_MBusy[0:(C_SPLB_NUM_MASTERS - 1)]	PLB	O	0	Slave busy indicator
P28	Sl_MWrErr[0:(C_SPLB_NUM_MASTERS - 1)]	PLB	O	0	Slave write error indicator
P29	Sl_MRdErr[0:(C_SPLB_NUM_MASTERS - 1)]	PLB	O	0	Slave read error indicator
P30	Sl_MIRQ	PLB	O	0	Unused
Primary PLB Unused I/O Signals					
P31	PLB_UABus[0:(C_SPLB_AWIDTH - 1)]	PLB	I	-	Unused
P32	PLB_rdPrim	PLB	I	-	Unused
P33	PLB_wrPrim	PLB	I	-	Unused
P34	PLB_busLock	PLB	I	-	Unused
P35	PLB_lockErr	PLB	I	-	Unused
P36	PLB_wrPendReq	PLB	I	-	Unused
P37	PLB_rdPendReq	PLB	I	-	Unused
P38	PLB_wrPendPri[0:1]	PLB	I	-	Unused
P39	PLB_rdPendPri[0:1]	PLB	I	-	Unused
P40	PLB_reqPri[0:1]	PLB	I	-	Unused

Table 1: **Bridge I/O Signals *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P41	PLB_TAttribute[0:15]	PLB	I	-	Unused
P42	PLB_abort	PLB	I	-	PLB abort bus request indicator
Secondary PLB I/O Signals					
P43	MPLB_Clk	PLB	I	-	PLB clock to the secondary side of the bridge
P44	MPLB_Rst	PLB	I	-	PLB reset
P45	MD_Error	PLB	O	0	Master error detection indicator
P46	M_request	PLB	O	0	Bus request the arbiter
P47	M_priority[0:1]	PLB	O	0	Bus request priority
P48	M_buslock	PLB	O	0	Bus lock request
P49	M_RNW	PLB	O	0	PLB read not write
P50	M_BE[0: (C_MPLB_DWIDTH - 1/8) - 1]	PLB	O	0	Master byte enables
P51	M_Msize[0:1]	PLB	O	0	Master data bus size
P52	M_size[0:3]	PLB	O	0	Master transfer size
P53	M_type[0:2]	PLB	O	0	Master transfer type
P54	M_ABus[0:(C_MPLB_ AWIDTH - 1)]	PLB	O	0	Master address bus
P55	M_wrBurst	PLB	O	0	Master burst write transfer indicator
P56	M_rdBurst	PLB	O	0	Master read write transfer indicator
P57	M_WrDBus[0:(C_MPLB_DWIDTH - 1)]	PLB	O	0	Master write data bus
P58	PLB_MaddrAck	PLB	I	-	PLB master address acknowledge
P59	PLB_MSSize[0:1]	PLB	I	-	PLB slave data bus size
P60	PLB_Mrearbitrate	PLB	I	-	PLB master bus rearbitrate indicator
P61	PLB_MTimeout	PLB	I	-	PLB master bus time out
P62	PLB_MRdErr	PLB	I	-	PLB master slave read error indicator
P63	PLB_MWrErr	PLB	I	-	PLB master slave write error indicator
P64	PLB_MRdDBus[0:(C_MPLB_DWIDTH - 1)]	PLB	I	-	PLB master read data bus
P65	PLB_MRdDAck	PLB	I	-	PLB master read data acknowledge
P66	PLB_RdBTerm	PLB	I	-	PLB master terminate read burst indicator
P67	PLB_MWrDAck	PLB	I	-	PLB master write data acknowledge
P68	PLB_MWrBTerm	PLB	I	-	PLB master terminate write burst indicator
Secondary PLB Unused I/O Signals					
P69	M_TAttribute[0:15]	PLB	O	0	Unused
P70	M_lockErr	PLB	O	0	Unused
P71	M_abort	PLB	O	0	Unused

Table 1: **Bridge I/O Signals** ***(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P72	M_UABus[0:(C_MPLB_DWIDTH - 1)]	PLB	O	0	Unused
P73	PLB_MBusy	PLB	I	-	Unused
P74	PLB_MIRQ	PLB	I	-	Unused
P75	PLB_MRdWdAddr[0:3]	PLB	I	-	Unused

Design Parameters

To allow the user to create a PLBv46 to PLBv46 Bridge that is uniquely tailored for the user's system, certain features can be parameterized. This allows the user to have a design that utilizes only the resources required by the system and runs at the best possible performance. The features that can be parameterized in the PLBv46 to PLBv46 Bridge core are as shown in Table 2.

Table 2: **Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System					
G1	Target device family	C_FAMILY	spartan3a, aspartan3a, spartan3, aspartan3, spartan3e, aspartan3e, spartan3adsp, aspartan3adsp, virtex4, qvirtex4, qrvirtex4, virtex5	virtex5	string
Bridge Features					
G2	Native data width	C_SPLB_NATIVE_ DWIDTH	32	32	integer
G3	Native data width	C_MPLB_NATIVE_ DWIDTH	32	32	integer
G4	The ratio of primary clock to the secondary clock	C_BUS_CLOCK_ RATIO	1, 2, 4	1	integer
G5	Prefetch timeout counter size (bits)	C_PREFETCH_ TIMEOUT	2 - 32	10	integer
PLB Slave (primary side) Interface					
G6	PLB base Address range for the bridge registers	C_BRIDGE_ BASEADDR	Valid address(1)	See note (4)	std_logic _vector
G7	PLB high Address range for the bridge registers	C_BRIDGE_ HIGHADDR	Valid address(1)	See note (4)	std_logic _vector
G8	Number of PLB address ranges	C_NUM_ADDR_RNG	1 - 4(2)	1	integer
G9	PLB base address for address range 1	C_RNG0_BASEADDR	Valid address(2)	See note (4)	std_logic _vector
G10	PLB high address for address range 1	C_RNG0_HIGHADDR	Valid address(2)	See note (4)	std_logic _vector

Table 2: **Design Parameters *(Cont'd)***

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G11	PLB base address for address range 2	C_RNG1_BASEADDR	Valid address(2)	See note (4)	std_logic_vector
G12	PLB high address for address range 2	C_RNG1_HIGHADDR	Valid address(2)	See note (4)	std_logic_vector
G13	PLB base address for address range 3	C_RNG2_BASEADDR	Valid address(2)	See note (4)	std_logic_vector
G14	PLB high address for address range 3	C_RNG2_HIGHADDR	Valid address(2)	See note (4)	std_logic_vector
G15	PLB base address for address range 4	C_RNG3_BASEADDR	Valid address(2)	See note (4)	std_logic_vector
G16	PLB high address for address range 4	C_RNG3_HIGHADDR	Valid address(2)	See note (4)	std_logic_vector
G17	Selects point-to-point or shared PLB topology	C_SPLB_P2P	0 = Shared bus topology 1 = Reserved(3)	0	integer
G18	PLB master ID width	C_SPLB_MID_WIDTH	$\log_2$ (C_SPLB_NUM_MASTERS). The minimum value is 1(5)	1	integer
G19	Number of PLB masters	C_SPLB_NUM_MASTERS	1 - 16	1	integer
G20	Width of the smallest master that will be interacting with the bridge	C_SPLB_SMALLEST_MASTER	32, 64, 128	32	integer
G21	Width of the biggest master that will be interacting with the bridge	C_SPLB_BIGGEST_MASTER	32, 64, 128	32	integer
G22	Address bus width of primary PLB	C_SPLB_AWIDTH	32	32	integer
G23	Data width of the primary PLB	C_SPLB_DWIDTH	32, 64, 128	32	integer
PLB Master (secondary side) Interface					
G24	Address bus width of secondary PLB	C_MPLB_AWIDTH	32	32	integer
G25	Data width of the secondary PLB	C_MPLB_DWIDTH	32, 64, 128	32	integer
G26	Width of the smallest slave that will be interacting with the bridge	C_MPLB_SMALLEST_SLAVE	32, 64, 128	32	integer

1. The range specified by C_BRIDGE_BASEADDR and C_BRIDGE_HIGHADDR must comprise a complete, contiguous power of two range such that range = 2^n, and the n least significant bits of C_BRIDGE_BASEADDR must be zero.
2. Four sets of address ranges can be specified for the bridge. The number of address ranges needed is set in the parameter C_NUM_ADDR_RNG. The range specified by the various base addresses and corresponding high addresses must comprise a complete, contiguous power of two range such that range = 2^n, and the n least significant bits of the base address must be zero. If an address range needs to support 16 word cacheline transactions, the base address for this address range must be aligned to a 64-byte address.
3. Currently point to point bus topology is not supported.
4. The user must set this value
5. Indicates the number of bits in the PLB_MasterID

Port Dependencies

The dependencies between the PLBv46 to PLBv46 Bridge core design parameters and I/O signals are described in Table 3. When certain features are parameterized out of the design, the related logic will no longer be a part of the design. The unused input signals are unconnected and related output signals are set to a constant value.

Table 3: **Parameter - Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G18	C_SPLB_MID_WIDTH	P9	G19	Affects the width of current master identifier signals and depends on $\log_2$(C_SPLB_NUM_MASTERS) with a minimum value of 1
G19	C_SPLB_NUM_MASTERS	P39, P40, P41	-	Affects the width of busy and error signals.
G22	C_SPLB_AWIDTH	P3	-	Affects number of bits in address bus
G23	C_SPLB_DWIDTH	P13, P18, P34	-	Affects number of bits in data buses and the byte enables
G24	C_MPLB_AWIDTH	P54	-	Affects number of bits in master address bus
G25	C_MPLB_DWIDTH	P50, P57, P64	-	Affects number of bits in master data bus
I/O Signals				
P3	PLB_ABus	-	G22	Width varies with the size of the PLB master address bus
P6	PLB_masterID	-	G18	Width varies with the size of the PLB number of masters
P9	PLB_BE	-	G23	Width varies with the size of the PLB data bus
P13	PLB_wrDBus	-	G23	Width varies with the size of the PLB data bus
P23	Sl_rdDBus	-	G23	Width varies with the size of the PLB data bus
P28	Sl_MBusy	-	G19	Width varies with the number of PLB masters
P29	Sl_MWrErr	-	G19	Width varies with the number of PLB masters
P30	Sl_MRdErr	-	G19	Width varies with the number of PLB masters
P50	M_BE	-	G25	Width varies with the size of the PLB master data bus
P54	M_ABus	-	G24	Width varies with the size of the PLB master address bus
P57	M_WrDBus	-	P64	Width varies with the size of the PLB master data bus
P64	PLB_MRdDBus	-	P64	Width varies with the size of the PLB master data bus

Register Descriptions

The Table 4 shows the PLBv46 to PLBv46 Bridge Registers and their addresses. The remaining unused addresses in the address-range size are reserved. These points should be considered when reading or writing the registers.

- Writing into the reserved registers has no effect.
- Reading of the reserved registers returns zero.
- Each register is addressable on a 32-bit boundary.
- All registers are defined for 32-bit access only.
- The register addresses are offset to the base address, C_BRIDGE_BASEADDR.

The detailed information about these registers is provided in the following section.

Table 4: **Registers**

Register Name	Base Address + Offset (hex)	Default Value (hex)	Access
Interrupt Status Register (ISR)	C_BRIDGE_BASEADDR + 20	0x0	R/TOW[1]
Global Interrupt Enable Register (GIER)	C_BRIDGE_BASEADDR + 1C	0x0	R/W
Interrupt Enable Register (IER)	C_BRIDGE_BASEADDR + 28	0x0	R/W

Notes:

1. TOW =Toggle On Write. Writing a '1' to a bit position within the register causes the corresponding bit position in the register to toggle.

Interrupt Status Register

When read, the content of Interrupt Status Register indicates the presence or absence of an active interrupt. Each bit in the register that is set to '1' indicates an active interrupt on the corresponding interrupt input. Bits that are '0' are not active. The bits in the ISR are independent of the interrupt enable bits in the IER. The bits in the IER do not stop the interrupts getting captured into the ISR.

The Interrupt Status Register is shown in Figure 2 and the bit definitions are described in Table 5. This register resides in the PLB Slave module, refer PLBV46_SLAVE Product Specification, DS513 for more information.

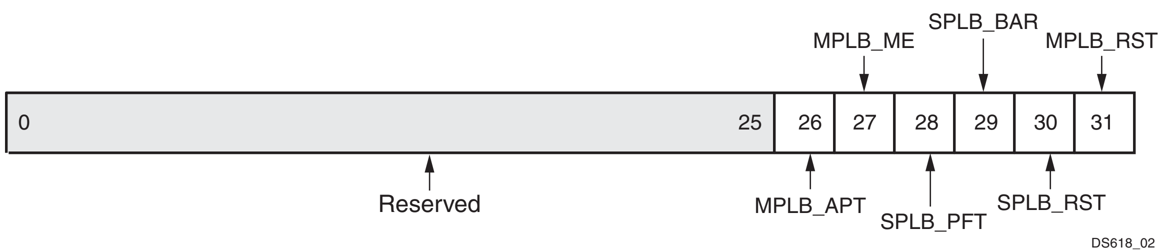

Figure 2: **Interrupt Status Register**

Table 5: **Interrupt Status Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 25	-			Reserved
26	MPLB_APT	R/W	'0'	Secondary PLB address phase time-out
27	MPLB_ME	R/W	'0'	Secondary PLB master error
28	SPLB_PFT	R/W	'0'	Primary PLB never reissues the read request when prefetching has been started
29	SPLB_BAR	R/W	'0'	Primary PLB over runs the address range of the bridge
30	SPLB_RST	R/W	'0'	Interrupt generated when reset is applied to primary PLB
31	MPLB_RST	R/W	'0'	Interrupt generated when reset is applied to secondary PLB

Global Interrupt Enable Register

The Global Interrupt Enable Register is shown in Figure 3 and the bit definitions are described in Table 6. The Global Interrupt Enable Register has a single defined bit, in the high-order position, that is used to globally enable the final interrupt output from the PLBv46 to PLBv46 Bridge to the IP2INTC_Irpt output port.

If interrupts are globally disabled, the GIE it is set to '0', there will be no interrupt from the PLBv46 to PLBv46 Bridge under any circumstances. Otherwise, there may be interrupts generated for interrupt sources that are active Interrupt Status Register and not disabled in the Interrupt Enable Register. This register resides in PLB Slave module, refer PLBV46_SLAVE Product Specification, DS513 for more information.

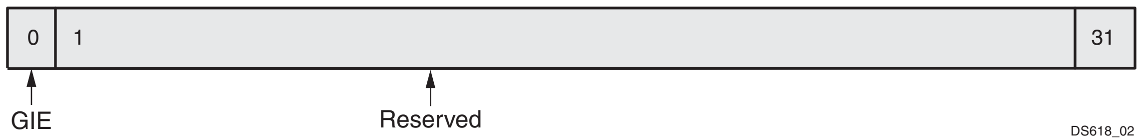

Figure 3: **Global Interrupt Enable Register**

Table 6: Global Interrupt Enable Register Bit Definitions

Bit(s)	Name	Core Access	Reset Value	Description
0	GIE	R/W	'0'	Global interrupt enable
1 - 31		-		Reserved

Interrupt Enable Register

Writing '1' in a bit location in the Interrupt Enable Register enables the corresponding ISR bit to cause assertion of interrupt output. An IER bit set to '0' does not inhibit an interrupt from being captured into the ISR.

The IER is shown in Figure 4 and the bit definitions are described in Table 7. The Interrupt Enable Register has an enable bit for each defined bit of the Interrupt Status Register. Writing a '1' to a bit in this register enables the corresponding ISR bit to cause assertion of the interrupt output. An IER bit set to '0' does not inhibit an interrupt condition from being captured. This register resides in PLB Slave, module, refer PLBV46_SLAVE Product Specification, DS513 for more information.

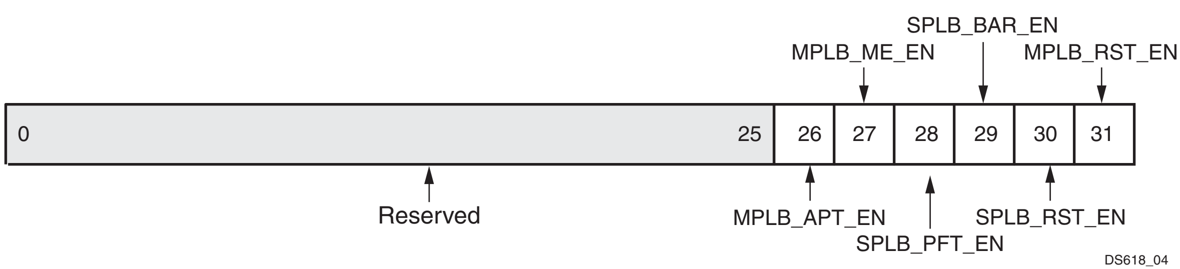

Figure 4: **Interrupt Enable Register**

Table 7: **Interrupt Enable Register Bits Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 25		-		Reserved
26	MPLB_APT_EN	R/W	'0'	Interrupt MPLB_APT enable
27	MPLB_ME_EN	R/W	'0'	Interrupt MPLB_ME enable

Table 7: **Interrupt Enable Register Bits Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
28	SPLB_PFT_EN	R/W	'0'	Interrupt SPLB_PFT enable
29	SPLB_BAR_EN	R/W	'0'	Interrupt SPLB_BAR enable
30	SPLB_RST_EN	R/W	'0'	Interrupt SPLB_RST enable
31	MPLB_RST_EN	R/W	'0'	Interrupt MPLB_RST enable

Clock Scheme

The PLBv46 to PLBv46 Bridge supports 1:1, 2:1, and 4:1 clock ratios from primary PLB to secondary PLB. The PLBv46 to PLBv46 Bridge implementation requires the primary PLB clock and secondary PLB clock to be synchronous. It is recommended to generate both the clocks by using the same DCM. This insures that the rising edges of both the clocks are aligned and that is necessary to constrain the signals that cross the clock boundary. The Clock scheme for clock ratios 1:1, 2:1, and 4:1 are illustrated in Figure 5 and Figure 6 respectively.

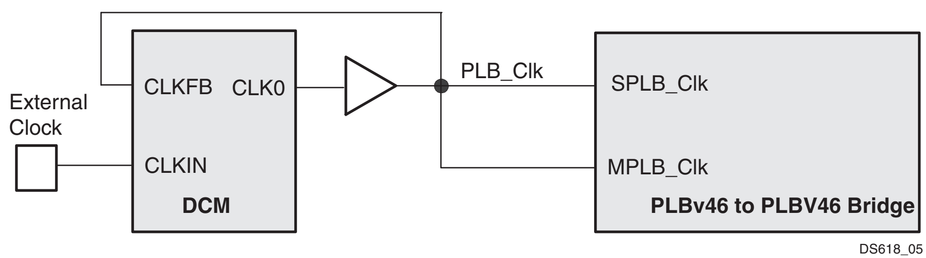

Figure 5: **Clock Scheme for clock ratio 1:1 from primary PLB to Secondary PLB**

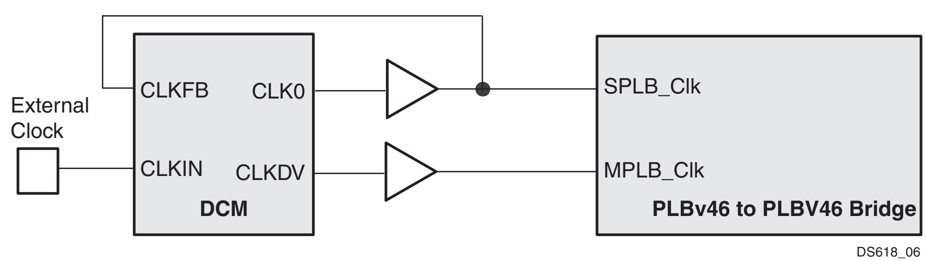

Figure 6: **Clock Scheme for clock ratios 2:1 and 4:1 from primary PLB to Secondary PLB**

Timing Diagrams

The following timing diagrams illustrates the PLBv46 to PLBv46 Bridge operation for various write and read transactions of different lengths.

Figure 7 shows the PLB single write transaction.

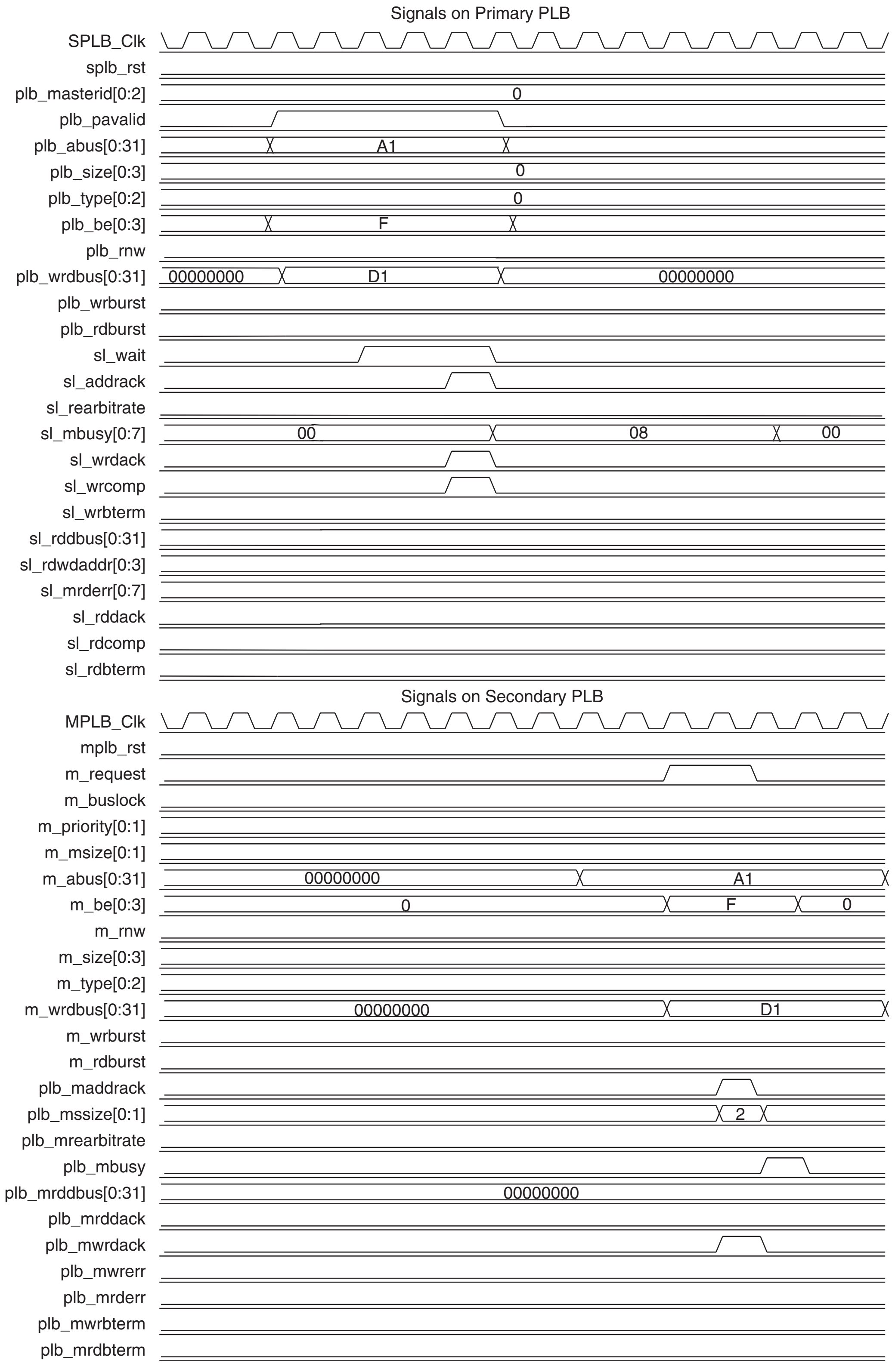

Figure 7: **PLB Single Write Transaction**

Figure 8 shows the PLB single read transaction.

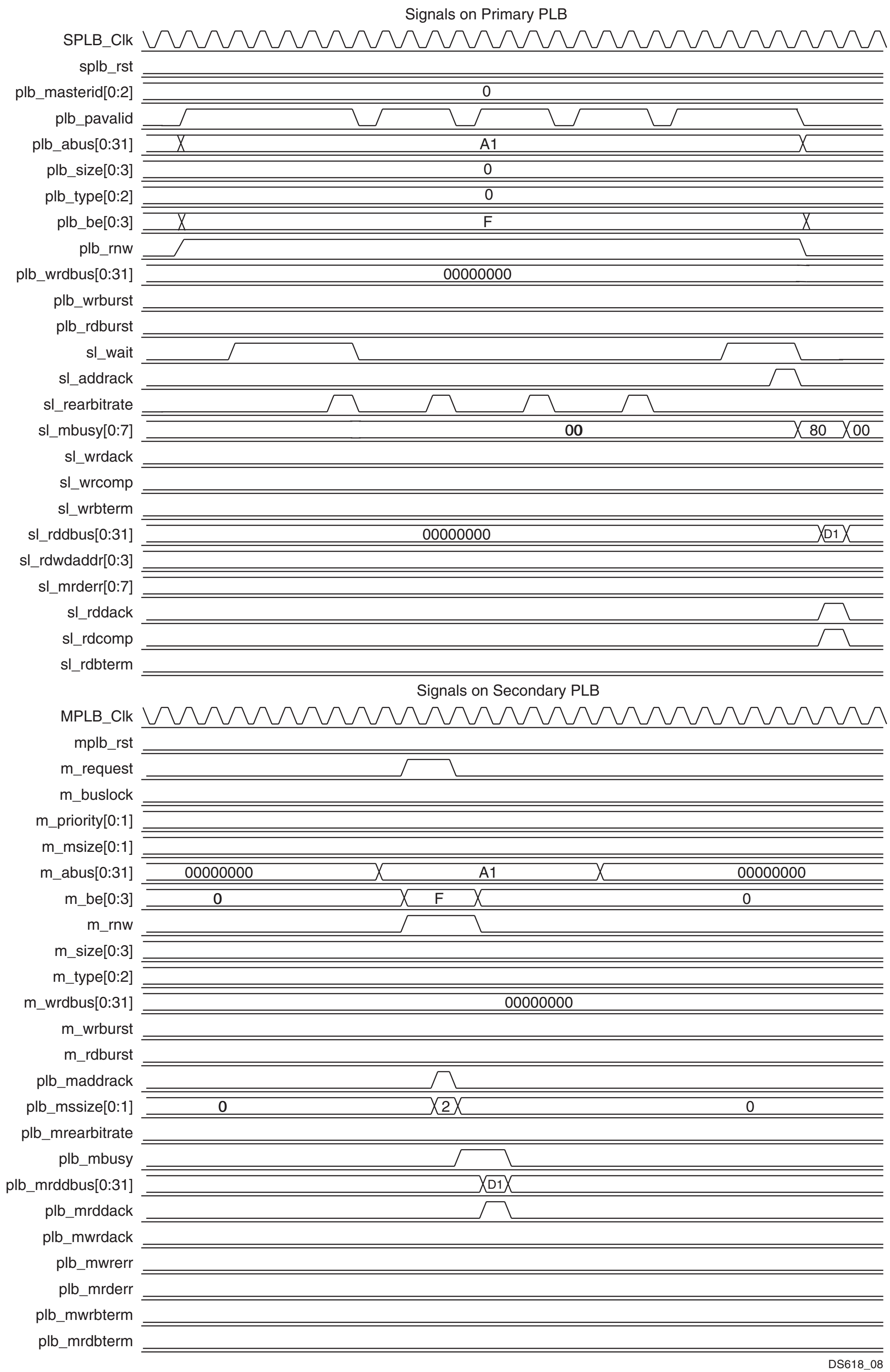

Figure 8: **PLB Single Read Transaction**

Figure 9 shows the PLB burst write transaction.

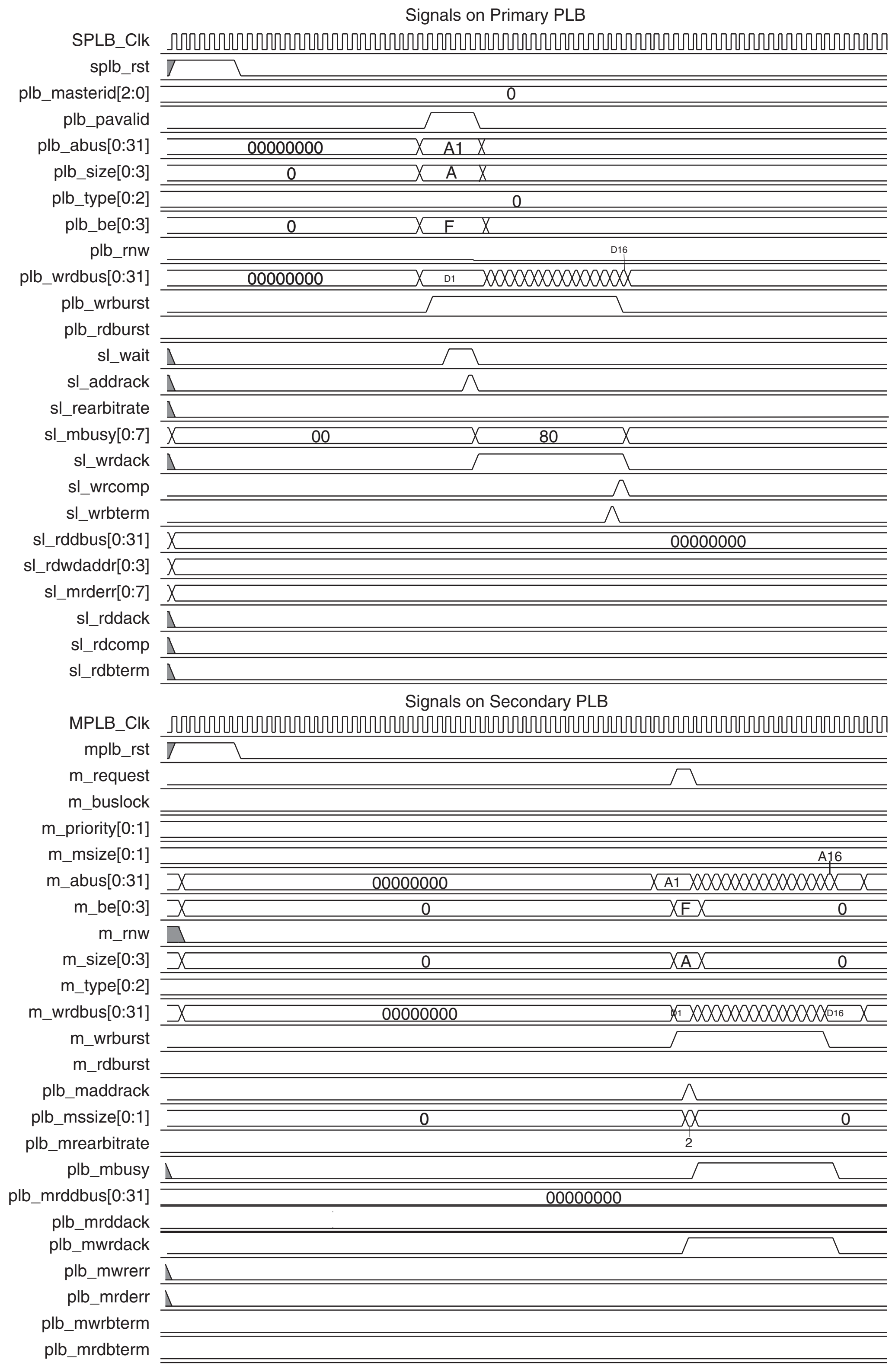

Figure 9: **PLB Burst Write Transaction**

Figure 10 shows the PLB burst read transaction.

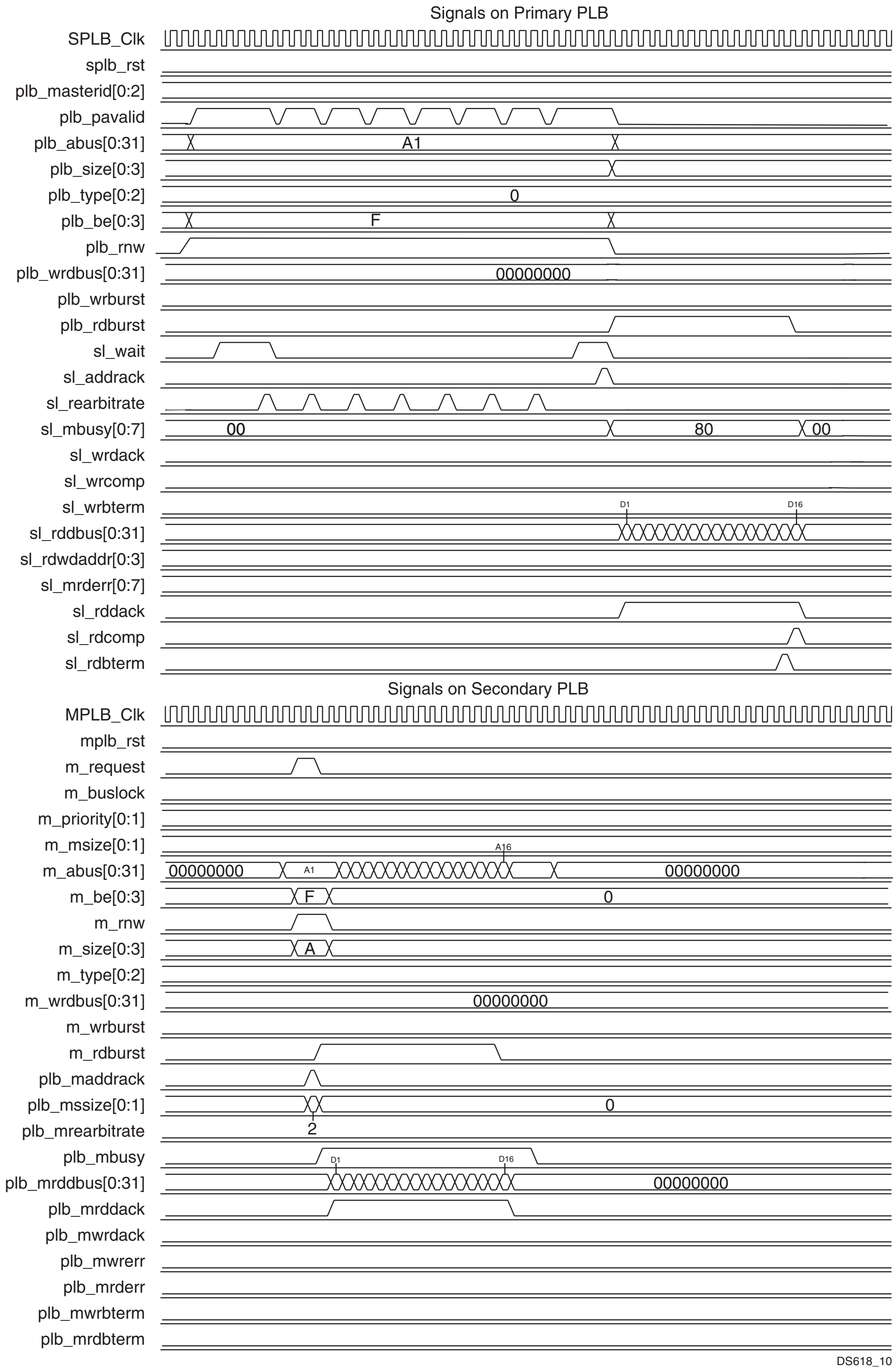

Figure 10: **PLB Burst Read Transaction**

Design Implementation

Target Technology

The intended target technology is an FPGA listed in the Supported Device Family field of the LogiCORE IP Facts Table.

Device Utilization and Performance Benchmarks

Because the PLBv46 to PLBv46 Bridge will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When thos core is combined with other designs in the system, the utilization of FPGA resources and timing of the core design will vary from the results reported here.

The PLBv46 to PLBv46 Bridge resource utilization for various parameter combinations measured with the Virtex-4 FPGA as the target device are detailed in Table 8.

Table 8: **PLBv46 to PLBv46 Bridge Performance and Resource Utilization for the Virtex-4 FPGA (xc4vlx40-ff1148-10)**

Parameter Values									Device Resources			Performance
C_NUM_ADDR_RNG	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_BIGGEST_MASTER	C_SPLB_DWIDTH	C_MPLB_DWIDTH	C_MPLB_SMALLEST_SLAVE	C_BUS_CLOCK_RATIO	C_PREFETCH_TIMEOUT	Slices	Slice Flip - Flops	LUTs	F_{Max} (MHz)
1	1	32	32	32	32	32	1	3	963	849	920	130
2	4	32	32	32	32	32	1	10	1000	867	930	127
3	8	64	64	32	32	64	1	10	1073	926	1042	127
4	8	64	128	64	64	128	2	10	1212	984	1207	127
4	8	32	32	128	128	32	4	20	1161	1018	1047	126
4	8	64	128	64	128	64	4	20	1213	1005	1239	126
4	8	128	128	128	64	64	4	20	1268	1022	1258	128
4	8	128	128	128	128	128	4	32	1264	1021	1222	128

The PLBv46 to PLBv46 Bridge resource utilization for various parameter combinations measured with the Virtex-5 FPGA as the target device are detailed in Table 9.

Table 9: **PLBv46 to PLBv46 Bridge Performance and Resource Utilization for the Virtex-5 FPGA (xc5vlx85-ff1153-1)**

Parameter Values									Device Resources		Performance
C_NUM_ADDR_RNG	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_BIGGEST_MASTER	C_SPLB_DWIDTH	C_MPLB_DWIDTH	C_MPLB_SMALLEST_SLAVE	C_BUS_CLOCK_RATIO	C_PREFETCH_TIMEOUT	Slice Flip - Flops	LUTs	F_{Max} (MHz)
1	1	32	32	32	32	32	1	3	845	725	155
2	4	32	32	32	32	32	1	10	869	732	156
3	8	64	64	32	32	64	1	10	893	769	158
4	8	64	128	64	64	128	2	10	984	919	155
4	8	32	32	128	128	32	4	20	1014	881	156
4	8	64	128	64	128	64	4	20	1004	929	157
4	8	128	128	128	64	64	4	20	1018	956	157
4	8	128	128	128	128	128	4	32	1024	996	158

The PLBv46 to PLBv46 Bridge resource utilization for various parameter combinations measured with the Spartan-3 FPGA as the target device are detailed in Table 10.

Table 10: **PLBv46 to PLBv46 Bridge Performance and Resource Utilization for the Spartan-3 FPGA (xc3s4000-fg1156-5)**

Parameter Values									Device Resources			Performance
C_NUM_ADDR_RNG	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_BIGGEST_MASTER	C_SPLB_DWIDTH	C_MPLB_DWIDTH	C_MPLB_SMALLEST_SLAVE	C_BUS_CLOCK_RATIO	C_PREFETCH_TIMEOUT	Slices	Slice Flip - Flops	LUTs	F_{Max} (MHz)
1	1	32	32	32	32	32	1	3	958	851	869	103
2	4	32	32	32	32	32	1	10	1028	874	866	101
3	8	64	64	32	32	64	1	10	1120	929	1010	103

Table 10: **PLBv46 to PLBv46 Bridge Performance and Resource Utilization for the Spartan-3 FPGA (xc3s4000-fg1156-5)**

4	8	64	128	64	64	128	2	10	1382	990	1216	129
4	8	32	32	128	128	32	4	20	1319	1018	1040	129
4	8	64	128	64	128	64	4	20	1401	1009	1221	129
4	8	128	128	128	64	64	4	20	1420	1022	1240	129
4	8	128	128	128	128	128	4	32	1417	1016	1226	125

The PLBv46 to PLBv46 Bridge resource utilization for various parameter combinations measured with the Spartan-6 FPGA as the target device are detailed in Table 11.

Table 11: **PLBv46 to PLBv46 Bridge Performance and Resource Utilization for the Spartan-6 FPGA (xc6slxt100-2-fgg676)**

Parameter Values									Device Resources		Performance
C_NUM_ADDR_RNG	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_BIGGEST_MASTER	C_SPLB_DWIDTH	C_MPLB_DWIDTH	C_MPLB_SMALLEST_SLAVE	C_BUS_CLOCK_RATIO	C_PREFETCH_TIMEOUT	Slice Flip - Flops	LUTs	F_{Max} (MHz)
1	1	32	32	32	32	32	1	3	849	901	101
2	4	32	32	32	32	32	1	10	873	954	100
3	8	64	64	32	32	64	1	10	898	1007	101
4	8	64	128	64	64	128	2	10	957	890	200
4	8	32	32	128	128	32	4	20	987	829	201
4	8	64	128	64	128	64	4	20	977	898	201
4	8	128	128	128	64	64	4	20	991	845	201
4	8	128	128	128	128	128	4	32	1022	994	201

The PLBv46 to PLBv46 Bridge resource utilization for various parameter combinations measured with the Virtex-6 FPGA as the target device are detailed in Table 12.

Table 12: **PLBv46 to PLBv46 Bridge Performance and Resource Utilization for the Virtex-6 FPGA (xc6vlx195t-1-ff1156)**

Parameter Values									Device Resources			Performance
C_NUM_ADDR_RNG	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_BIGGEST_MASTER	C_SPLB_DWIDTH	C_MPLB_DWIDTH	C_MPLB_SMALLEST_SLAVE	C_BUS_CLOCK_RATIO	C_PREFETCH_TIMEOUT	Slices	Slice Flip - Flops	LUTs	F_{Max} (MHz)
1	1	32	32	32	32	32	1	3	386	849	961	221
2	4	32	32	32	32	32	1	10	389	870	1015	209
3	8	64	64	32	32	64	1	10	425	897	1041	208
4	8	64	128	64	64	128	2	10	470	957	1150	201
4	8	32	32	128	128	32	4	20	414	987	1057	204
4	8	64	128	64	128	64	4	20	491	977	1174	211
4	8	128	128	128	64	64	4	20	482	991	1135	217
4	8	128	128	128	128	128	4	32	508	1022	1188	222

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE Design Suite Embedded Edition software under the terms of the Xilinx End User License. The core is generated using the Xilinx ISE Embedded Edition software (EDK).

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE modules and software, please contact your local Xilinx sales representative.

Reference Documents

The following documents contain reference information important to understanding the design of the PLBv46 to PLBv46 bridge:

1. DS513 PLBV46_SLAVE Product Specification
2. DS565 PLBV46 Master Burst specification

Revision History

Date	Version	Revision
08/10/07	1.0	Initial Xilinx release.
1/14/08	1.1	Added Virtex-II Pro support; modified parameters table.
4/21/08	1.2	Added Automotive Spartan-3E, Automotive Spartan-3A, Automotive Spartan-3, and Automotive Spartan-3A DSP support.
7/28/08	1.3	Added QPro Virtex-4 Hi-Rel and QPro Virtex-4 Rad Tolerant FPGA support.
9/16/09	1.4	Updated for EDK_L 11.3 release; added resource utilization tables for Spartan-6 and Virtex-6; updated images.
5/06/10	1.5	Created the document for the plbv46_plbv46_bridge_v1_03_a
7/23/10	1.6	Updated for the 12.2 release; converted to current DS template.

Notice of Disclaimer