LogiCORE IP PLBV46 to AXI Bridge (v2.00a)

DS711 June 22, 2011 **Product Specification**

Introduction

The Processor Local Bus (PLB v4.6) to AMBA® Advanced eXtensible Interface (AXI) Bridge translates PLBV46 transactions into AXI4 transactions. It functions as a slave on the PLBV46 and as a master on the AXI4. The PLBV46 to AXI Bridge main use model is to connect the AXI slaves with PLB masters.

Features

The Xilinx PLBV46 to AXI Bridge is a soft IP core with the following features:

PLBV46 Slave Interface

- Connects as a 32/64-bit slave on PLB v4.6 buses of 32, 64 or 128 bits
- Supports 1:1 (PLB:AXI) synchronous clock ratio
- Supports access by 32, 64-bit PLB masters
- Supports Xilinx simplified PLBv46 protocol
  - Single transfers of 1 to 8 bytes
  - Optional line transfers of 4 and 8 words
  - Optional Fixed length burst transfers of 2 to 16 data beats of words and double words
- Supports optional two levels of address pipelining
- Supports split bus architecture (simultaneous read and write operations)
- Supports optional PLB status/interrupt registers and generates interrupts
- Supports optional low latency PLB Point-to-Point topology
- Supports 1 to 4 address ranges with selectable cache encoding and protection unit support

AXI Master Interface

- Connects as a 32/64-bit master on 32/64-bit AXI4 interface
- Connects as a 32-bit master on 32-bit AXI4-Lite interface
- Support burst transfers of 1 to 32 words or 1 to 16 double words of INCR type and burst transfers of 4 and 8 only of WRAP type
- Supports optional generation of two outstanding addresses and supports out-of-order read transaction completion and out-of-order write transaction completion
- Supports optional limited cache encoding (cacheable/bufferable) and limited protection unit support (secure/non-secure)

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Artix-7, Virtex-7, Kintex-7[2], Virtex-6[3], Spartan-6[4]
Supported User Interfaces	PLBV46,AXI4/AXI4-Lite
Resources	
See Table 14 through Table 18.	
Provided with Core	
Documentation	Product Specification
Design Files	VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	None
Simulation Model	None
Tested Design Tools	
Design Entry Tools	XPS 13.2
Simulation	Mentor Graphics ModelSim[5]
Synthesis Tools	XST 13.2
Support	
Provided by Xilinx, Inc.	

Notes:

1. For a complete list of supported derivative devices, please see the IDS Embedded Edition Derivative Device Support.
2. For more information, see DS180, *7 Series FPGAs Overview*.
3. For more information, see DS150, *Virtex-6 Family Overview*.
4. For more information, see DS160, *Spartan-6 Family Overview*.
5. For the supported versions of the tools, see the ISE Design Suite 13: Release Notes Guide.

Functional Description

Overview

The PLBV46 to AXI Bridge translates PLB transactions into AXI transactions. The bridge functions as a slave on the PLB and as a master on the AXI.

PLBV46 to AXI Bridge block diagram is shown in Figure 1 and described in following sections.

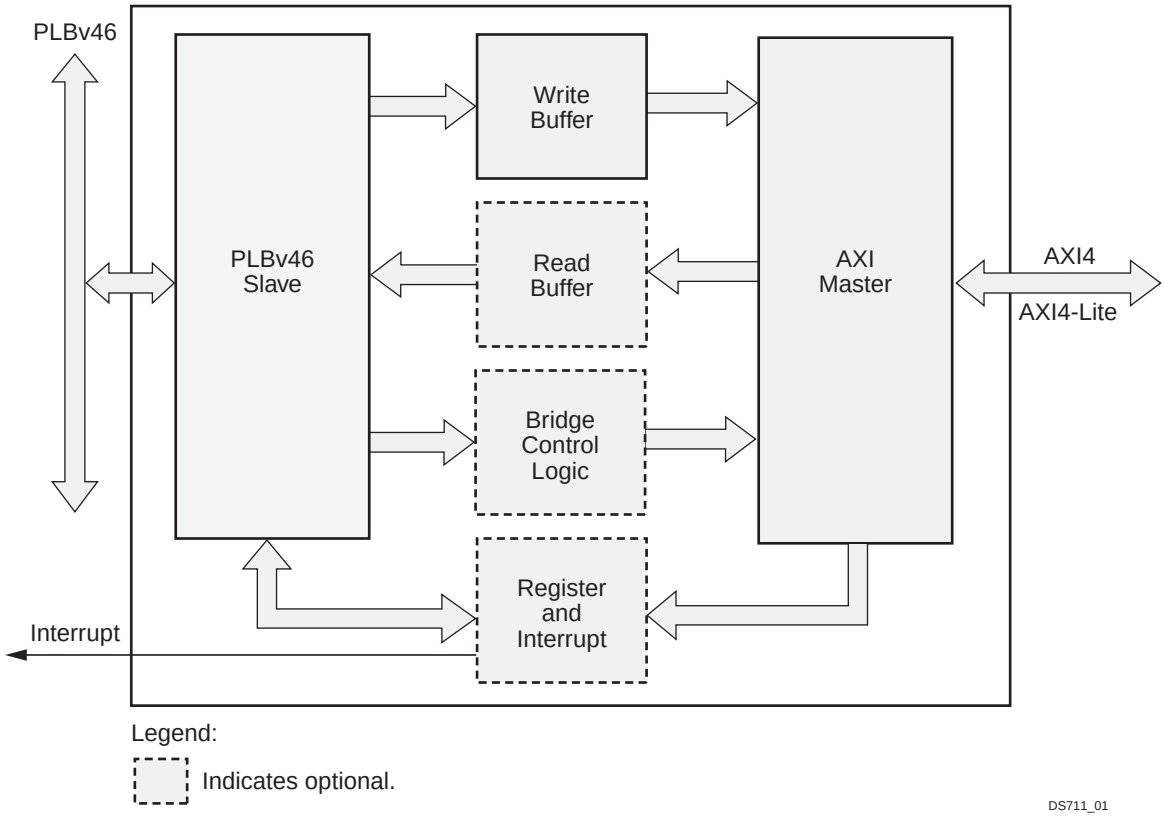

Figure 1: **PLBV46 to AXI Bridge Block Diagram**

PLBv46 Slave

The PLBv46 Slave module provides a bi-directional slave interface to the PLB. The PLB data bus width can be configured by setting the parameters as shown in Table 2. This module decodes the address for the bridge registers and for the slaves on the AXI when C_SPLB_P2P = 0. This module also implements the logic to detect if overlapping write and read requests are issued from PLB. As AXI has independent read and write channels, these requests are issued in such a way that the data coherency is maintained.

Write Buffer

The Write Buffer stores the write data from the PLBv46 Slave module during the posted write transactions. This is enabled when C_SPLB_SUPPORT_BURSTS = 1. The write buffer is implemented in the bridge to free up the master transactions to other cores that may be on the PLB. The Write Buffer contains a FIFO of width 32/64-bit and depth of 16. The width of the FIFO is directly dependent on C_SPLB_NATIVE_DWIDTH. The Write Buffer passes the write data to the AXI Master module.

Read Buffer

The Read Buffer stores the read data from the AXI Master module during out-of-order read transactions. This is enabled when C_M_AXI_SUPPORTS_THREADS = 1. When enabled the address pipelining depth on PLB is two and outstanding addresses issued on AXI are two. The read buffer is needed when these back to back read transfers on AXI are responded in out-of-order by AXI slaves. The Read Buffer contains a FIFO of width 32/64-bit and depth of 16. The width of the FIFO is directly dependent on C_SPLB_NATIVE_DWIDTH. The Read Buffer passes the read data to the PLBv46 Slave module.

Bridge Control Logic

The PLBV46 to AXI Bridge needs to split a burst transfer that crosses a 4K byte boundary as required by AXI. The Bridge Control Logic module generates the 4KB crossing control signals and provides the length and address signals to the AXI Master module. This module is not used when C_SPLB_SUPPORT_BURSTS = 0 as AXI4-Lite interface is used on AXI side.

Register and Interrupt

The Register and Interrupt module contains the bridge registers and generates interrupt. This is enabled when both parameters C_EN_ERR_REGS and C_SPLB_SUPPORT_BURSTS are set to 1. These registers capture the PLB request status and qualifiers as well as the target address when a write or read transaction generates an error on the AXI side. Interrupt is generated to report these errors. Please refer Register Descriptions section for more details.

The register accesses are always 32-bit and only PLB single transfers are acknowledged in register address space. The slave size is always 32-bit even when C_SPLB_NATIVE_DWIDTH is 64. This module is not implemented when C_SPLB_SUPPORT_BURSTS = 0 and the error information is sent on Sl_MRdErr and Sl_MWrErr signals. Also the interrupt signal is not used.

AXI Master

The AXI Master module provides a bi-directional AXI master interface on the AXI. This interface can be AXI memory-mapped interface (full AXI4) or AXI4-Lite interface (control interface) depending on the parameter C_SPLB_SUPPORT_BURSTS. When C_SPLB_SUPPORT_BURST = 0, only single transfers on PLB will be supported and AXI4-Lite interface will be used on AXI side. When C_SPLB_SUPPORT_BURSTS = 1, AXI full interface is used on AXI. The AXI data bus width can be 32 or 64-bits in AXI full interface and always fixed at 32 when AXI4-Lite interface is used. This module receives read data from AXI and transmits to either read buffer when read buffer is enabled or to PLBv46 Slave module when read buffer is disabled. During write transfers the write data is received from the write buffer. Depending on the design parameters, AXI Master module controls the supported limited cache encoding (cacheable/bufferable) and limited protection encoding (secure/non-secure) signals.

I/O Signals

Table 1 shows the I/O signals of the PLBV46 to AXI Bridge.

Table 1: **I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
PLB System Signals					
P1	SPLB_Clk	System	I	-	PLB clock
P2	SPLB_Rst	System	I	-	PLB reset, active high
P3	Interrupt(1)	System	O	0	Bridge Interrupt (Edge sensitive, rising)

Table 1: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
PLB Interface Signals					
P4	SPLB_ABus[0 : C_SPLB_AWIDTH -1]	PLB	I	-	PLB address bus
P5	SPLB_PAValid	PLB	I	-	PLB primary address valid
P6	SPLB_masterID[0 : C_SPLB_MID_WIDTH - 1]	PLB	I	-	PLB current master identifier
P7	SPLB_RNW	PLB	I	-	PLB read not write
P8	SPLB_BE[0 : (C_SPLB_DWIDTH/8) - 1]	PLB	I	-	PLB byte enables
P9	SPLB_size[0 : 3]	PLB	I	-	PLB size of requested transfer
P10	SPLB_type[0 : 2]	PLB	I	-	PLB transfer type
P11	SPLB_wrDBus[0 : C_SPLB_DWIDTH - 1]	PLB	I	-	PLB write data bus
P12	SPLB_SAValid	PLB	I	-	PLB secondary address valid
P13	SPLB_MSize[0 : 1]	PLB	I	-	PLB data bus width indicator
PLB Slave Interface Signals					
P14	Sl_addrAck	PLB	O	0	Slave address acknowledge
P15	Sl_SSize[0 : 1]	PLB	O	0	Slave data bus size
P16	Sl_wait	PLB	O	0	Slave wait
P17	Sl_rearbitrate	PLB	O	0	Slave bus rearbitrate
P18	Sl_wrDAck	PLB	O	0	Slave write data acknowledge
P19	Sl_wrComp	PLB	O	0	Slave write transfer complete
P20	Sl_rdDBus[0 : C_SPLB_DWIDTH - 1]	PLB	O	0	Slave read data bus
P21	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
P22	Sl_rdComp	PLB	O	0	Slave read transfer complete
P23	Sl_MBusy[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave busy
P24	Sl_MRdErr[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave read error
P25	Sl_MWrErr[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave write error
P26	Sl_rdWdAddr[0 : 3]	PLB	O	0	Slave read word address
P27	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P28	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
Unused PLB Signals					
P29	SPLB_UABus[0 : 31]	PLB	I	-	PLB upper address bits
P30	SPLB_rdPrim	PLB	I	-	PLB secondary to primary read request indicator

Table 1: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P31	SPLB_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P32	PLB_abort	PLB	I	-	PLB abort bus request
P33	SPLB_busLock	PLB	I	-	PLB bus lock
P34	SPLB_lockErr	PLB	I	-	PLB lock error
P35	SPLB_wrBurst	PLB	I	-	PLB burst write transfer
P36	SPLB_rdBurst	PLB	I	-	PLB burst read transfer
P37	SPLB_wrPendReq	PLB	I	-	PLB pending bus write request
P38	SPLB_rdPendReq	PLB	I	-	PLB pending bus read request
P39	SPLB_wrPendPri[0 : 1]	PLB	I	-	PLB pending write request priority
P40	SPLB_rdPendPri[0 : 1]	PLB	I	-	PLB pending read request priority
P41	SPLB_reqPri[0 : 1]	PLB	I	-	PLB current request priority
P42	SPLB_TAttribute[0 : 15]	PLB	I	-	PLB transfer attribute
P43	Sl_MIRQ[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Master interrupt request
AXI Interface Signals (2)					
AXI Write Address Channel Signals					
P44	M_AXI_AWID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	AXI_FULL	O	0	Write address ID: This signal is the identification tag for the write address group of signals
P45	M_AXI_AWADDR[C_M_AXI_ADDR_WIDTH-1 : 0]	AXI_FULL/ AXI_LITE	O	0	AXI Write address: The write address bus gives the address of the first transfer in a write burst transaction
P46	M_AXI_AWLEN[7 : 0]	AXI_FULL	O	0	Burst length: This signal gives the exact number of transfers in a write burst
P47	M_AXI_AWSIZE[2 : 0]	AXI_FULL	O	0	Burst size: This signal indicates the size of each transfer in the write burst.
P48	M_AXI_AWBURST[1 : 0]	AXI_FULL	O	0	Burst type: This signal coupled with the size information, details how the address for each write transfer within the burst is calculated
P49	M_AXI_AWCACHE[3 : 0]	AXI_FULL	O	0	Cache type: This signal provides additional information about the cacheable characteristics of the write transfer.
P50	M_AXI_AWPROT[2 : 0]	AXI_FULL/ AXI_LITE	O	2	Protection type: This signal indicates the normal, privileged, or secure protection level of the write transaction and whether the transaction is a data access or an instruction access. The default value is normal non secure data access
P51	M_AXI_AWVALID	AXI_FULL/ AXI_LITE	O	0	Write address valid: This signal indicates that valid write address and control information are available
P52	M_AXI_AWREADY	AXI_FULL/ AXI_LITE	I	-	Write address ready: This signal indicates that the slave is ready to accept an address and associated control signals
AXI Write Channel Signals					

Table 1: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P53	M_AXI_WDATA[C_M_AXI_DATA_WIDTH-1 : 0]	AXI_FULL/ AXI_LITE	O	0	Write data bus
P54	M_AXI_WSTB[C_M_AXI_DATA_WIDTH/8-1 : 0]	AXI_FULL/ AXI_LITE	O	0	Write strobes: This signal indicates which byte lanes to update in memory
P55	M_AXI_WLAST	AXI_FULL/ AXI_LITE	O	0	Write last: This signal indicates the last transfer in a write burst
P56	M_AXI_WVALID	AXI_FULL/ AXI_LITE	O	0	Write valid: This signal indicates that valid write data and strobes are available
P57	M_AXI_WREADY	AXI_FULL/ AXI_LITE	I	-	Write ready: This signal indicates that the slave can accept the write data
AXI Write Response Channel Signals					
P58	M_AXI_BID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	AXI_FULL	I	-	Write response ID: This signal is 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
P59	M_AXI_BRESP[1 : 0]	AXI_FULL, AXI_LITE	I	-	Write response: This signal indicates the status of the write transaction
P60	M_AXI_BVALID	AXI_FULL/ AXI_LITE	I	-	Write response valid: This signal indicates that a valid write response is available
P61	M_AXI_BREADY	AXI_FULL/ AXI_LITE	O	1	Response ready: This signal indicates that the master can accept the response information
AXI Read Address Channel Signals					
P62	M_AXI_ARID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	AXI_FULL	O	0	Read address ID: This signal is the identification tag for the read address group of signals
P63	M_AXI_ARADDR[C_M_AXI_ADDR_WIDTH -1 : 0]	AXI_FULL/ AXI_LITE	O	0	Read address: The read address bus gives the initial address of a read burst transaction
P64	M_AXI_ARLEN[7 : 0]	AXI_FULL	O	0	Burst length: The burst length gives the exact number of transfers in a read burst.
P65	M_AXI_ARSIZE[2 : 0]	AXI_FULL	O	0	Burst size: This signal indicates the size of each transfer in the read burst.
P66	M_AXI_ARBURST[1 : 0]	AXI_FULL	O	0	Burst type: The burst type, coupled with the size information, details how the address for each read transfer within the burst is calculated.
P67	M_AXI_ARCACHE[3 : 0]	AXI_FULL	O	0	Cache type: This signal provides additional information about the cacheable characteristics of the read transfer.
P68	M_AXI_ARPROT[2 : 0]	AXI_FULL/ AXI_LITE	O	2	Protection type: This signal provides protection unit information for the read transaction. The default value is normal non secure data access
P69	M_AXI_ARVALID	AXI_FULL/ AXI_LITE	O	0	Read address valid: This signal indicates, when HIGH, that the read address and control information is valid and will remain stable until the address acknowledgement signal, ARREDY, is high.
P70	M_AXI_ARREADY	AXI_FULL/ AXI_LITE	I	-	Read address ready: This signal indicates that the slave is ready to accept an address and associated control signals.

Table 1: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
AXI Read Data Channel Signals					
P71	M_AXI_RID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	AXI_FULL	I	-	Read ID tag: This signal is the 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.
P72	M_AXI_RDATA[C_M_AXI_DATA_WIDTH -1 : 0]	AXI_FULL/ AXI_LITE	I	-	Read data bus
P73	M_AXI_RRESP[1 : 0]	AXI_FULL/ AXI_LITE	I	-	Read response: This signal indicates the status of the read transfer.
P74	M_AXI_RLAST	AXI_FULL/ AXI_LITE	I	-	Read last: This signal indicates the last transfer in a read burst
P75	M_AXI_RVALID	AXI_FULL/ AXI_LITE	I	-	Read valid: This signal indicates that the required read data is available and the read transfer can complete
P76	M_AXI_RREADY	AXI_FULL/ AXI_LITE	O	1	Read ready: This signal indicates that the master can accept the read data and response information
P77	M_AXI_AWLOCK	AXI_FULL	O	0	Lock type: This signal provides additional information about the atomic characteristics of the write transfer
P78	M_AXI_ARLOCK	AXI_FULL	O	0	Lock type: This signal provides additional information about the atomic characteristics of the read transfer.
Unused AXI Signals					
P79	M_AXI_ACLK	AXI_FULL/ AXI_LITE	I	-	AXI Clock - SPLB_Clk is used on AXI side
P80	M_AXI_ARESETN	AXI_FULL/ AXI_LITE	I	-	AXI Reset - SPLB_Rst is used on AXI side

Notes:

1. This signal is not used when C_SPLB_SUPPORT_BURSTS = 0 or C_EN_ERR_REGS = 0 as error registers are not enabled.
2. AXI_FULL interface refers to AXI Memory mapped interface (AXI Full) enabled when C_SPLB_SUPPORT_BURSTS = 1 and AXI_LITE interface refers to AXI4-Lite interface enable when C_SPLB_SUPPORT_BURSTS = 0.

Design Parameters

Table 2 shows the design parameters of the PLBV46 to AXI Bridge.

Inferred Parameters

In addition to the parameters listed in Table 2, there are also parameters that are inferred for each AXI interface in the EDK tools. Through the design, these EDK-inferred parameters control the behavior of the AXI Interconnect.

For a complete list of the interconnect settings related to the AXI interface, see DS768 AXI Interconnect IP Data Sheet.

Table 2: **Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Values	VHDL Type
System Parameter					
G1	Target FPGA family	C_FAMILY	virtex6, spartan6	virtex6	string
PLB Parameters					
G2	PLB least significant address bus width	C_SPLB_AWIDTH	32	32	integer
G3	PLB data width	C_SPLB_DWIDTH	32, 64, 128	32	integer
G4	Width of the Slave Data Bus	C_SPLB_NATIVE_DWIDTH	32,64	32	integer
G5	Selects point-to-point or shared bus topology 0 = Shared Bus Topology 1 = Point-to-Point Bus Topology	C_SPLB_P2P (1)	0 - 1	0	integer
G6	PLB Master ID Bus Width	C_SPLB_MID_WIDTH	$\log_2$(C_SPLB_NUM_MASTERS) with a minimum value of 1	1	integer
G7	Number of PLB Masters	C_SPLB_NUM_MASTERS	1 - 16	1	integer
G8	Support Bursts 0 = Do not support bursts (AXI4-Lite on AXI interface) 1 = Support bursts (AXI Full on AXI interface)	C_SPLB_SUPPORT_BURSTS	0 - 1	1	integer
G9	Support Cacheline transfers 0 = Do not support cacheline transfers 1 = Support cacheline transfers	C_SPLB_SUPPORT_CACHELINE(2)	0 - 1	0	integer
G10	Number of AXI address ranges	C_SPLB_NUM_ADDR_RNGS	1 - 4(3)	1	integer
G11	PLB Offset Address for all ranges	C_SPLB_RNGS_OFFSET	Valid address(4)(5)	0x0	std_logic_vector
G12	PLB base address for address range 1	C_SPLB_RNG1_BASEADDR	Valid address(5)	None(4)	std_logic_vector
G13	PLB high address for address range 1	C_SPLB_RNG1_HIGHADDR	Valid address(6)	None(4)	std_logic_vector
G14	Range1 non-secure or secure access 0 = Secure normal data access 1 = Non-secure normal data access	C_SPLB_RNG1_NONSEC_SEC(7)	0 - 1	1	integer
G15	Range1 cache encoding See Cache Support, page 16 for details.	C_SPLB_RNG1_CACHEABLE_BUFFERABLE(8)	0 - 3	0	integer
G16	PLB base address for address range 2	C_SPLB_RNG2_BASEADDR	Valid address(5)	None(4)	std_logic_vector

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

Generic	Feature/Description	Parameter Name	Allowable Values	Default Values	VHDL Type
G17	PLB high address for address range 2	C_SPLB_RNG2_HIGHADDR	Valid address(6)	None(4)	std_logic_vector
G18	Range 2 non-secure or secure access 0 = Secure normal data access 1 = Non-secure normal data access	C_SPLB_RNG2_NONSEC_SEC(7)	0 - 1	1	integer
G19	Range2 cache encoding See Cache Support, page 16 for details.	C_SPLB_RNG2_CACHEABLE_BUFFERABLE(8)	0 - 3	0	integer
G20	PLB base address for address range 3	C_SPLB_RNG3_BASEADDR	Valid address(6)	None(4)	std_logic_vector
G21	PLB high address for address range 3	C_SPLB_RNG3_HIGHADDR	Valid address(6)	None(4)	std_logic_vector
G22	Range 3 non-secure or secure access 0 = Secure normal data access 1 = Non-secure normal data access	C_SPLB_RNG3_NONSEC_SEC(7)	0 - 1	1	integer
G23	Range3 cache encoding See Cache Support, page 16 for details.	C_SPLB_RNG3_CACHEABLE_BUFFERABLE(8)	0 - 3	0	integer
G24	PLB base address for address range 4	C_SPLB_RNG4_BASEADDR	Valid address(5)	None(4)	std_logic_vector
G25	PLB high address for address range 4	C_SPLB_RNG4_HIGHADDR	Valid address(6)	None(4)	std_logic_vector
G26	Range 4 non-secure or secure access 0 = Secure normal data access 1 = Non-secure normal data access	C_SPLB_RNG4_NONSEC_SEC(7)	0 - 1	1	integer
G27	Range4 cache encoding See Cache Support, page 16 for details.	C_SPLB_RNG4_CACHEABLE_BUFFERABLE(8)	0 - 3	0	integer
G28	Bridge Base Address when internal debug registers are enabled	C_SPLB_BRIDGE_BASEADDR	Valid address(9)	None(4)	std_logic_vector
G29	Bridge High Address when internal debug registers are enabled	C_SPLB_BRIDGE_HIGHADDR	Valid address(9)	None(4)	std_logic_vector
AXI Parameters					
G30	AXI Identification tag width	C_M_AXI_THREAD_ID_WIDTH(10)	1-2	1	integer

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

Generic	Feature/Description	Parameter Name	Allowable Values	Default Values	VHDL Type
G31	Indicates generation of more than one outstanding transfers 0 = Master generates one master ID 1= Master generates two master IDs	C_M_AXI_SUPPORTS_THREADS(10)(11)	0-1	0	integer
G32	AXI most significant address bus width	C_M_AXI_ADDR_WIDTH	32	32	integer
G33	AXI data bus width	C_M_AXI_DATA_WIDTH	32, 64	32(12)	integer
EDK Tool Parameters					
G34	Supports narrow bursts	C_SUPPORTS_NARROW_BURST	0-1	0(13)	integer
G35	Maximum number of data-active read transactions generated. This will be set as the READ_ACCEPTANCE parameter on the interconnect.	C_INTERCONNECT_M_AXI_READ_ISSUING	1-4	2(14)	integer
G36	Maximum number of data-active write transactions generated. This will be set as the WRITE_ACCEPTANCE parameter on the interconnect.	C_INTERCONNECT_M_AXI_WRITE_ISSUING	1-4	2(14)	integer
G37	AXI interface type	C_M_AXI_PROTOCOL(15)	axi4,axi4lite	axi4	string
PLBV46 to AXI Bridge specific Parameters					
G38	Enable Error Registers for error information and generating interrupt 0 = No error registers are implemented 1 = Error registers are implemented	C_EN_ERR_REGS(16)(17)	0 - 1	0	integer
G39	Enable byte swapping from PLB to AXI 0 = No swapping is performed 1 = Byte swapping is performed	C_EN_BYTE_SWAP	0-1	0	integer

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

Generic	Feature/Description	Parameter Name	Allowable Values	Default Values	VHDL Type

Notes:

1. When C_SPLB_P2P is set to 1, the PLBV46 to AXI Bridge does not require address range specified by C_SPLB_RNGx_BASEADDR and C_SPLB_RNGx_HIGHADDR. Also C_SPLB_RNGS_OFFSET is not valid.
2. This can be enabled only when C_SPLB_SUPPORT_BURSTS = 1. When C_SPLB_SUPPORT_CACHELINE is set to zero, 4-word and/or 8-word cache line transactions are not supported on PLB. It is recommended to set this to 1 when PLB master generates cache line transfers.
3. Four sets of address ranges can be specified for the bridge so that different protection and cache encoding can be selected for different address ranges. 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 burst transactions, the base address for this address range must be aligned to a 64-byte address.
4. No default value will be specified to insure that the actual value is set, i.e., if the value is not set, a compiler error will be generated. High address - base address must be a power of 2.
5. Only valid if C_SPLB_P2P = 0 and should be word aligned. C_SPLB_RNGx_BASEADDR+C_SPLB_OFFSET represents the base AXI address that PLB is allowed to access for the range x (x varies from 1 to 4). For example, if C_SPLB_OFFSET is 0x00000000, C_SPLB_RNG1_BASEADDR will represent the physical address of AXI. C_SPLB_RNG1_BASEADDR value of 0x00000000 will go to physical address 0x00000000. A value of 0x02000000 will go to physical address 0x02000000. If you increase the C_SPLB_OFFSET to 0x03000000, a C_SPLB_RNG1_BASEADDR value of 0x00000000 will go to physical address 0x03000000, a C_SPLB_RNG1_BASEADDR value of 0x02000000 will go to physical address 0x05000000.
6. C_SPLB_RNGx_HIGHADDR+C_SPLB_OFFSET represents the high AXI address that the PLB is allowed to access for the range x.
7. The selected protection level is used for the entire range of bridge address and for all the AXI transactions. M_AXI_ARPROT[0], M_AXI_AWPROT[0], M_AXI_ARPROT[2], M_AXI_AWPROT[2] M_AXI_ARPROT[3] and M_AXI_AWPROT[3] bits are set to zero.
8. The selected transaction attributes are used for the entire range of bridge address and for all the AXI transactions. Read allocate and Write allocate are set to zero.
9. The user must set these values only when C_EN_ERR_REGS = 1. The C_SPLB_BRIDGE_BASEADDR must be a multiple of the range, where the range is C_SPLB_BRIDGE_HIGHADDR - C_SPLB_BRIDGE_BASEADDR + 1.
10. This parameter is not used when C_SPLB_SUPPORT_BURSTS = 0, as the AXI interface is AXI4-Lite and the number of outstanding transfers is always 1.
11. SPLB_SAValid will be used only when C_M_AXI_SUPPORTS_THREADS = 1.
12. C_M_AXI_DATA_WIDTH value will be set same as C_SPLB_NATIVE_DWIDTH. C_M_AXI_DATA_WIDTH will be set to 32 when C_SPLB_SUPPORT_BURSTS = 0 as AXI4-Lite interface is used on AXI side.
13. This parameter is used by Interconnect and updated automatically. When C_SPLB_NATIVE_DWIDTH is 64, C_SUPPORTS_NARROW_BURST is set to 1. When C_SPLB_NATIVE_DWIDTH is 32, C_SUPPORTS_NARROW_BURST is set to 0 as narrow transfers are not generated.
14. This parameter is used by Interconnect and updated automatically. See Table 7 and section Outstanding Requests on AXI for more details.
15. When C_SPLB_SUPPORT_BURSTS = 1, C_M_AXI_PROTOCOL is updated automatically to axi4 and when C_SPLB_SUPPORT_BURSTS = 0, C_M_AXI_PROTOCOL is updated automatically to axi4lite.
16. When C_SPLB_P2P = 1, and C_EN_ERR_REGS = 1 all the PLB requests other than the register space address range (C_SPLB_BRIDGE_BASEADDR to C_SPLB_BRIDGE_BASEADDR + 0xF) is translated to AXI. When C_SPLB_P2P = 1, and C_EN_ERR_REGS = 0 all the PLB requests are translated to AXI.
17. C_EN_ERR_REGS is set to 0, when C_SPLB_SUPPORT_BURSTS = 0 as error registers are not required.

Allowable Parameter Combinations

When C_EN_ERR_REGS = 1 and C_SPLB_SUPPORT_BURSTS = 1, C_SPLB_BRIDGE_BASEADDR and C_SPLB_BRIDGE_HIGHADDR must be specified. The address range specified by C_SPLB_BRIDGE_BASEADDR and C_SPLB_BRIDGE_HIGHADDR must be a power of 2, and must be at least 0xF in size.

For example, if C_SPLB_BRIDGE_BASEADDR = 0xE0000000, C_SPLB_BRIDGE_HIGHADDR must be at least = 0xE000000F.

Parameter - I/O Signal Dependencies

The dependencies between the PLBV46 to AXI Bridge core design parameters and I/O signals are described in Table 3. In addition, 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 and related output signals are set to a specified value.

Table 3: **Parameter-I/O Signal Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G3	C_SPLB_DWIDTH		G4	C_SPLB_DWIDTH should be greater than or equal to C_SPLB_NATIVE_DWIDTH
G3	C_SPLB_DWIDTH	P8, P11, P20	-	Affects the number of bits of read and write data bus and byte enables
G4	C_SPLB_NATIVE_DWIDTH		G8	The allowed value of C_SPLB_NATIVE_DWIDTH is 32 when C_SPLB_SUPPORT_BURSTS = 0
G5	C_SPLB_P2P	G10 to G27		When C_SPLB_P2P = 1, as address decoding is not needed the generics related to address ranges are not used
G5	C_SPLB_P2P	P16, P17		When C_SPLB_P2P = 1, Sl_wait is driven when the bridge is busy. When C_SPLB_P2P = 0, Sl_rearbitrate is driven when the bridge is busy.
G6	C_SPLB_MID_WIDTH	P6	G9	This value is calculated as: $\log_2$(C_SPLB_NUM_MASTERS) with a minimum value of 1
G7	C_SPLB_NUM_MASTERS	P23, P24, P25	-	Affects the width of the Sl_MBusy, Sl_MWrErr and Sl_MRdErr
G8	C_SPLB_SUPPORT_BURSTS	P44, P46 to P49, P58, P62, P64 to P67, P71, P77, P78		When burst support is disabled AXI4-Lite interface is used and signals that are used for AXI Full interface are not used
G9	C_SPLB_SUPPORT_CACHELINE		G8	C_SPLB_SUPPORT_CACHELINE is valid only when C_SPLB_SUPPORT_BURSTS = 1
G28	C_SPLB_BRIDGE_BASEADDR		G8, G34	C_SPLB_BRIDGE_BASEADDR is valid only when C_EN_ERR_REGS = 1 and C_SPLB_SUPPORT_BURSTS = 1
G29	C_SPLB_BRIDGE_HIGHADDR		G8, G34	C_SPLB_BRIDGE_HIGHADDR is valid only when C_EN_ERR_REGS = 1 and C_SPLB_SUPPORT_BURSTS = 1
G30	C_M_AXI_THREAD_ID_WIDTH		G8	C_M_AXI_THREAD_ID_WIDTH is valid only when C_SPLB_SUPPORT_BURSTS = 1
G31	C_M_AXI_SUPPORTS_THREADS		G8	C_M_AXI_SUPPORTS_THREADS is valid only when C_SPLB_SUPPORT_BURSTS = 1

Table 3: **Parameter-I/O Signal Dependencies *(Cont'd)***

Generic or Port	Name	Affects	Depends	Relationship Description
G33	C_M_AXI_DATA_WIDTH		G4, G8	C_M_AXI_DATA_WIDTH is same as C_SPLB_NATIVE_DWIDTH when C_SPLB_SUPPORT_BURSTS = 1. Will be fixed at 32 when C_SPLB_SUPPORT_BURSTS = 0
G33	C_M_AXI_DATA_WIDTH		P53, P54, P72	Affects the number of bits of read and write data bus and byte enables
G34	C_EN_ERR_REGS		G8	C_M_AXI_SUPPORTS_THREADS is valid only when C_SPLB_SUPPORT_BURSTS = 1
I/O Signals				
P3	Interrupt	-	G8, G34	Interrupt signal is available only when C_EN_ERR_REG = 1 and C_SPLB_SUPPORT_BURSTS = 1
P6	SPLB_masterID[0 : C_SPLB_MID_WIDTH - 1]	-	G6	Width of the SPLB_mastedID varies according to C_SPLB_MID_WIDTH
P8	SPLB_BE[0 : (C_SPLB_DWIDTH/8) -1]	-	G3	Width of the SPLB_BE varies according to C_SPLB_DWIDTH
P11	SPLB_wrDBus[0 : C_SPLB_DWIDTH - 1]	-	G3	Width of the SPLB_wrDBus varies according to C_SPLB_DWIDTH
P20	Sl_rdDBus[0 : C_SPLB_DWIDTH - 1]	-	G3	Width of the Sl_rdDBus varies according to C_SPLB_DWIDTH
P23	Sl_MBusy[0 : C_SPLB_NUM_MASTERS - 1]	-	G7	Width of the Sl_MBusy varies according to C_SPLB_NUM_MASTERS
P24	Sl_MWrErr[0 : C_SPLB_NUM_MASTERS - 1]	-	G7	Width of the Sl_MWrErr varies according to C_SPLB_NUM_MASTERS
P25	Sl_MRdErr[0 : C_SPLB_NUM_MASTERS - 1]	-	G7	Width of the Sl_MRdErr varies according to C_SPLB_NUM_MASTERS
P53	M_AXI_WDATA[C_M_AXI_DATA_WIDTH -1 : 0]	-	G33	Width of the M_AXI_WDATA varies according to C_M_AXI_DATA_WIDTH
P54	M_AXI_WSTRB[(C_M_AXI_DATA_WIDTH/8) -1 : 0]	-	G33	Width of the M_AXI_WSTRB varies according to C_M_AXI_DATA_WIDTH
P72	M_AXI_RDATA[C_M_AXI_DATA_WIDTH -1 : 0]	-	G33	Width of the M_AXI_RDATA varies according to C_M_AXI_DATA_WIDTH

Design Details

Clocking

The PLBV46 to AXI Bridge is a synchronous design and will use the PLB clock at both PLB and AXI interfaces.

Reset

SPLB_Rst is synchronous reset input that resets the bridge upon assertion. The SPLB_Rst is also used to reset AXI interface.

Byte Invariance

AXI is little endian and PLB is big endian. The PLBV46 to AXI Bridge maintains byte invariance, or using Xilinx IP terminology, byte addressing integrity is maintained for both 32 and 64-bit width data in the bridge design when C_EN_BYTE_SWAP = 1. This means that a 32/64-bit data from any address on the PLBV46 bus has the bytes swapped in traversing the bridge so that byte data of byte lanes of the same numerical address offsets yields the same byte data when read from the little endian AXI side or by a remote master on the big endian PLB side. For byte transactions, any byte addressed data read from the AXI side or the PLB side yields the same byte of data. Write strobe signals from the AXI master port are similarly swapped. Byte and strobe swapping are shown in Figure 2 for 32-bit data width on PLB and AXI (C_SPLB_NATIVE_DWIDTH = 32). When C_EN_BYTE_SWAP = 0, no bytes are swapped. The following Table 4 shows the data bits swap and Table 5 shows byte enables swap from PLB to AXI for different values of C_EN_BYTE_SWAP.

Table 4: **Data bits swap from PLB to AXI when C_SPLB_NATIVE_DWIDTH = 32**

PLB data bits	AXI data bits when C_EN_BYTE_SWAP = 1	AXI data bits when C_EN_BYTE_SWAP = 0
D0 - D7	D7 - D0	D31 - D24
D8 - D15	D15 - D8	D23 - D16
D16- D23	D23 - D16	D15 - D8
D24 - D31	D31 - D24	D7 - D0

Table 5: **Byte enables swap from PLB to write strobes on AXI when C_SPLB_NATIVE_DWIDTH = 32**

PLB byte enables	AXI write strobes when C_EN_BYTE_SWAP = 1	AXI write strobes when C_EN_BYTE_SWAP = 0
BE0	WSTRB0	WSTRB3
BE1	WSTRB1	WSTRB2
BE2	WSTRB2	WSTRB1
BE3	WSTRB3	WSTRB0

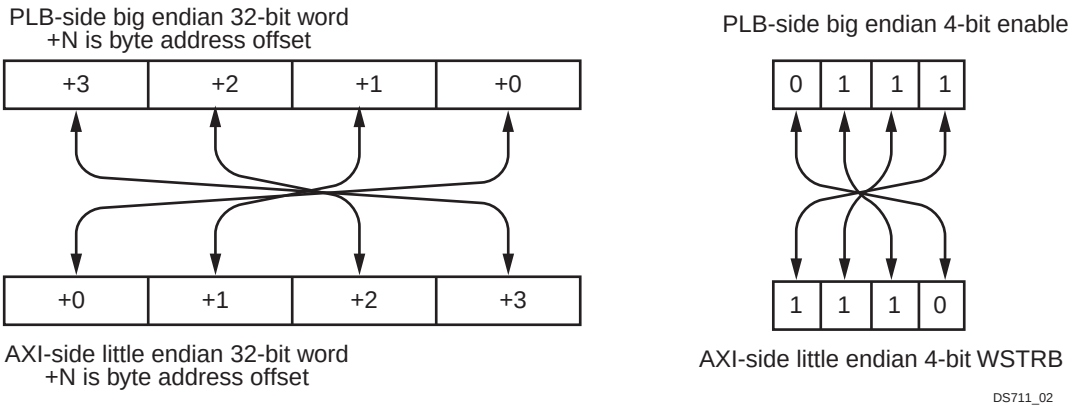

Figure 2: **Byte Data Swap and WrSTRB Swap to BEs as Data Traverses the PLBV46 to AXI Bridge**

Memory Mapping

The AXI memory map and the PLB memory map are one single complete 32-bit (4 GB) memory space. The PLBV46 slave module in the bridge does not modify the address for AXI; hence, the address that is presented on the AXI is exactly as received on the PLB when C_SPLB_RNGS_OFFSET is set to "0x00000000".

Address Decoding

Address decoding is required in a shared bus interconnect scheme when C_SPLB_P2P = 0. In a Point to Point configuration (C_SPLB_P2P = 1), there is only one PLB master that communicates with the PLBV46 to AXI Bridge. So, the bridge responds to all addresses regardless of the address and the PLB Slave module may be able to reduce resource utilization by eliminating the address decode function and modifying interface behavior to allow for a reduction in latency.

In a shared bus topology (C_SPLB_P2P = 0), the PLBV46 to AXI Bridge decodes the address presented on the address bus.

Relationships Between the Write AXI Channels

As the relationship between the address, write data, and write response channels is flexible on AXI, the PLBV46 to AXI bridge issues the write address independent of write data and vice versa.

Read Ordering

When C_SPLB_SUPPORT_BURSTS = 1 and C_M_AXI_SUPPORTS_THREADS = 1 the PLBV46 to AXI Bridge issues the reads on AXI with different read transfer ID values. The transfers that are requested on SPLB_SAValid are sent on AXI with different M_AXI_ARID. The read reordering depth is 2 and read data interleaving is supported among these transfers. The out-of-order read completion on AXI is supported by storing the read data. The AXI read slave error is not sent on PLB (on Sl_MRdErr) for the reads that are completed out of order on AXI.

When C_M_AXI_SUPPORTS_THREADS is set to 0, the AXI master module issues the reads with same read transfer ID values so that they are received in order.

When C_SPLB_SUPPORT_BURSTS = 0, only PLB single read transfers are supported and IDs are not used.

Write Ordering

When C_SPLB_SUPPORT_BURSTS = 1 and C_M_AXI_SUPPORTS_THREADS = 1, the PLBV46 to AXI Bridge issues write transactions with different transfer ID values where the data ordering depth is 2 and allows the write responses in out-of-order. The transfers that are requested on SPLB_SAValid are sent on AXI with different M_AXI_WID. However the bridge issues the data of write transaction in the same order in which it issues the transaction addresses as PLB sends the write data in order.

When C_M_AXI_SUPPORTS_THREADS is set to 0, the AXI master module issues the writes with same write transfer ID values so that they are received in order.

The write error response is not sent on PLB (on Sl_MWrErr) as the write data acknowledge is sent on PLB before the data is sent on AXI. The user has to enable the error registers (set C_EN_ERR_REGS = 1) for such errors.

When C_SPLB_SUPPORT_BURSTS = 0, only PLB single write transfers are supported and IDs are not used.

Read and Write Ordering

When a read followed by a write (or vice versa) is issued to the same address from PLB, the PLBV46 to AXI Bridge implements an address check against the outstanding transactions and makes sure the transactions are issued and completed in order.

When a write followed by write (or read followed by read) to the same address is issued from PLB, the PLBV46 to AXI Bridge does not implement the address check against the two addresses and issues these transactions with different ID values and assumes that the transactions will complete in order.

AXI Response Signaling

EXOKAY is considered as OKAY.

Protection Unit Support

Protection unit support is limited in PLBV46 to AXI Bridge. Privileged and instruction accesses are not supported. All the transactions are normal data accesses. Either secure or non-secure is selected during configuration by the parameter C_SPLB_RNGx_NONSEC_SEC. When this is set to 0, M_AXI_ARPROT[1] & M_AXI_AWPROT[1] are set to 0 for address range x (x varies from 1 to 4). When this is set to 1, M_AXI_ARPROT[1] & M_AXI_AWPROT[1] are set to 1.

When C_SPLB_P2P = 1, M_AXI_ARPROT[1] & M_AXI_AWPROT[1] are set to '1' and remaining bits are set to zero.

Cache Support

The bufferable and cacheable transaction attributes of AXI transfers are selected by parameter C_SPLB_RNGx_CACHEABLE_BUFFERABLE. Assignment of M_AXI_AWCACHE and M_AXI_ARCACHE for different values of C_SPLB_RNGx_CACHEABLE_BUFFERABLE is shown in Table 6. When C_SPLB_P2P = 1, M_AXI_ARCACHE[3:0] & M_AXI_AWCACHE[3:0] are set to zeroes for all the AXI requests.

Table 6: **Assignment of M_AXI_AWCACHE and M_AXI_ARCACHE**

C_SPLB_RNGx_CACHEABLE_BUFFERABLE	M_AXI_AWCACHE[3:0]	M_AXI_ARCACHE[3:0]
0	"0000"	"0000"
1	"0001"	"0001"
2	"0010"	"0010"
3	"0011"	"0011"

Bridge Error Conditions

An error on AXI results with the response of SLVERR or DECERR. As the bridge supports posted writes and out-of-order reads, these errors can't be sent on PLB. For this reason the PLBV46 to AXI Bridge implements optional Slave Error Address Register (SEAR) and Slave Error Status Register (SESR). The SESR/SEAR registers are accessible from the PLB and are used for system integration and debug or error event logging by a user application. These registers captures PLB request status and qualifiers as well as the target address when a read or write transaction generates an error on the AXI side. Interrupt signal is driven by the PLB slave to the system interrupt controller to report these errors, when interrupts are enabled by using Device Global Interrupt Enable Register (DGIE) and Device Interrupt Enable Register (DIER). When both write and read requests on AXI generates errors, write error has more priority than read error, so the status qualifiers shows the information of write request that caused the error.

When C_SPLB_P2P = 0, BAR (Base Address Roll over) error is generated when the PLB address overruns the C_SPLB_RNGx_HIGHADDR for range x. When C_SPLB_P2P = 1, BAR error will be generated when PLB address overruns 0xFFFFFFFF. BAR error is not applicable for single transfers when C_SPLB_SUPPORT_BURSTS = 0.

It is the user's responsibility not to issue burst transfers that crosses the PLBV46 to AXI bridge's high address. During such transfers PLB address will not be acknowledged by the bridge and PLB_MTimeout will be issued by the arbiter after 16 clock cycles. An edge-sensitive interrupt will be generated by the bridge if C_EN_ERR_REGS is 1 and interrupts are enabled. The SESR register shows the status of the transfer that caused BAR error and the SEAR shows the address of the transfer.

Bridge Time Out Condition

Data phase time out is not implemented inside the bridge. When a request is issued from PLB, the bridge translates this request into corresponding AXI transfer and requests on AXI. If this request is not responded by AXI, the PLBV46 to AXI bridge and hence PLB waits indefinitely. There is no mechanism implemented inside the PLBV46 to AXI bridge to come out of this kind of situation. It is assumed that AXI responds to all of the AXI requests.

4 KB Crossing

As per AXI specification, bursts must not cross 4 KB boundaries to prevent them from crossing boundaries between slaves and to limit the size of the address incrementer required within slaves. PLBV46 to AXI Bridge takes care of this inside the bridge by splitting the PLB burst transfer into 2 requests when PLB issues a burst transfer that crossed 4KB boundary.

Outstanding Requests on AXI

The number of outstanding write /read requests on AXI can be more than 1 when C_SPLB_SUPPORT_BURSTS = 1 and C_M_AXI_SUPPORTS_THREADS = 1. The read/write transfers that are requested on SPLB_SAValid are requested on AXI with different ID and the reordering depth is 2. Therefore, the outstanding read/write request are 2. When 4 KB crossing is detected in a PLB word or double-word burst in both primary and secondary transfers, outstanding write/read requests are 4 (2 for the requests on SPLB_PAValid and 2 for the requests on SPLB_SAValid).

The following Table 7 shows more details on number of outstanding requests that are generated on AXI depending on the generic combinations.

Table 7: **Outstanding write/read requests**

C_SPLB_SUPPORT_BURSTS	C_M_AXI_SUPPORT_THREADS	C_INTERCONNECT_M_AXI_READ_ISSUING/C_INTERCONNECT_M_AXI_WRITE_ISSUING
0	NA	1
1	0	2
1	1	4

AXI4-Lite Operation

When C_SPLB_SUPPORT_BURST = 0, only single transfers will be supported on PLB and AXI4-Lite interface will be used on AXI side. For all the other PLB transfers (Example: line and burst transfers), the PLBV46 to AXI bridge will not respond and PLB_MTimeout will be issued by the arbiter after 16 clock cycles.

Register Descriptions

Table 8 shows all the PLBV46 to AXI Bridge registers and their addresses. These registers are enabled by setting C_EN_ERR_REGS to 1. The registers are not used when C_SPLB_SUPPORT_BURSTS = 0 as no posted writes and out of order reads are supported.

Table 8: **PLBV46 to AXI Bridge Registers** (1)

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
C_SPLB_BRDIGE BASEADDR + 0x0	SESR	R/W(2)	0x0	Slave Error Status Register
C_SPLB_BRIDGE_BASEADDR + 0x4	SEAR	R(3)	0x0	Slave Error Address Register
C_SPLB_BRIDGE_BASEADDR + 0x8	DGIE	R/W	0x0	Device Global Interrupt Enable Register
C_SPLB_BRIDGE_BASEADDR + 0xC	DIER	R/W	0x0	Device Interrupt Enable Register

Notes:

1. These registers are included only when C_EN_ERR_REGS is set to 1.
2. This register is written with a data value of 0xA0000000 to reset SESR and SEAR.
3. Read only register. Writing into this register has no effect.

Slave Error Status Register (SESR) and Slave Error Address Register (SEAR)

The following section details the register descriptions of the SESR and SEAR. These registers are included only when C_EN_ERR_REGS is set to 1.

They are used to provide transaction error information to the user application. When these registers are enabled, a BAR error and slave error or decode error from the AXI will cause a capture trigger to occur for the SESR and the SEAR. The SESR captures the PLB transaction qualifiers and the SEAR captures the PLB address for the first offending command. Once captured, the data is retained until the user application reads the data from the registers and then rearms the capture mechanism by writing a 0xA0000000 to the SESR address. This write clears the captured information from the SESR and SEAR. Any other write access to SESR will not generate an error on PLB and has no effect.

It should be noted by the user that the assertion of BAR error, slave error or decode error can be used to generate an interrupt to the User Application. This requires enabling Device Global Interrupt Enable Register and Device Interrupt Enable Register. This interrupt may then be used by the User Application to signal the need to service the SESR and SEAR.

When C_EN_ERR_REGS is set to 0, the BAR error and errors on AXI cannot be reported to PLB. It is assumed that the User Application does not issue transactions that generates errors on AXI.

The SESR is shown in Figure 3 and detailed in Table 9. The SEAR is shown in Figure 4 and detailed in Table 10.

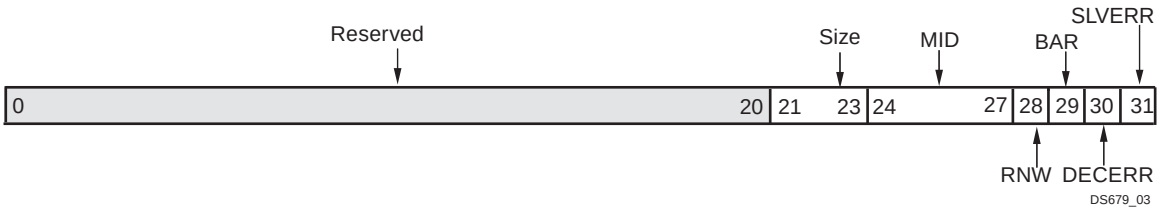

Figure 3: **Slave Error Status Register (SESR)**

Table 9: Slave Error Status Register (SESR) Bit Definitions

Bit(s)	Name	Core Access	Reset Value	Description
0-20	Reserved	N/A	0	Reserved
21-23	Size	R/W(1)	“0000”	PLB Size: This value reflects the SPLB_size qualifier at the time of error capture.See IBM PLB Specification for SPLB_size description.
24-27	MID	R/W(1)	“0000”	PLB Master ID: This value reflects the SPLB_masterID qualifier at the time of error capture. See IBM PLB Specification for SPLB_masterID description.
28	RNW	R/W(1)	‘0’	PLB RNW: This bit reflects the state of the SPLB_RNW signal at the time of the error capture. ‘0’ = Write command. ‘1’ = Read command.
29	BAR	R/W(1)	‘0’	BAR Error:(2)(3) This bit will be asserted when a PLB address over runs the address range of the bridge. ‘0’ = No BAR Error asserted. ‘1’ = BAR Error asserted.
30	DECERR	R/W(1)	‘0’	Decode Error: This bit will be asserted when a decode error (DECERR) is received from the AXI interconnect component. This indicates that there is no slave at the transaction address. ‘0’ = No Decode Error asserted. ‘1’ = Decode Error asserted.
31	SLVERR	R/W(1)	‘0’	Slave Error: This bit will be asserted whenever a slave error (SLVERR) is received from the AXI Slave. This indicates that the access has reached the AXI slave successfully, but the slave wishes to return an error condition. ‘0’ = No Slave Error asserted. ‘1’ = Slave Error asserted.

Notes:

1. This register is cleared by the user application via system reset or a write to the SESR address with a data value of 0xA0000000.
2. During BAR error, the PLBV46 to AXI bridge does not send address acknowledge due to which PLB_MTimeout is asserted by arbiter. This transfer is not sent on AXI.
3. BAR error is applicable for only burst transfers. This bit is always zero when C_SPLB_SUPPORT_BURSTS = 0.

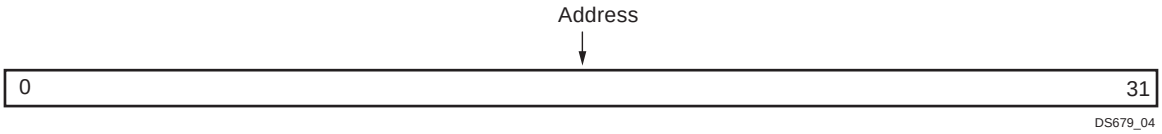

Figure 4: **Slave Error Address Register (SEAR)**

Table 10: **Slave Error Address Register (SEAR) Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0-31	Address(0 to 31)	R[1]	Zeros	Transaction Address(0-31): This value reflects the PLB address (0 to 31) qualifier at the time of error capture. If the PLB Address bus is wider than 32 bits, this register will contain the Least Significant 32-bit slice of the address.

Notes:

1. This register is cleared by the user application via reset or a write to the SESR address with a data value of 0xA0000000

Device Global Interrupt Enable Register (DGIE)

The Device Global Interrupt Enable Register provides the final enable/disable for the interrupt output and resides in the Register and Interrupt Module. It is a read/write register addressed at an offset 0x8 from base address C_SPLB_BRIDGE_BASEADDR. If interrupts are globally disabled (the DGIE bit is set to '0', there will be no interrupt from the bridge under any circumstances. This is a single bit read/write register as shown in Figure 5. Table 11 shows the DGIE bit definitions.

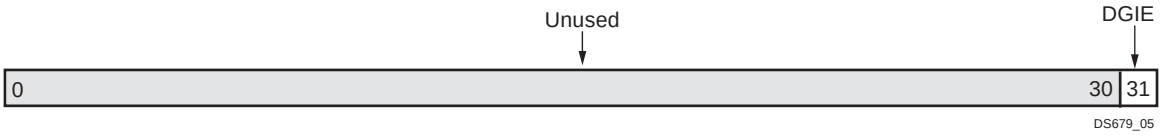

Figure 5: **Device Global Interrupt Enable Register (DGIE)**

Table 11: **Device Global Interrupt Enable Register (DGIE) Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 to 30	Unused	N/A	0	Unused
31	DGIE	Read/Write	'0'	Device Global Interrupt Enable: Master Enable for routing Device Interrupt to the System Interrupt Controller. '1' = Enabled '0' = Disabled

Device Interrupt Enable Register (DIER)

The Device Interrupt Enable Register (DIER) is shown in Figure 6. It is a read/write register addressed at an offset 0xC from base address C_SPLB_BRIDGE_BASEADDR. The bit definitions of this register are as shown in Table 12. The Device Global Interrupt Enable Register provides the final enable/disable for the interrupt output to the processor and resides in the Register and Interrupt Module.

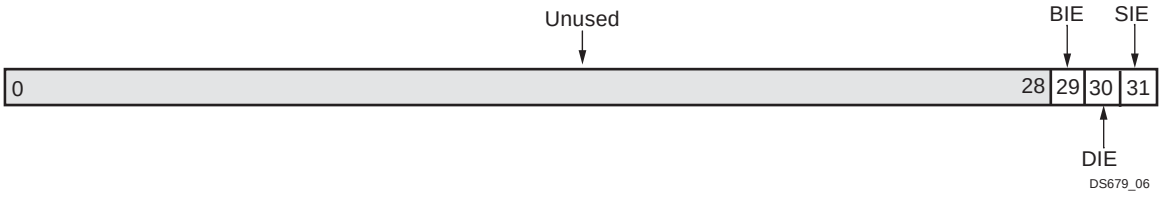

Figure 6: **Device Interrupt Enable Register (DIER)**

Table 12: **Device Interrupt Enable Register (DIER) Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 to 30	Unused	N/A	0	Unused
29	BIE (1)	Read/Write	'0'	BAR Interrupt Enable: Interrupt Enable bit for routing BAR error to the System Interrupt Controller. '1' = Interrupt will assert in response to BAR Error '0' = Interrupt will not assert in response to BAR Error
30	DIE	Read/Write	'0'	DECERR Interrupt Enable: Interrupt Enable bit for routing Decode error to the System Interrupt Controller. '1' = Interrupt will assert in response to DECERR '0' = Interrupt will not assert in response to DECERR
31	SIE	Read/Write	'0'	SLVERR Interrupt Enable: Interrupt Enable bit for routing Slave error to the System Interrupt Controller. '1' = Interrupt will assert in response to SLVERR '0' = Interrupt will not assert in response to SLVERR

Notes:

1. BAR error is applicable for only burst transfers. This bit is not used when C_SPLB_SUPPORT_BURSTS = 0.

Bridge Transaction Translation

Table 13 shows translation of PLBV46 transaction to AXI transactions. For one PLB transaction, two AXI transactions must be requested when a 4KB cross is detected in a PLB transfer. AXI allows WRAP type burst transactions of 2, 4, 8, and 16 words; however, PLB only supports 4, 8 word line transactions.

When C_SPLB_NATIVE_DWIDTH = 64, the M_AXI_DATA_WIDTH is set to 64, and a 32-bit PLB master request of a word burst of length 16, is sent on AXI as a INCR burst transfer of length 16 with burst size 4 bytes in transfer (as a narrow transfer).

When C_SPLB_NATIVE_DWIDTH = 32, the M_AXI_DATA_WIDTH is set to 32 and a 64-bit PLB master request of a double word burst of length 16, is sent on AXI as a INCR burst transfer of burst length 32 with burst size 4 bytes as the maximum burst length supported on AXI4 is 256.

Table 13: **PLB Transaction to AXI Transaction**

PLB Transaction	AXI Transaction	Description
Single read or write of 1 to 4 bytes on a 32-bit PLB	Burst read or write of INCR type with burst length as 1.	When PLB issues a single read with 1/2/3 bytes enabled, AXI issues it as INCR burst with burst length 1 and burst size 2 (number of bytes as 4) and controls strobes during writes and discards the unused bytes during read
Single read or write of 1 to 8 bytes on a 64-bit PLB	Burst read or write of INCR type with burst length as 1.	When PLB issues a single read with 1 to 7 bytes enabled, AXI issues it as INCR burst with burst length 1 and burst size 3(number of bytes as 8) and controls strobes during writes and discards the unused bytes during read

Table 13: **PLB Transaction to AXI Transaction**

PLB Transaction	AXI Transaction	Description
4 word cacheline read or write	Burst read or write of WRAP type with burst length as 4.	AXI is always target word first.
8 word cacheline read or write	Burst read or write of WRAP type with burst length as 8	AXI is always target word first.
Word Burst read or write of length 2 to 16	Burst read or write of INCR type with burst length as 2 to 16 respectively.	One PLB burst transaction is translated to one AXI burst transaction.
Double word burst read or write of length 2 to 16	Burst read or write of INCR type with burst length as 4 to 32 respectively when the SPLB_NATIVE_DWIDTH is 32. Burst read or write of INCR type with burst length as 2 to 16 respectively, when the SPLB_NATIVE_DWIDTH is 64.	One PLB burst transaction is translated to one AXI burst transactions.
Word/Double word Burst read or write that crosses 4KB boundary	Burst read or write of INCR type with burst lengths that depends on the requested address and length of PLB transfer.	One PLB burst transaction is translated to two AXI burst transactions.

Not Supported Features and Limitations

PLBV46 Slave interface

- PLB master size greater than 64 bit

The following PLB features and behaviors are not supported as Xilinx simplification of PLBV46 does not support them:

- Aborts
- Non-Memory transfer types (DMA Flyby, Buffered, peripheral to memory, memory to peripheral and DMA memory to memory are ignored)
- Fixed length burst transfer requests of 17 to 256 data beats
- Fixed length bursts of size byte and half word
- Premature fixed length burst terminations
- Indeterminate burst transfers
- Cache line transfers of 16 words
- Parity
- Transfer attributes
- PLB bus locked transfers
- Pending request and priority input information
- Slave to master interrupts

AXI Master interface

- Interface initialization is not supported
- Quality of service signalling is not supported

The following AXI features are not supported as PLBV46 never generates them:

- FIXED Burst type is not supported
- AXI cache support is limited
  - Bufferable and cacheable attributes can be selected during configuration
  - Read allocate and write allocate attributes are not supported
- Protection unit support is limited
  - Privileged and instruction accesses are not supported
  - Either secure or non-secure is selected during configuration
- Atomic exclusive transactions and lock transactions are not supported. All the AXI transactions are normal accesses
- Unaligned transfers are not supported
- Barrier transfers/Debug transfers/User signals are not supported

Timing Diagrams

The following timing diagrams illustrate the PLBV46 to AXI Bridge operation for various read and write transfers.

1. PLB single write transfer is shown in Figure 7
2. PLB single read transfer is shown in Figure 8
3. PLB 4 word line write transfer are shown in Figure 9
4. PLB 8 word line read transfers are shown in Figure 10
5. PLB Burst write of length 15 transfer are shown in Figure 11
6. PLB Burst read of length 16 transfer are shown in Figure 12
7. PLB Burst Write of length 10 that crosses 4KB boundary is shown in Figure 13. One PLB transfer is split into 2 transfers on AXI as 4KB boundary is crossed
8. PLB Burst Read of length 10 that crosses 4KB boundary is shown in Figure 14. One PLB transfer is split into 2 transfers on AXI as 4KB boundary is crossed
9. PLB back to back read and write transfers are shown in Figure 15
10. PLB Single Write and Read transfer to the PLBV46 to AXI Bridge register DGIE is shown in Figure 16

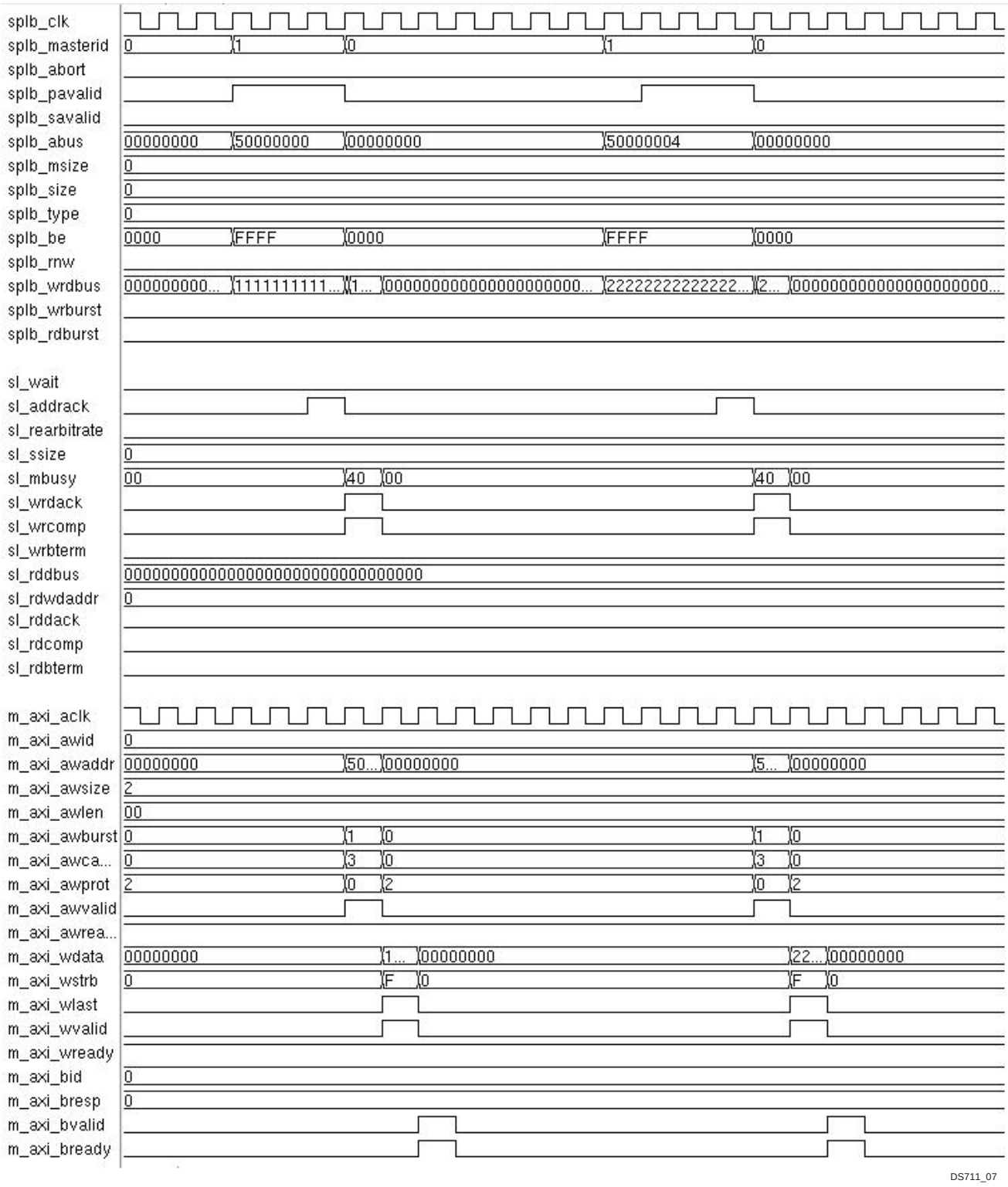

Figure 7: **Single Write Transfer**

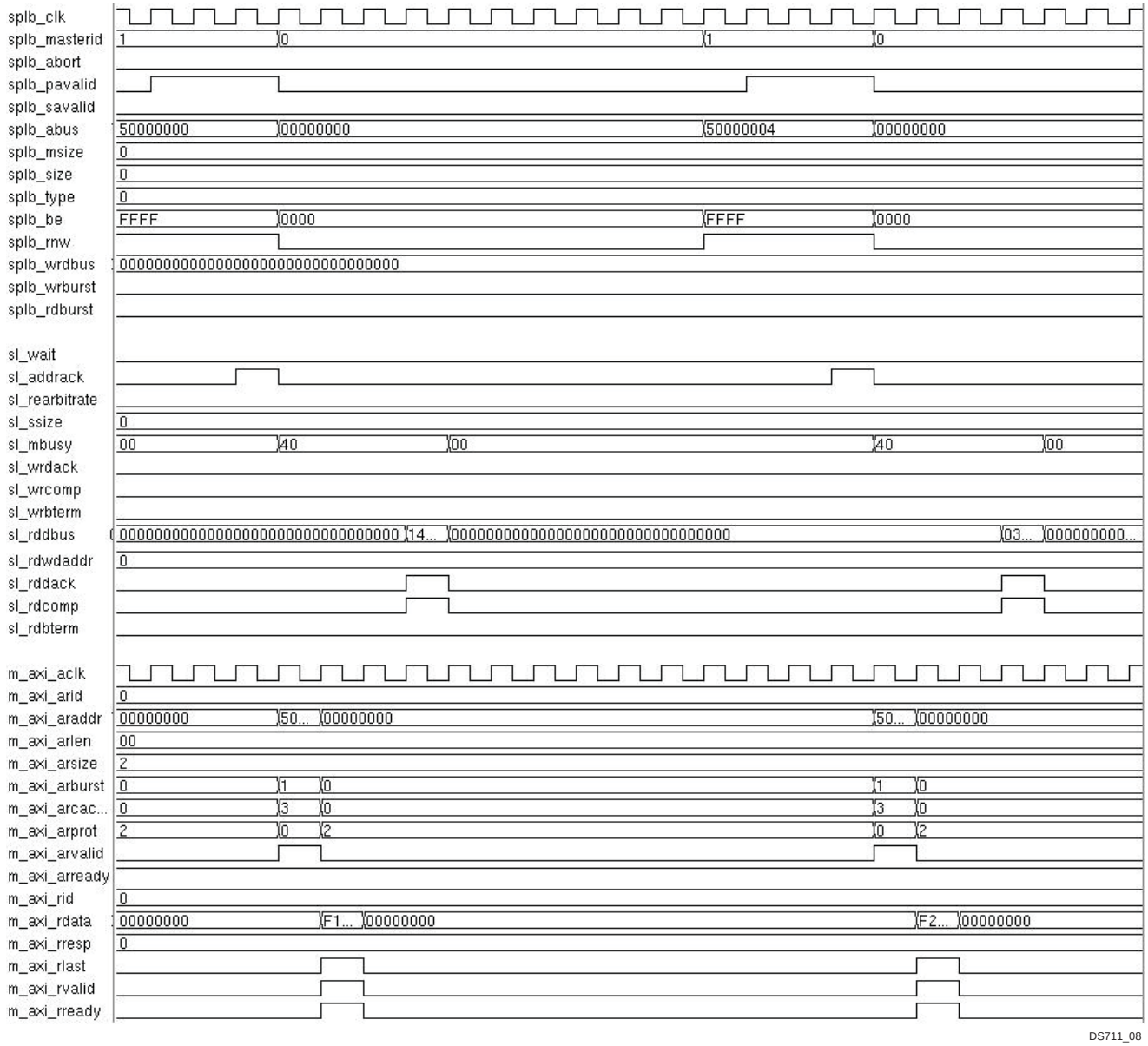

Figure 8: **Single Read Transfer**

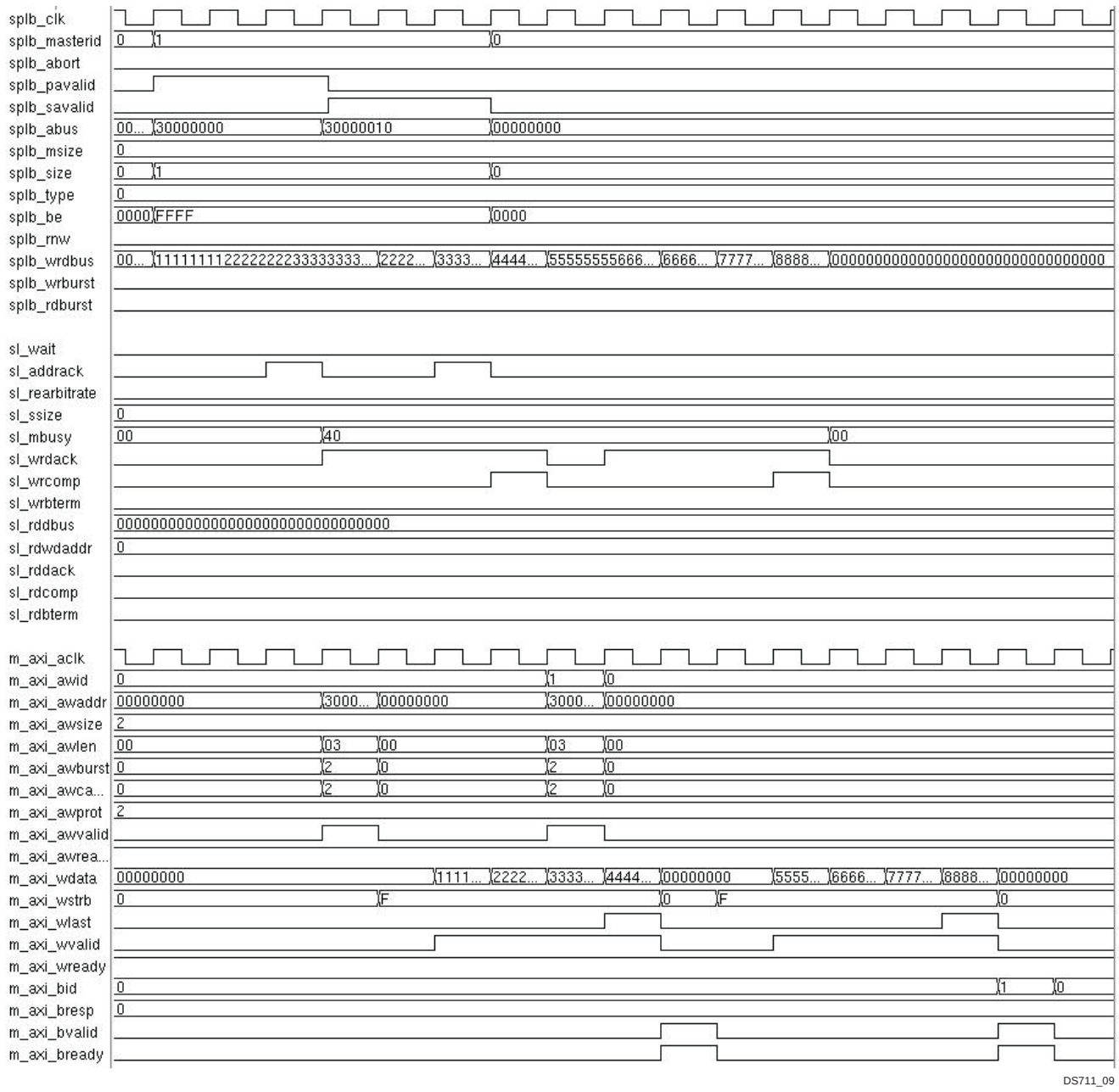

Figure 9: **4 Word Line Write Transfer**

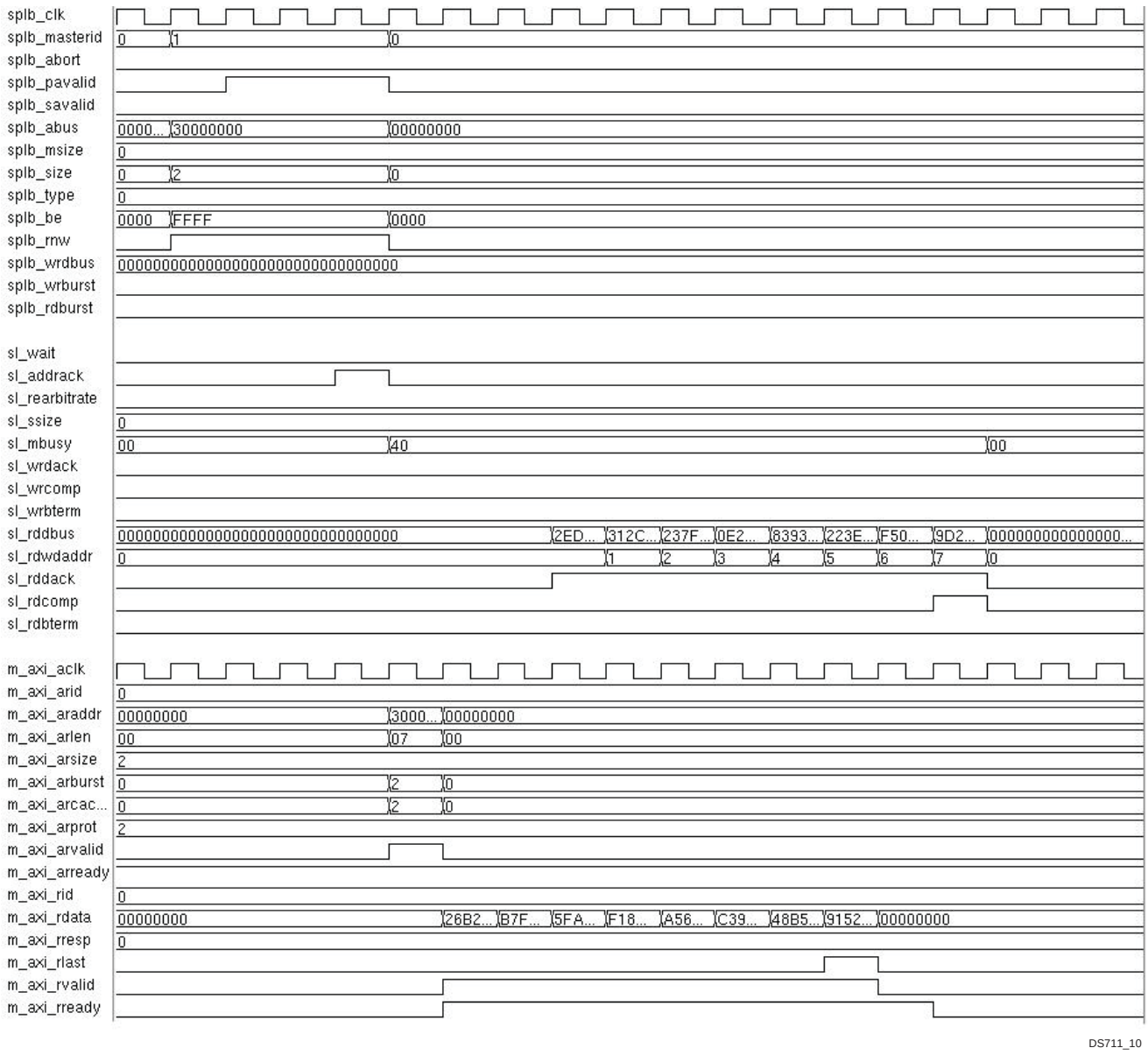

Figure 10: **8 Word Line Read Transfer**

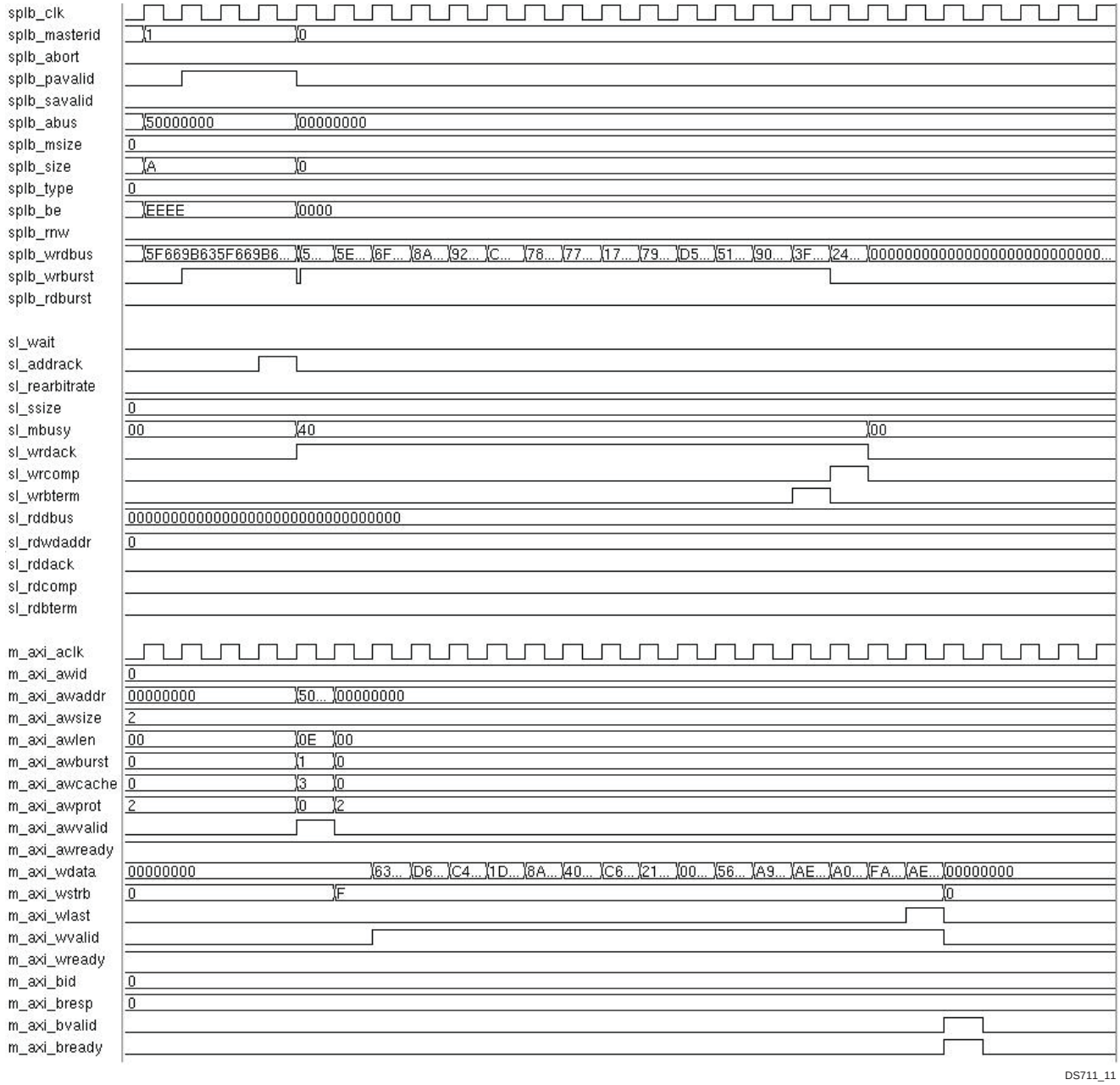

Figure 11: **Burst Write Transfer of Length 15**

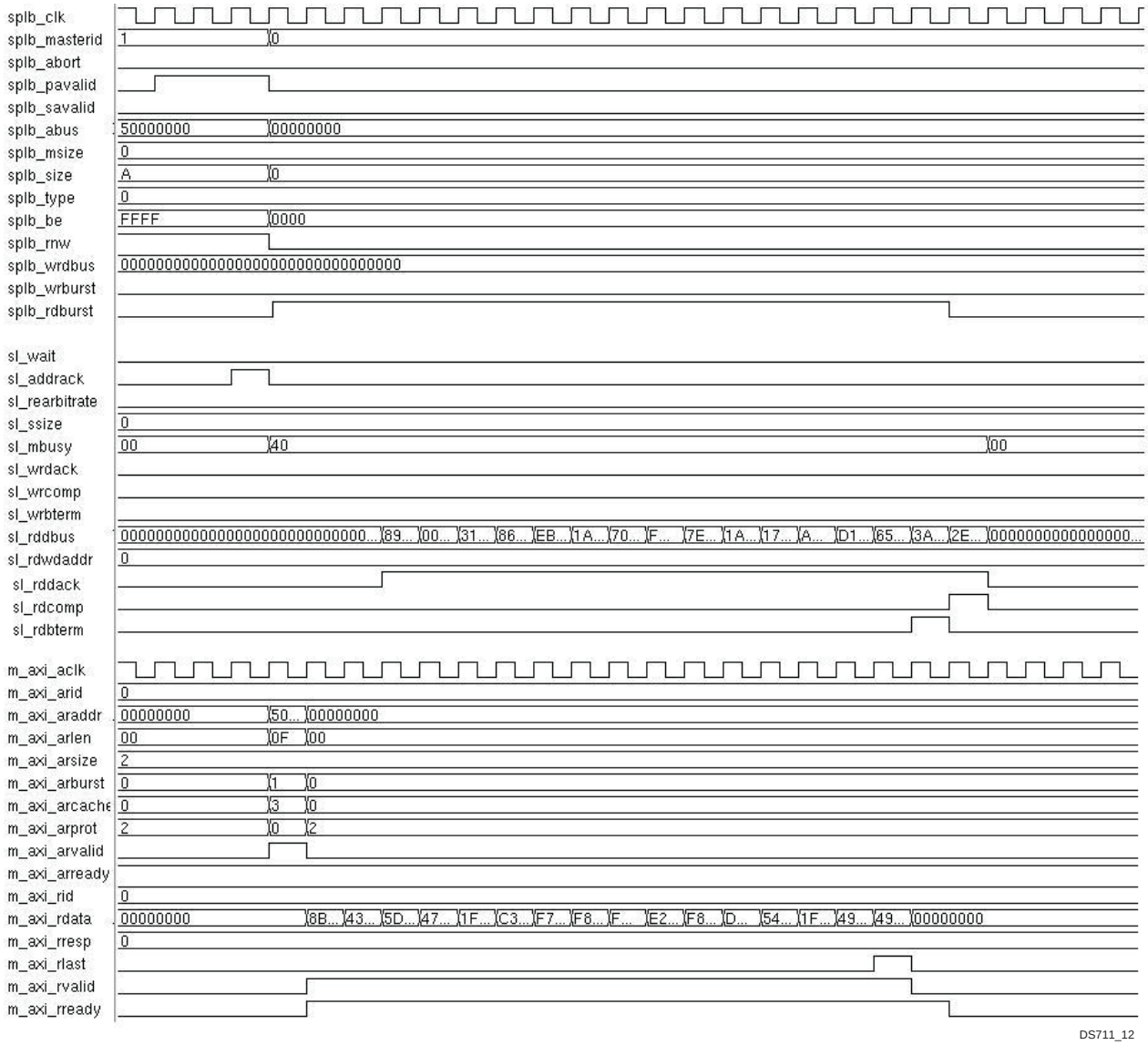

Figure 12: **Burst Read Transfer of Length 16**

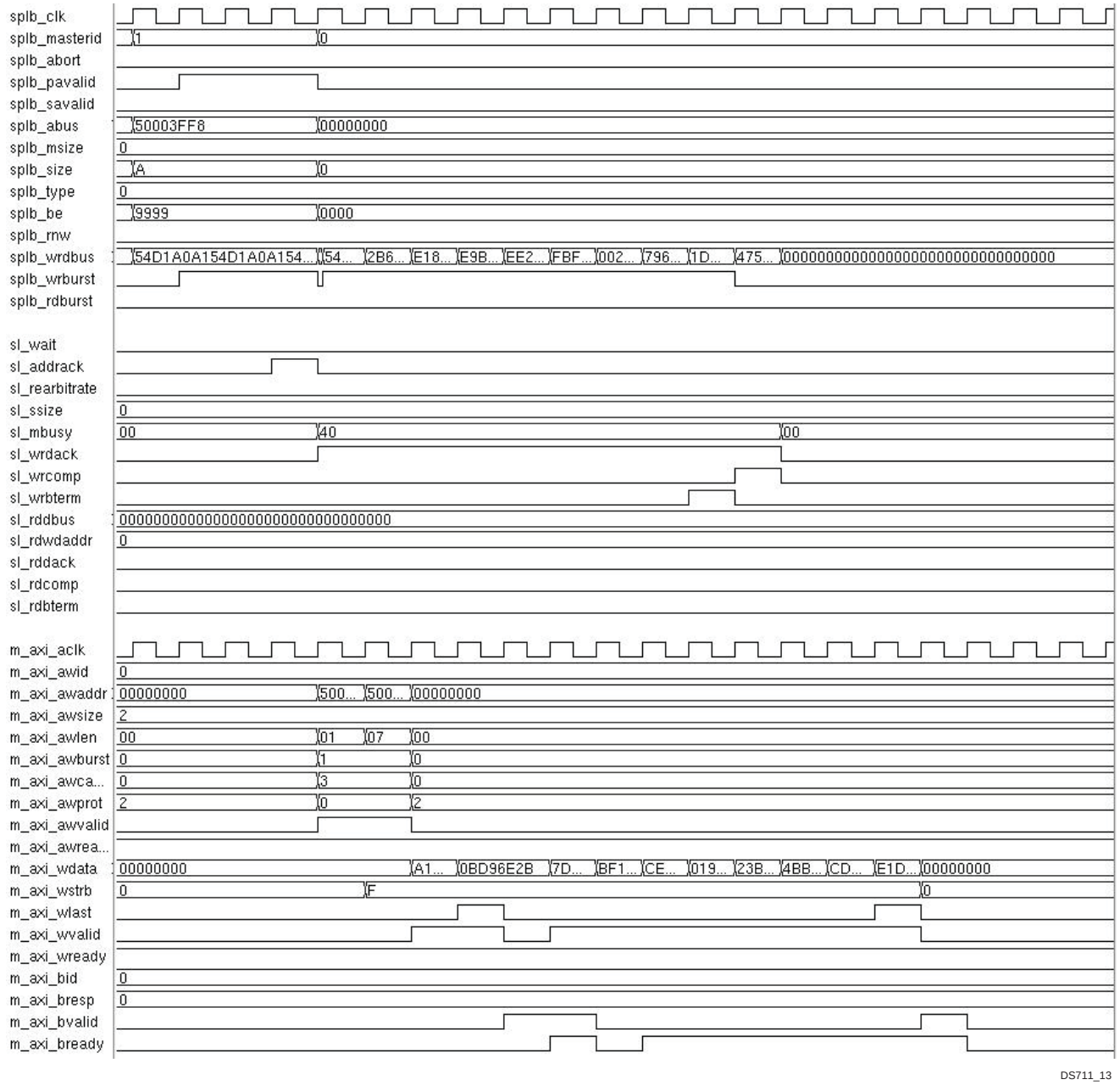

Figure 13: **Burst Write Transfer of Length 10 That Crosses 4KB Boundary on AXI**

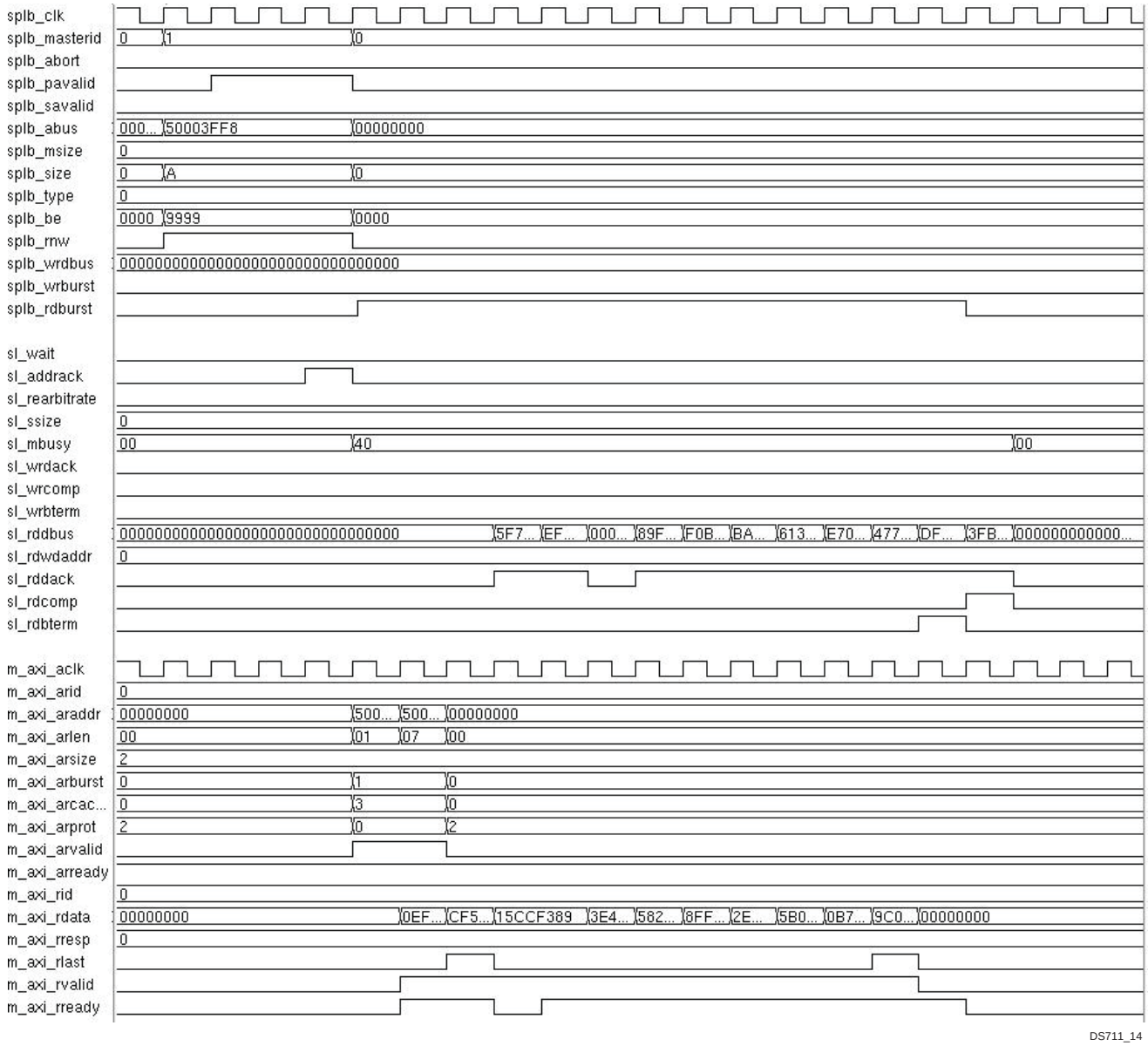

Figure 14: **Burst Read Transfer of Length 10 That Crosses 4KB Boundary on AXI**

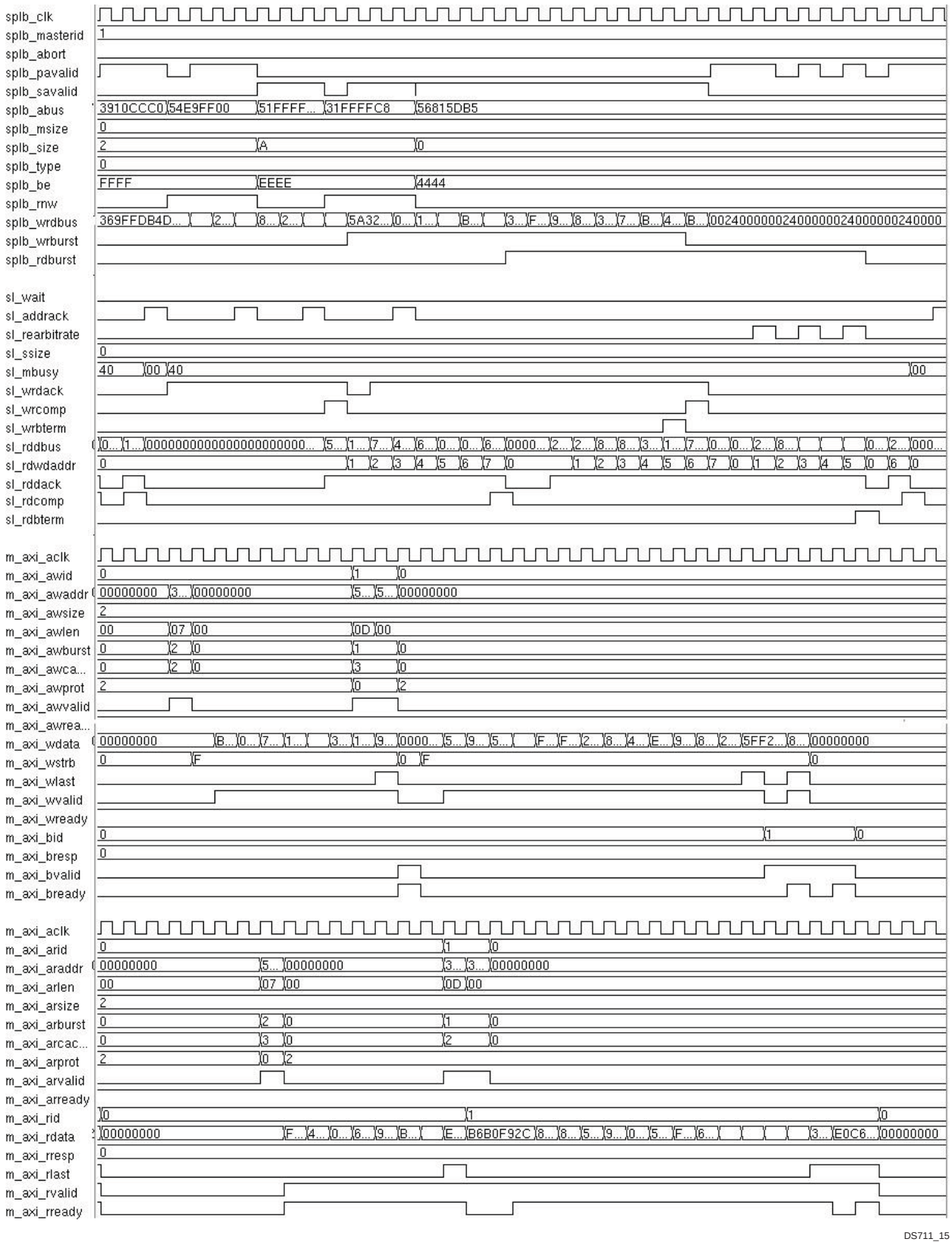

Figure 15: **Back to Back Write and Read Transfers**

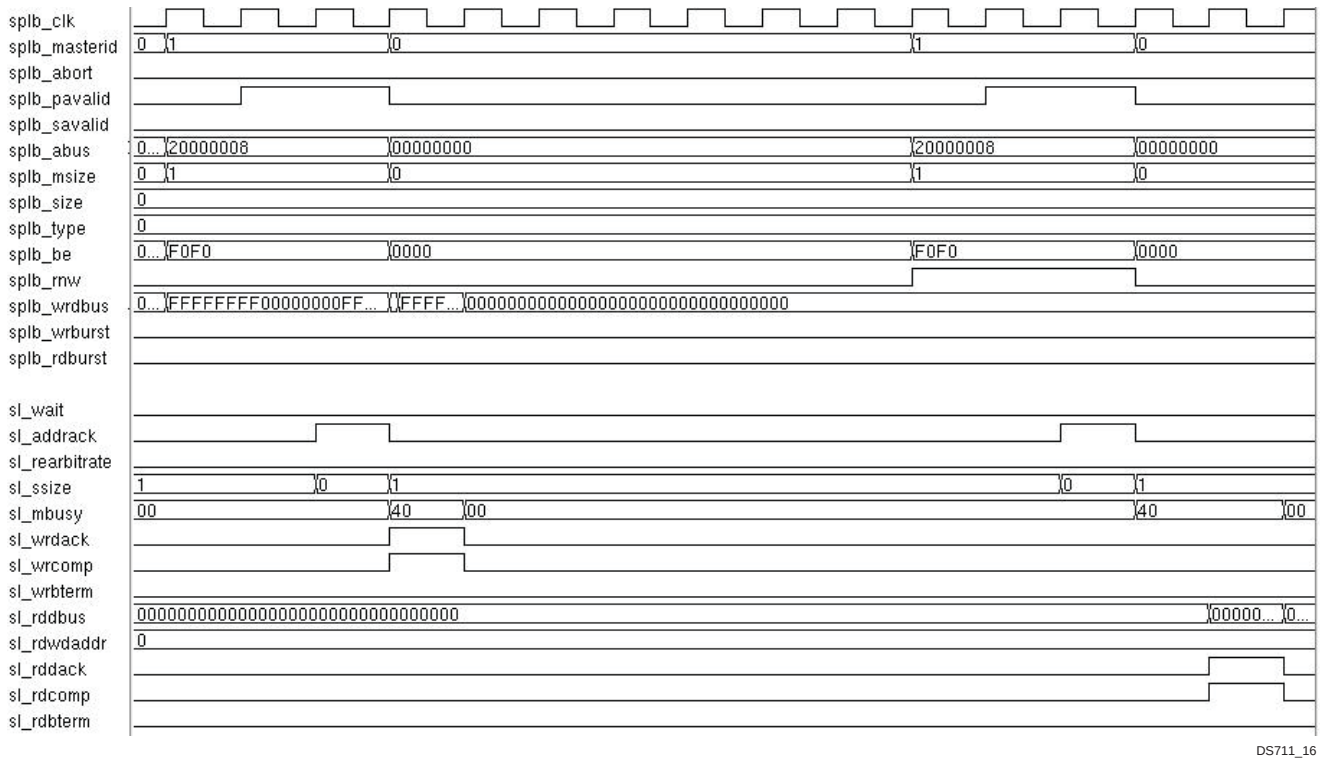

Figure 16: **Single Write and Read Transfers to PLBV46 to AXI Bridge Register (DGIE)**

Device Utilization and Performance Benchmarks

Core Performance

Because the PLBV46 to AXI Bridge is a module that will be used with other design pieces in the FPGA, the resource utilization and timing numbers reported in this section are estimates only. When the PLBV46 to AXI Bridge is combined with other pieces of the FPGA design, the utilization of FPGA resources and timing of the design will vary from the results reported here.

The PLBV46 to AXI Bridge resource utilization benchmarks for a variety of parameter combinations measured with the Artix™-7 FPGA as the target device are shown in Table 14.

Table 14: **Performance and Resource Utilization Benchmarks Artix-7 FPGA (XC7A355TDIE-3)**

Parameter Values (other parameters at default value)									Device Resources			Performance
C_SPLB_P2P	C_SPLB_NATIVE_DWIDTH	C_SPLB_DWIDTH	C_SPLB_SUPPORT_BURSTS	C_SPLB_SUPPORT_CACHELINE	C_EN_ERR_REGS	C_M_AXI_SUPPORTS_THREADS	C_SPLBI_NUM_ADDR_RNGS	C_EN_BYTE_SWAP	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
1	32	32	0	NA	NA	NA	NA	1	63	228	211	257
0	32	64	0	NA	NA	NA	1	0	69	292	204	200
0	32	128	0	NA	NA	NA	1	0	64	292	211	198
0	32	32	1	0	0	0	1	1	188	544	565	223
0	32	32	1	0	1	0	2	1	240	660	650	207
0	32	32	1	1	1	0	3	1	314	690	708	203
0	64	64	1	1	1	0	3	1	322	799	817	200
0	32	128	1	1	1	1	4	1	463	1038	1252	211
0	64	128	1	1	1	1	4	1	513	1170	1467	206
0	64	128	1	1	1	1	1	0	478	1170	1420	210

The PLBV46 to AXI Bridge resource utilization benchmarks for a variety of parameter combinations measured with the Virtex®-7 FPGA as the target device are shown in Table 15.

Table 15: **Performance and Resource Utilization Benchmarks Virtex-7 FPGA (XC7V855T-FFG1157-3)**

Parameter Values (other parameters at default value)									Device Resources			Performance
C_SPLB_P2P	C_SPLB_NATIVE_DWIDTH	C_SPLB_DWIDTH	C_SPLB_SUPPORT_BURSTS	C_SPLB_SUPPORT_CACHELINE	C_EN_ERR_REGS	C_M_AXI_SUPPORTS_THREADS	C_SPLBI_NUM_ADDR_RNGS	C_EN_BYTE_SWAP	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
1	32	32	0	NA	NA	NA	NA	1	63	228	210	250
0	32	64	0	NA	NA	NA	1	0	61	292	220	240
0	32	128	0	NA	NA	NA	1	0	64	292	216	240
0	32	32	1	0	0	0	1	1	197	544	551	250
0	32	32	1	0	1	0	2	1	244	660	682	210
0	32	32	1	1	1	0	3	1	280	690	706	236
0	64	64	1	1	1	0	3	1	316	799	816	222
0	32	128	1	1	1	1	4	1	445	1038	1297	239
0	64	128	1	1	1	1	4	1	516	1170	1449	218
0	64	128	1	1	1	1	1	0	494	1170	1453	231

The PLBV46 to AXI Bridge resource utilization benchmarks for a variety of parameter combinations measured with the Kintex™-7 FPGA as the target device are shown in Table 16.

Table 16: **Performance and Resource Utilization Benchmarks Kintex-7 FPGA (XC7K410T-FFG676-3)**

Parameter Values (other parameters at default value)									Device Resources			Performance
C_SPLB_P2P	C_SPLB_NATIVE_DWIDTH	C_SPLB_DWIDTH	C_SPLB_SUPPORT_BURSTS	C_SPLB_SUPPORT_CACHELINE	C_EN_ERR_REGS	C_M_AXI_SUPPORTS_THREADS	C_SPLBI_NUM_ADDR_RNGS	C_EN_BYTE_SWAP	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
1	32	32	0	NA	NA	NA	NA	1	63	228	210	291
0	32	64	0	NA	NA	NA	1	0	61	292	206	257
0	32	128	0	NA	NA	NA	1	0	67	292	209	236
0	32	32	1	0	0	0	1	1	196	544	554	243
0	32	32	1	0	1	0	2	1	240	660	669	255

Table 16: **Performance and Resource Utilization Benchmarks Kintex-7 FPGA (XC7K410T-FFG676-3)**

Parameter Values (other parameters at default value)									Device Resources			Performance
C_SPLB_P2P	C_SPLB_NATIVE_DWIDTH	C_SPLB_DWIDTH	C_SPLB_SUPPORT_BURSTS	C_SPLB_SUPPORT_CACHELINE	C_EN_ERR_REGS	C_M_AXI_SUPPORTS_THREADS	C_SPLBI_NUM_ADDR_RNGS	C_EN_BYTE_SWAP	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	32	32	1	1	1	0	3	1	247	690	701	269
0	64	64	1	1	1	0	3	1	269	799	836	254
0	32	128	1	1	1	1	4	1	473	1038	1253	249
0	64	128	1	1	1	1	4	1	562	1170	1426	224
0	64	128	1	1	1	1	1	0	548	1170	1409	234

The PLBV46 to AXI Bridge resource utilization benchmarks for a variety of parameter combinations measured with the Virtex-6 FPGA as the target device are shown in Table 17.

Table 17: **Performance and Resource Utilization Benchmarks Virtex-6 FPGA (XC6VLX130T-FF1156-1)**

Parameter Values (other parameters at default value)									Device Resources			Performance
C_SPLB_P2P	C_SPLB_NATIVE_DWIDTH	C_SPLB_DWIDTH	C_SPLB_SUPPORT_BURSTS	C_SPLB_SUPPORT_CACHELINE	C_EN_ERR_REGS	C_M_AXI_SUPPORTS_THREADS	C_SPLBI_NUM_ADDR_RNGS	C_EN_BYTE_SWAP	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
1	32	32	0	NA	NA	NA	NA	1	74	230	243	218
0	32	64	0	NA	NA	NA	1	0	77	306	263	168
0	32	128	0	NA	NA	NA	1	0	73	306	263	170
0	32	32	1	0	0	0	1	1	209	559	640	176
0	32	32	1	0	1	0	2	1	252	667	764	185
0	32	32	1	1	1	0	3	1	208	693	805	177
0	64	64	1	1	1	0	3	1	343	807	934	170
0	32	128	1	1	1	1	4	1	469	1039	1449	160
0	64	128	1	1	1	1	4	1	526	1179	1666	160
0	64	128	1	1	1	1	1	0	504	1179	1675	171

The PLBV46 to AXI Bridge resource utilization benchmarks for a variety of parameter combinations measured with the Spartan®-6 FPGA as the target device are shown in Table 18.

Table 18: **Performance and Resource Utilization Benchmarks Spartan-6 FPGA (XC6SLX100t-FGG900-2)**

Parameter Values (other parameters at default value)									Device Resources			Performance
C_SPLB_P2P	C_SPLB_NATIVE_DWIDTH	C_SPLB_DWIDTH	C_SPLB_SUPPORT_BURSTS	C_SPLB_SUPPORT_CACHELINE	C_EN_ERR_REGS	C_M_AXI_SUPPORTS_THREADS	C_SPLB_NUM_ADDR_RNGS	C_EN_BYTE_SWAP	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
1	32	32	0	NA	NA	NA	NA	1	74	230	242	125
0	32	64	0	NA	NA	NA	1	0	88	306	263	100
0	32	128	0	NA	NA	NA	1	0	79	306	266	101
0	32	32	1	0	0	0	1	1	195	559	630	108
0	32	32	1	0	1	0	2	1	254	667	747	104
0	32	32	1	1	1	0	3	1	272	693	791	107
0	64	64	1	1	1	0	3	1	188	559	618	102
0	32	128	1	1	1	1	4	1	254	667	742	104
0	64	128	1	1	1	1	4	1	460	1179	1589	100
0	64	128	1	1	1	1	1	0	533	1179	1619	100

Read Latency and PLB Bandwidth Utilization

The core is configured for best possible configuration for calculation of latency and bandwidth utilization.
The read latency from address valid (SPLB_PAValid) to first data beat (Sl_rdDAck) of PLBV46 to AXI Bridge is as shown in Table 19.

Table 19: **Read latency in PLB clocks**

C_SPLB_SUPPORT_BURSTS	C_SPLB_P2P	Read Latency
0	1	3 clocks
0	0	4 clocks
1	1	5 clocks
1	0	6 clocks

Best case PLB bandwidth utilization, is calculated on PLB by issuing back to back burst read and write transfers of length 16 and observed in simulation by requesting 1000 transfers, is as shown in Table 20. For improving core performance C_SPLB_SUPPORT_BURSTS and C_M_AXI_SUPPORTS_THREADS need to be set to 1.

Table 20: **PLB Bandwidth utilization**

Transfer Type	Utilization in Percentage
Back to back writes	76%
Back to back reads	80%
Back to back reads and writes	146%

Reference Documents

The following documents contain reference information important to understanding the PLBV46 to AXI Bridge design:

1. *IBM CoreConnect 128-bit Processor Local Bus: Architecture Specification*, version 4.6
2. *ARM® AMBA® AXI4 Protocol Version: 2.0 Specification*
3. *Xilinx PLBv46 Interconnect and Interfaces Simplifications and Feature Subset Specification* (Rev 0.6), August 15, 2006

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

1. *AMBA® AXI Protocol Version: 2.0 Specification* (ARM IHI 0022C)
2. DS768, *AXI Interconnect IP Data Sheet*
3. DS150, *Virtex-6 Family Overview*
4. DS160, *Spartan-6 Family Overview*
5. DS180, *7 Series FPGAs Overview*

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
9/21/10	1.0	Initial Xilinx release
3/1/11	1.1	Updated to v2.00a for the 13.1 release.
6/22/11	2.0	Updated for XPS v13.2. Added support for Artix-7, Kintex-7, and Virtex-7 devices.

Notice of Disclaimer