PLBV46 Master Burst (v1.01a)

DS565 December 14, 2010 **Product Specification**

Introduction

The PLBV46 Master Burst is a continuation of the Xilinx family of IBM CoreConnect compatible LogiCORE products. It provides a bi-directional interface between a User IP core and the PLB v4.6 bus standard. This version of the PLBV46 Master Burst has been designed for PLBV46 Master operations consisting of single data beat read or write transfers and Fixed Length Burst Transfers of 2 to 16 data beats.

Features

- Compatible with IBM CoreConnect 32, 64 and 128-bit PLB
- Parameterizable data width of Client IP Interface (IPIC) to 32, 64, or 128 bits
- Supports Single Beat Read and Write data transfers up to the IPIC data width
    - Automatic Conversion Cycle support for single data beat transfers to/from narrower PLB Slave devices
- Supports Fixed Length Burst Read and Write data transfers of 2 to 16 data beats on the PLB
    - Transfer width is equal to the parameterized IPIC data width
    - Automatic Burst Length Adjustment for bursting to/from narrower PLB Slave devices
- The User interface consists of a Command/Status interface and Read and Write LocalLink interfaces for the data transfer
    - LocalLink transfers can be 1 to 4092 bytes in length with data width equal to the IPIC data width
    - The Master will automatically break IP Client transfer requests requiring more than 16 data beats into multiple fixed length bursts (2 to 16 data beats) on the PLB

LogiCORE IP Facts Table					
Core Specifics					
Supported Device Family	Virtex®-6, Virtex-5, Virtex-4, Spartan®-6, Spartan-3E, Automotive Spartan-3E, Spartan-3, Automotive Spartan-3, Spartan-3A, Automotive Spartan-3A, Spartan-3A DSP, Automotive Spartan-3A DSP				
Supported User Interfaces	PLBV46, LocalLink				
Supported Operating Systems	Windows XP Professional 32-Bit/64-Bit, Windows Vista Business 32-Bit/64-Bit, Red Hat Enterprise Linux WS v4.0 32-bit/64-bit, Red Hat Enterprise Desktop v5.0 32-bit/64-bit (with Workstation Option), SUSE Linux Enterprise (SLE) desktop and server v10.1 32-bit/64-bit				
	Resources				
Configuration	**LUTs**	**FFs**	**DSP Slices**	**Block RAMs**	**Max. Freq.**
	263-645	107-194	161-376	None	
Provided with Core					
Documentation	Product Specification				
Design File Formats	VHDL				
Constraints File	None				
Instantiation Template	VHDL Wrapper				
Reference Designs & Application Notes	None				
Design Tool Requirements					
Xilinx Implementation Tools	ISE® 12.4				
Verification	Mentor Graphics ModelSim 6.5c				
Simulation	Mentor Graphics ModelSim 6.5c				
Synthesis Tools	XST				
Support: Provided by Xilinx, Inc.					

Functional Description

The PLBV46 Master Burst is designed to provide a User with a quick way to implement a mastering interface between User logic and the IBM PLB V4.6. Figure 1 shows a block diagram of the PLBV46 Master Burst. The port references and groupings are detailed in Table 1. The design allows for parameterization of both the Master's internal data width (Native Data Width) and the PLB data width of 32, 64, or 128 bits. Transfer request protocol between the PLB and the User Logic is provided by the Read and Write Controller block. The Bus Width Adapter and Steering Logic block provides the necessary function to connect the Master's internal logic to the three available PLB widths; 32, 64, and 128-bits. The PLB width must be greater than or equal to the Master's Native Data Width.

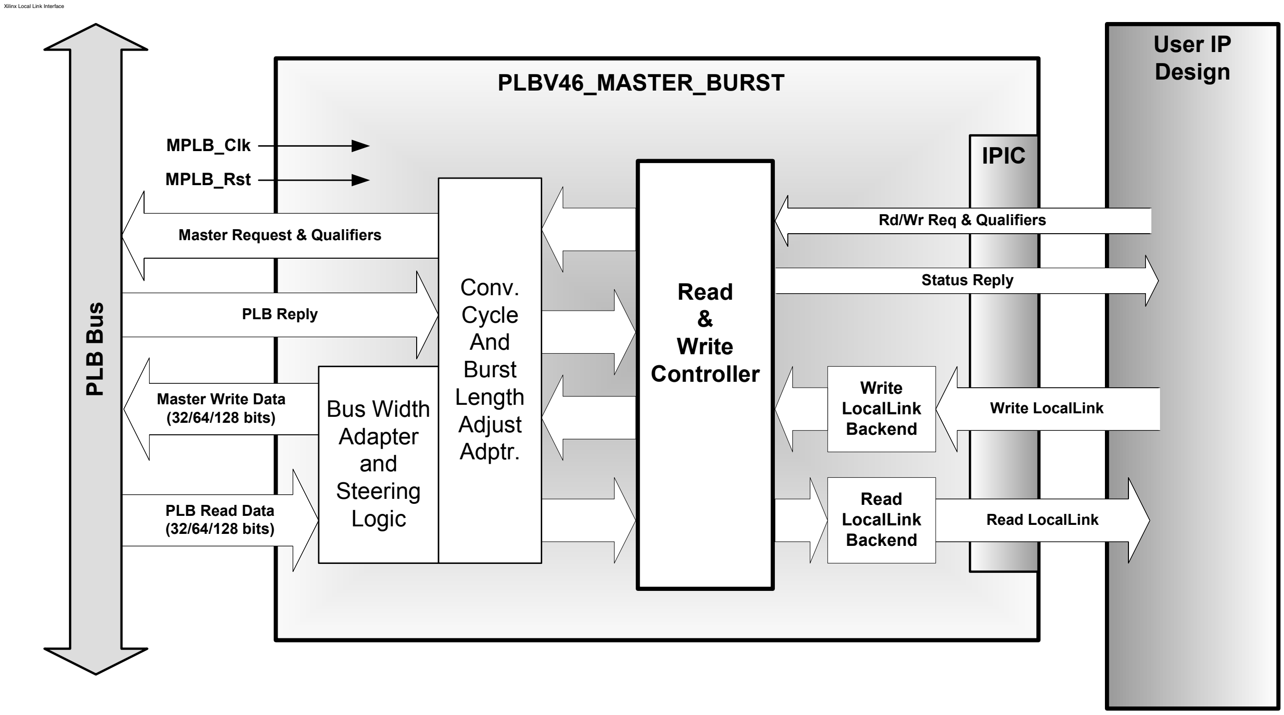

Figure 1: **PLBV46 Master Burst Block Diagram**

The Client IP receives data from and transmits data to the PLB Master via the Xilinx LocalLink Interface protocol. LocalLink is a point-to-point, synchronous interface intended for high data rate applications. Because data flow is unidirectional, the PLB Master employs two LocalLink interfaces, one for IP Client data read operations and one for IP Client data write operations.

LocalLink is based upon the concept of a Source device transmitting data to a Destination device. Data flow is unidirectional; always from the Source to the Destination. Both Source and Destination can throttle transfers as well as choose to discontinue the transfer. In order for a transfer data beat to complete., both the Source and the Destination must signal that they are ready at the rising edge of the transfer synchronization clock (clk). The Source indicates a ready condition by asserting the src_rdy_n signal. The Destination indicates ready by asserting the dst_rdy_n signal.

Data (d[n:0]) is transferred in a delimited group otherwise known as a packet. The start of a packet is delimited with the assertion of the Start-of-Frame signal (sof_n) by the Source. The assertion of End-of-Frame by the Source (eof_n) delimits the last data beat of a packet. A single data beat transfer is delimited with simultaneous assertion of sof_n and eof_n.

Transfer acknowledge/throttling is accomplished with the assertion of src_rdy_n and dst_rdy_n. De-assertion of either signal will throttle the transfer. If the Destination device can no longer transfer data or no longer needs data, it may assert the dst_dsc_n to discontinue the transfer. Conversely, the Source may terminate transmission prematurely with the assertion of the src_dsc_n signal.

Note: The current implementation of the PLBV46 Master Burst does not support discontinue assertion by the Client IP LocalLink interface. See the I/O signal descriptions of the individual discontinue signals inTable 1**.**

The rem[0:n] signal (short for remainder) is set by the Source during each data beat in which a delimiter flag is set (sof_n, sop_n, eop_n, eof_n). The value asserted specifies the valid bytes in that data beat and are somewhat application specific depending on the needs of the source and destination devices. The rem can be either an encoded value or a masked value and either active high or active low assertion levels. **For the PLBV46 Master Burst, the rem bits are always a mask representation and active low assertion levels**. Byte lane ordering follows PLB byte lane ordering.

A basic LocalLink data transfers are shown in Figure 2. The data packet consists of 16 data beats of 32 bits wide. The diagram shows both the Source and Destination throttling the transfer. In this case, the sop_n and eop_n are not shown because header and footer data is not being transmitted in the packet,

Note: Note: The Xilinx LocalLink Interface specification allows the use of either right-to-left or left-to-right bit ordering as long the Source and Destination are consistent. The PLBV46 Master Burst follows the IBM CoreConnect convention of left-to-right bit ordering and Big Endian byte ordering.

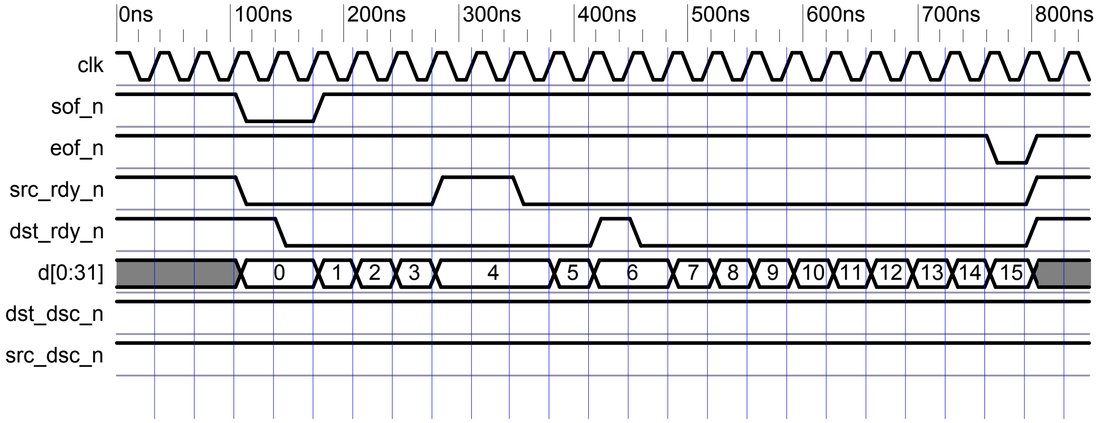

Figure 2: **Basic LocalLink Transfer**

I/O Signals

The PLBV46 Master Burst signals are listed and described in the following table.

Table 1: **PLBV46 Master Burst I/O Signal Description**

Signal Name	Interface	Signal Type	Init Status	Description
PLB Clock and Reset				
MPLB_Clk	PLB Bus	I		PLB main bus clock. See Note 1.
MPLB_Rst	PLB Bus	I		PLB main bus reset.See Note 1.
Other System Signal				
MD_error	PLB Bus	O	0	Master Detected Error Status Output (active high).
PLB Request and Qualifier Signals				
M_request	PLB Bus	O	0	See Note 1.
M_priority	PLB Bus	O	0	See Note 1.
M_buslock	PLB Bus	O	0	See Note 1.
M_RNW	PLB Bus	O	0	See Note 1.
M_BE(0:[**C_MPLB_DWIDTH**/8]-1)	PLB Bus	O	zeros	See Note 1.
M_Msize(0:1)	PLB Bus	O	00	See Note 1.
M_size(0:3)	PLB Bus	O	0000	See Note 1.
M_type(0:2)	PLB Bus	O	000	See Note 1.
M_ABus(0:**31**)	PLB Bus	O	zeros	See Note 1.
M_wrBurst	PLB Bus	O	0	See Note 1.
M_rdBurst	PLB Bus	O	0	See Note 1.
M_wrDBus(0:**C_MPLB_DWIDTH**-1)	PLB Bus	O	zeros	See Note 1.
PLB Reply Signals				
PLB_MSSize(0:1)	PLB Bus	I		Unused. See Note 2.
PLB_MaddrAck	PLB Bus	I		See Note 1.
PLB_Mrearbitrate	PLB Bus	I		See Note 1.
PLB_MTimeout	PLB Bus	I		See Note 1.
PLB_MRdErr	PLB Bus	I		See Note 1.
PLB_MWrErr	PLB Bus	I		See Note 1.
PLB_MRdDBus(0:**C_MPLB_DWIDTH**-1)	PLB Bus	I		See Note 1.
PLB_MRdDAck	PLB Bus	I		See Note 1.
PLB_MWrDAck	PLB Bus	I		See Note 1.
PLB_RdBTerm	PLB Bus	I		See Note 1.
PLB_MWrBTerm	PLB Bus	I		See Note 1.
PLB Signal Ports Included in the Design but Unused Internally				
M_TAttribute(0 to 15)	PLB Bus	O	0	Unused. See Note 2.

Table 1: **PLBV46 Master Burst I/O Signal Description**

Signal Name	Interface	Signal Type	Init Status	Description
M_lockerr	PLB Bus	O	0	Unused. See Note 2.
M_abort	PLB Bus	O	0	Unused. See Note 2.
M_UABus(0:**31)**)	PLB Bus	O	zeros	Unused. See Note 2.
PLB_MBusy	PLB Bus	I		Unused. See Note 2.
PLB_MIRQ	PLB Bus	I		Unused. See Note 2.
PLB_RdWdAddr(0:3)	PLB Bus	I		Unused. See Note 2.
IPIC Command Interface Signals				
IP2Bus_MstRd_Req	IPIC	I		User Logic Read Request.
IP2Bus_MstWr_Req	IPIC	I		User Logic Write Request.
IP2Bus_Mst_Addr(0 to C_MPLB_AWIDTH-1)	IPIC	I		User Logic Request Address. See Note 4.
IP2Bus_Mst_BE(0 to [C_MPLB_NATIVE_DWIDTH/8]-1)	IPIC	I		User Logic Request Byte Enables (only used during single data beat requests).
IP2Bus_Mst_Length(0 to 11)	IPIC	I		User Logic Request Length (bytes) for Fixed Length Burst Transfers. See Note 4.
IP2Bus_Mst_Type	IPIC	I		User Logic Request Type Indicator 0 = Single Data Beat 1 = Fixed Length Burst See Note 5.
IP2Bus_Mst_Lock	IPIC	I		Reserved: Tie to logic Low. User Logic Bus Lock Request.
IP2Bus_Mst_Reset	IPIC	I		Optional User Logic Reset. Request.
Bus2IP_Mst_CmdAck	IPIC	O	0	Command Acknowledge Status.
Bus2IP_Mst_Cmplt	IPIC	O	0	Command Complete Status.
Bus2IP_Mst_Error	IPIC	O	0	Command Error Status.
Bus2IP_Mst_Rearbitrate	IPIC	O	0	Command Rearbitrate Status. User Logic should ignore this signal.
Bus2IP_Mst_Timeout	IPIC	O	0	Command Timeout Status.
IPIC Read LocalLink Interface Signals				
Bus2IP_MstRd_d(0 to C_MPLB_NATIVE_DWIDTH-1)	IPIC	O	zeros	Read data output to User Logic.
Bus2IP_MstRd_REM(0 to [C_MPLB_NATIVE_DWIDTH/8]-1)	IPIC	O	zeros	LocalLink Remainder Indicators (Mask format, active low). See Note 6.
Bus2IP_MstRd_sof_n	IPIC	O	1	Active low signal indicating the starting data beat of a Read LocalLink transfer.

Table 1: **PLBV46 Master Burst I/O Signal Description**

Signal Name	Interface	Signal Type	Init Status	Description
Bus2IP_MstRd_eof_n	IPIC	O	1	Active low signal indicating the ending data beat of a Read LocalLink transfer.
Bus2IP_MstRd_src_rdy_n	IPIC	O	1	Active low signal indicating that the data value asserted on the Bus2IP_MstRd_d Bus is valid.
Bus2IP_MstRd_src_dsc_n	IPIC	O	1	Active low signal indicating that the Read LocalLink Source (Master) needs to discontinue the transfer. This will only be asserted if the Master encounters a PLB Timeout during the address phase of a parent or child request to the PLB.
IP2Bus_MstRd_dst_rdy_n	IPIC	I		Active low signal indicating that the data value asserted on the Bus2IP_MstRd_d Bus is being accepted by the LocalLink destination (User Logic).
IP2Bus_MstRd_dst_dsc_n	IPIC	I		Active low signal indicating that the Read LocalLink Destination (User Logic) needs to discontinue the transfer. This is currently unsupported in this Master. User Logic should tie this signal to logic high.
IPIC Write LocalLink Interface Signals				
IP2Bus_MstWr_d(0 to C_MPLB_NATIVE_DWIDTH-1)	IPIC	I		Write data input from the User Logic.
IP2Bus_MstWr_REM(0 to C_REM_WIDTH-1)	IPIC	I		LocalLink Remainder input, ignored by the PLB Master Burst. User should tie to logic 0.
IP2Bus_MstWr_sof_n	IPIC	I		Active low signal indicating the starting data beat of a Write LocalLink transfer.
IP2Bus_MstWr_eof_n	IPIC	I		Active low signal indicating the ending data beat of a Write LocalLink transfer.
IP2Bus_MstWr_src_rdy_n	IPIC	I		Active low signal indicating that the data value asserted on the IP2Bus_MstWr_d Bus is valid.
IP2Bus_MstWr_src_dsc_n	IPIC	I		Active low signal indicating that the Write LocalLink Source (User Logic) needs to discontinue the transfer. This is currently unsupported in this Master. User Logic should tie this signal to logic high.

Table 1: **PLBV46 Master Burst I/O Signal Description**

Signal Name	Interface	Signal Type	Init Status	Description
Bus2IP_MstWr_dst_rdy_n	IPIC	O	1	Active low signal indicating that the data value asserted on the IP2Bus_MstWr_d Bus is being accepted by the LocalLink destination (Master).
IP2Bus_MstWr_dst_dsc_n	IPIC	O	1	Active low signal indicating that the Write LocalLink Destination (Master) needs to discontinue the transfer. This will only be asserted if the Master encounters a PLB Timeout during the address phase of a parent or child request to the PLB.

Note 1

This signal's function and timing is defined in the IBM® 128-Bit Processor Local Bus Architecture Specification Version 4.6.

Note 2

Output ports that are not used are driven to constant logic levels that are consistent with the inactive state for the subject signal. Input ports that are required but not used are internally ignored by the design.

Note 3

For fixed length burst requests, the starting address for the request as specified by the IP2Bus_Mst_Addr(0:31) input must be aligned on an address boundary matching the C_MPLB_NATIVE_DWIDTH value.

Note 4

The request length is specified in bytes and must be a multiple of C_MPLB_NATIVE_DWIDTH/8.

Note 5

The requested data transfer width for a fixed length burst request will be automatically set to the native data width of the Master which is assigned with the C_MPLB_NATIVE_DWIDTH parameter.

Note 6

The PLBV46 Master Burst only supports Mask representation (as opposed to encoded representation) for values on the LocalLink REM buses. In addition, the REM values must be asserted active low.

Design Parameters

The PLBV46 Master Burst provides for User interface tailoring via VHDL Generic parameters. These parameters are detailed in the following table. The FPGA Family Type parameter is used to select the target FPGA family type. Currently, this design supports Virtex-4, Virtex-5, and Spartan-3 family of devices.

Table 2: **PLBV46 Master Burst Design Parameters**

Feature/Description	Parameter Name	Allowable Values	Default Values	VHDL Type
PLB I/O Specification				
Specifies the Number of Used Address bits out of the available 64 bits of PLBV46 addressing	C_MPLB_AWIDTH	32	32	integer
Width of the PLB Data Bus to which the Master is attached	C_MPLB_DWIDTH	32, 64, 128	32	integer
Specifies the internal native data width of the Master	C_MPLB_NATIVE_DWIDTH	32, 64, 128	32	integer
Narrow Slave Support				
Indicates the smallest Native Data Width of any Slave attached to the PLBV46 Bus used by the Master (1)	C_MPLB_SMALLEST_SLAVE	32, 64, 128	32	integer
This parameter is used to override the automatic inclusion of the Conversion Cycle and Burst length Expansion logic (1)	C_INHIBIT_CC_BLE_INCLUSION	0, 1 0 = Allow automatic inclusion of CC and BLE logic 1 = Inhibit automatic inclusion of the CC and BLE logic	0	integer
FPGA Family Type				
Xilinx FPGA Family	C_FAMILY	virtex4,virtex5, spartan3a, aspartan3a, spartan3, aspartan3, spartan3e, aspartan3e, spartan3adsp, aspartan3adsp	virtex5	string

Note: If the Master is parameterized to have 64 or 128 bit Native Data Width and it potentially can access a Slave that is narrower than the requested data transfer size by the Master (indicated by the C_MPLB_SMALLEST_SLAVE parameter value), then Conversion Cycle and Burst Length Expansion logic is required by the Master to complete the transfer. Masters that are parameterized to 32-bit Native Data Width do not need the logic regardless of target Slave data width.

Allowable Parameter Combinations

The current implementation of the PLBV46 Master Burst has the following restrictions that apply to parameter value settings:

- The assigned value for C_MPLB_AWIDTH is currently restricted to 32.
- The value of C_MPLB_DWIDTH must be greater than or equal to the value assigned to C_MPLB_NATIVE_DWIDTH.

Parameter - Port Dependencies

Table 3: **PLBV46 Master Burst Parameter-Port Dependencies**

Name (Generic or Port)	Affects (Port)	Depends (Generic)	Relationship Description
Design Parameters			
C_MPLB_AWIDTH	IP2Bus_Mst_Addr		The Parameter directly sets the ports width.
C_MPLB_DWIDTH	M_BE		The BE Bus width is derived from the parameter value by dividing it by 8.
C_MPLB_DWIDTH	M_wrDBus		The port width is directly set by the parameter value.
C_MPLB_DWIDTH	PLB_MRdDBus		The port width is directly set by the parameter value.
C_MPLB_NATIVE_DWIDTH	IP2Bus_Mst_BE		The IPIC BE Bus width is derived from the parameter value by dividing it by 8.
C_MPLB_NATIVE_DWIDTH	Bus2IP_MstRd_d		The IPIC Read LocalLink Data port width is directly set by the parameter value.
C_MPLB_NATIVE_DWIDTH	IP2Bus_MstWr_d		The IPIC LocalLink Write Data port width is directly set by the parameter value.
C_MPLB_NATIVE_DWIDTH	Bus2IP_MstRd_REM		The IPIC Read LocalLink REM port width is derived from the parameter value by dividing it by 8.
C_MPLB_NATIVE_DWIDTH	IP2Bus_MstWr_REM		The IPIC Write LocalLink REM port width is derived from the parameter value by dividing it by 8.

Parameter Detailed Descriptions

C_MPLB_AWIDTH

This integer parameter is used by the PLBV46 Master Burst to size internal address related components and the input address from the User logic on the Command Interface. The parameter is provided for future growth beyond 32-bit addressing. Currently, the parameter value is only allowed to be set 32.

C_MPLB_DWIDTH

This integer parameter is used by the PLBV46 Master Burst to size and optimize the PLBV46 data bus interface logic. This value should be set to match the actual width of the PLBV46 bus, 32, 64 or 128-Bits.

C_MPLB_NATIVE_DWIDTH

This integer parameter is used to specify the internal data width of the PLBV46 Master Burst as well as the IPIC data width to the User Logic. The parameter may be set to 32, 64, or 128.

C_MPLB_SMALLEST_SLAVE

This parameter is defined as an integer and is set to the smallest Native Data Width of any Slave that is attached to the same PLBV46 bus as the Master. Allowed values are 32, 64, and 128. The parameter is used when the Master is parameterized with a Native Data Width of 64 or 128 bits. If the value of the C_MPLB_SMALLEST_SLAVE is less than the Native Data Width of the Master, then Conversion Cycle and Burst Length Expansion logic is automatically included in the Master’s implementation.

C_INHIBIT_CC_BLE_INCLUSION

This parameter is used to inhibit the automatic inclusion of the Conversion Cycle and Burst Length Expansion logic if it is known by the User that the Master will not be accessing the narrower Slaves or the requested transfer widths for any access will not exceed the Native Data Width of any targeted Slave.

C_FAMILY

This parameter is defined as a string. It specifies the target FPGA technology for implementation of the PLB Slave. This parameter is required for proper selection of FPGA primitives. Currently, the PLBV46 Master Burst does not implement any FPGA primitives that require the use of this parameter.

IPIC Transaction Timing

The following section shows timing relationships for PLBV46 and IPIC interface signals during read and write transfers. Single data beat and Fixed Length Burst transfers are shown.

Single Data Beat Read Operation

Two single beat read cycles are shown in Figure 3. The Master has a Native Data Width of 32 bits and the PLB data width is 32 bits. The first cycle shows the PLB Slave address and data acknowledging the read cycle at the earliest allowed times by the PLB specification. The second read transfer indicates a more typical address acknowledge sequence and a delayed read data acknowledge by the PLB Slave device.

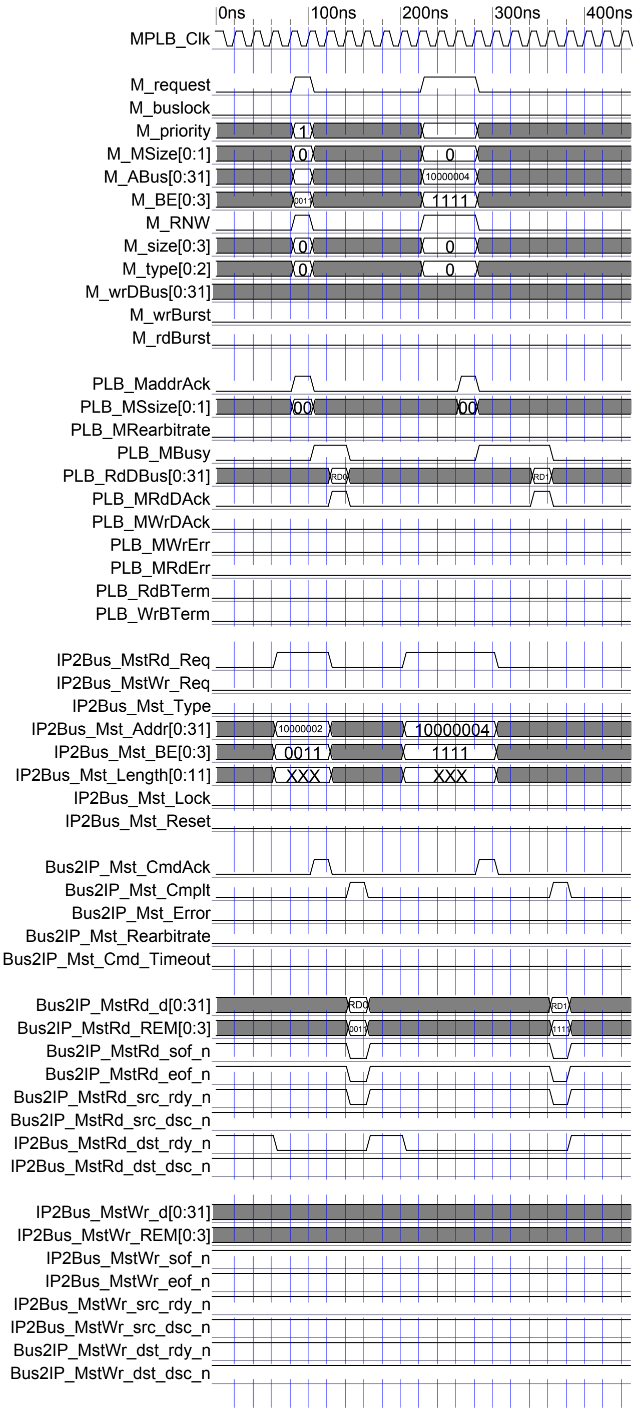

Figure 3: **PLB Single Data Beat Read Timing**

Single Data Beat Write Operation

Two single beat write cycles are shown in Figure 4. The Master has a Native Data Width of 32 bits and the PLB data width is 32 bits. The first cycle shows the PLB Slave address and data acknowledging the write cycle at the earliest allowed times by the PLB specification. The second write transfer indicates a more typical address acknowledge sequence and a delayed write data acknowledge by the PLB Slave device.

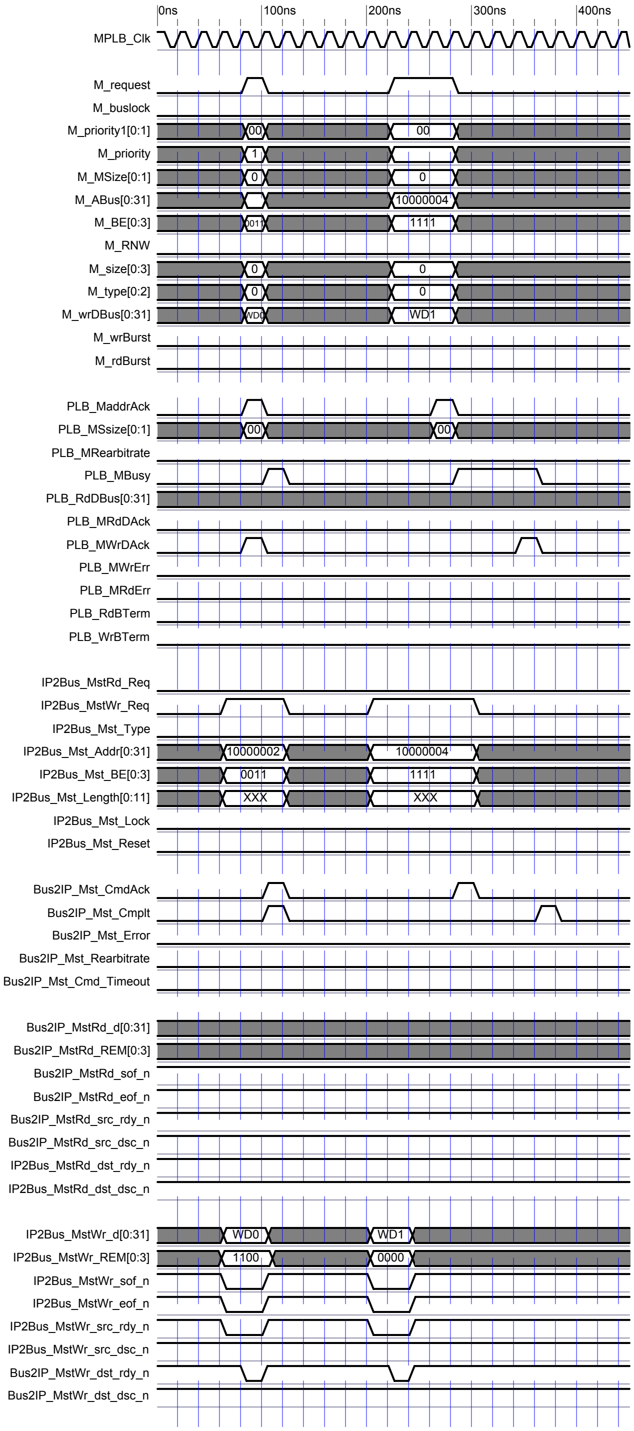

Figure 4: **PLB Single Data Beat Write Timing**

Single Data Beat Read With Error

Single data beat Read transfers with a Slave reported error is shown in Figure 5. The Master has a Native Data Width of 32 bits and the PLB data width is 32 bits. For both transfers a Slave data error is reported and the Master's MD_Error output is asserted and held. The first assertion of MD_Error is cleared by the IP2Bus_Mst_Reset input from the IPIC interface. The second assertion of MD_Error is cleared by the MPLB_Rst input from the PLBV46 interface.

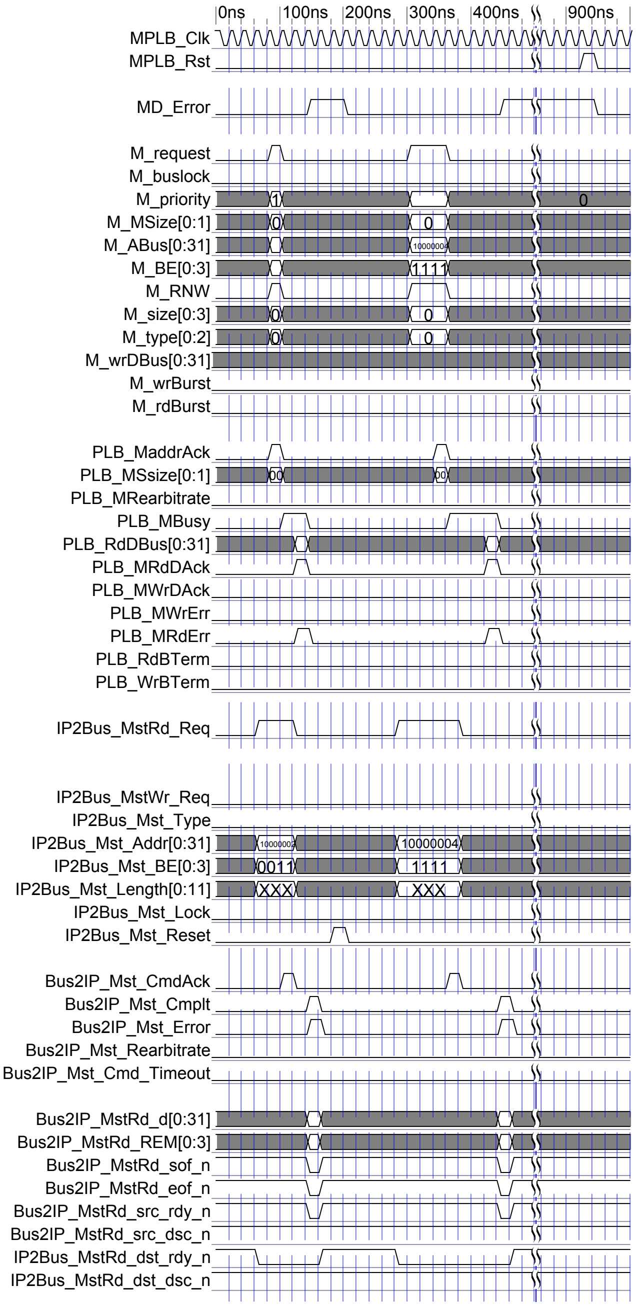

Figure 5: **PLB Single Data Beat Read Error Timing**

Single Data Beat Write With Error

Two single beat write cycles are shown in Figure 6. The Master has a Native Data Width of 32 bits and the PLB data width is 32 bits. For both transfers, a Slave data error is reported and the Master's MD_Error output is asserted and held. The first assertion of MD_Error is cleared by the IP2Bus_Mst_Reset input from the IPIC interface. The second assertion of MD_Error is cleared by the MPLB_Rst input from the PLBV46 interface.

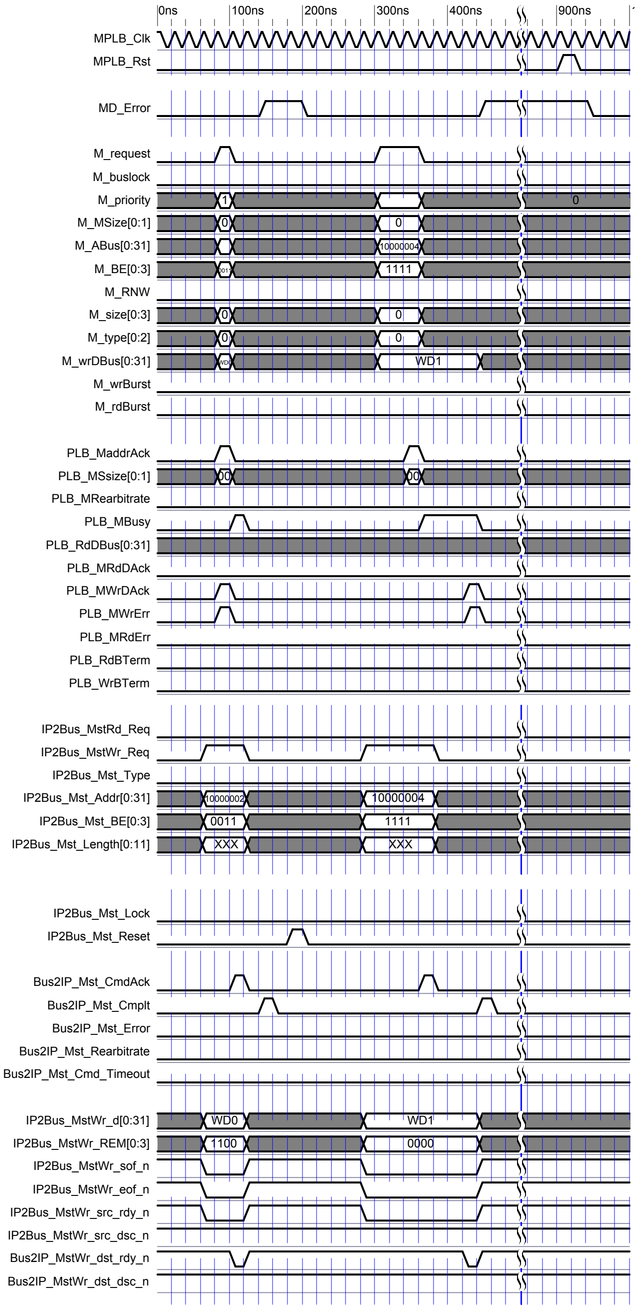

Figure 6: **PLB Single Data Beat Write Error Timing**

Fixed Length Burst Read Operation

A fixed length burst read is shown in Figure 7. The Master has a native data width of 32-bits and the PLB data width is 32-bits. The burst length is set to 32 (0x20) bytes, which results in eight beats of 32-bit data transferred across the IPIC and PLB interfaces. The burst is address aligned to at least a 32-bit word, and, consequently, the Bust2IP_MstRd_REM is set to 0's. The IP2Bus_Mst_BE port is ignored during bursts.

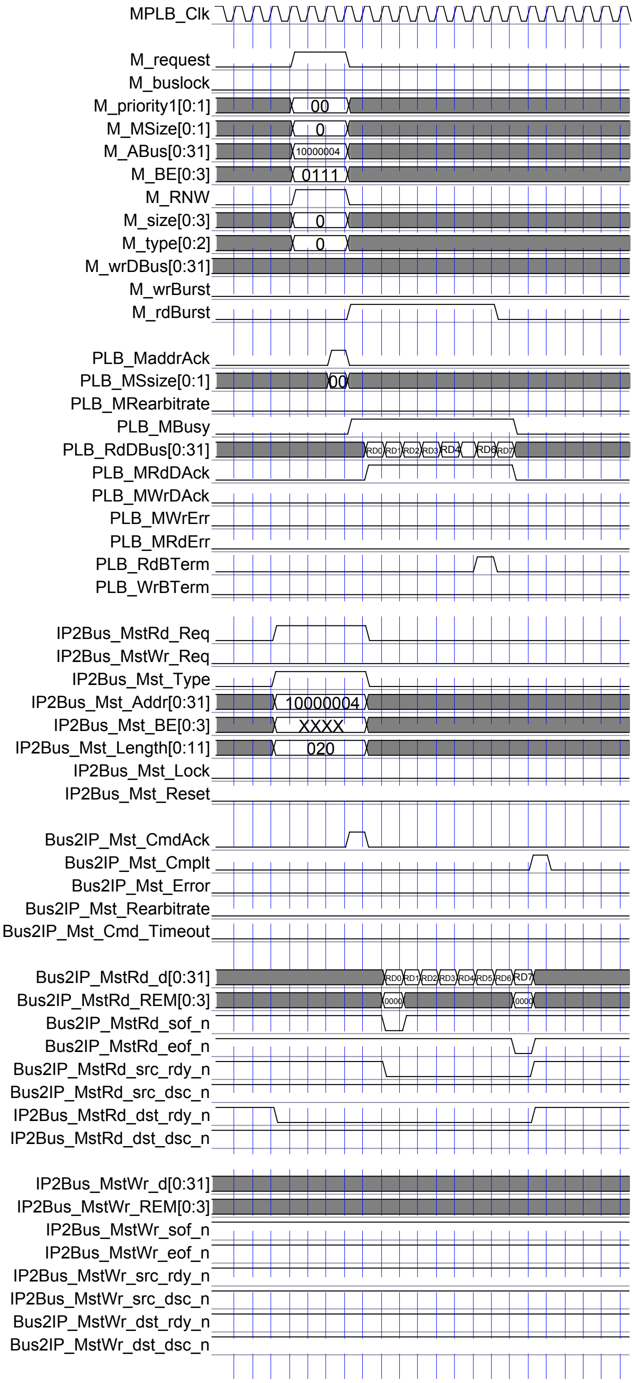

Figure 7: **Fixed Length Burst Read Timing**

Fixed Length Burst Write Operation

A fixed length burst write is shown in Figure 8. The Master has a native data width of 32-bits and the PLB data width is 32-bits. The burst length is set to 32 (0x20) bytes, which results in eight beats of 32-bit data transferred across the IPIC and PLB interfaces. The burst is address-aligned to at least a 32-bit word, and, consequently, the Bust2IP_MstWr_REM is set to 0's. The IP2Bus_Mst_BE port is ignored during bursts. The Bus2IP_MstWr_dst_rdy_n throttles the incrementing of write data due to delays on the PLB bus.

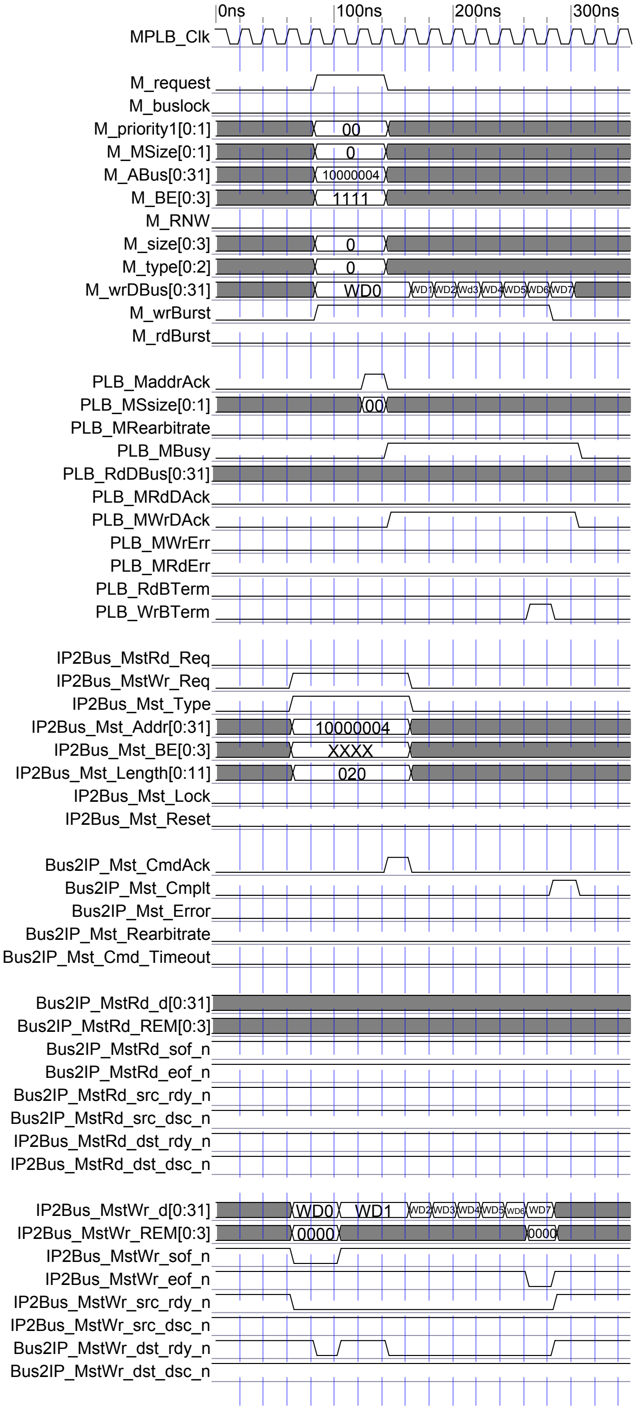

Figure 8: **Fixed Length Burst Write Timing**

Read Address Phase Timeout

An attempted single data beat read operation that results in an Address Phase timeout is shown in Figure 9. The Master has a Native Data Width of 32 bits and the PLB data width is 32 bits. The Master's MD_Error output is asserted and held upon detection of the PLB_MTimeout assertion. The Master's request, address, and qualifiers are removed from the PLB on the following PLB clock after the timeout indication and a timeout status is relayed to the User logic on the IPIC. The associated LocalLink interface will be forced to Discontinue by the Master. For this example, the assertion of MD_Error is cleared by the MPLB_Rst input from the PLBV46 interface.

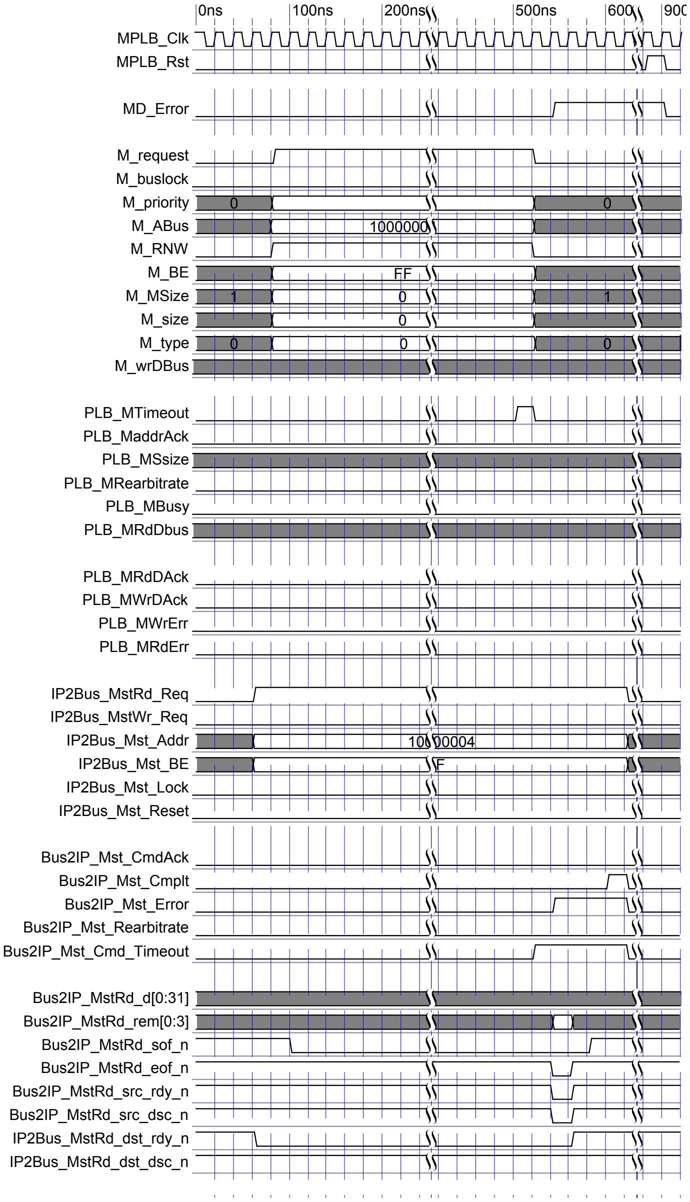

Figure 9: **Single Data Beat Read Resulting in PLB Address Phase Timeout**

Write Address Phase Timeout

An attempted single data beat write operation that results in an Address Phase timeout is shown in Figure 10. The Master has a Native Data Width of 32 bits and the PLB data width is 32 bits. The Master's MD_Error output is asserted and held upon detection of the PLB_MTimeout assertion. The Master's request, address, and qualifiers are removed from the PLB on the following PLB clock after the timeout indication and a timeout status is relayed to the User logic on the IPIC. The associated LocalLink interface (if it is still active) will be forced to Discontinue by the Master. For this example, the assertion of MD_Error is cleared by the MPLB_Rst input from the PLBV46 interface.

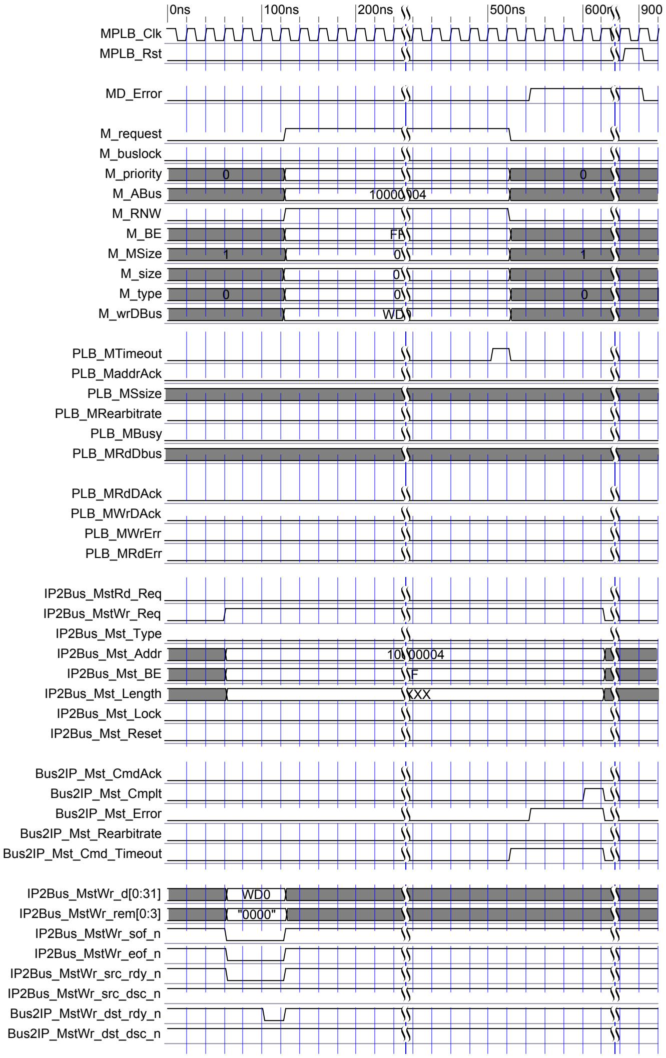

Figure 10: **Single Data Beat Write Resulting in PLB Address Phase Timeout**

Read Conversion Cycle

A single data beat read operation that requires Conversion Cycles is shown in Figure 11. The Master's Native Data Width is set to 128 and it attempts to read 16 bytes from a 32-bit Slave. This scenario requires the Master to initiate 3 Conversion Cycles to complete the needed data transfer.

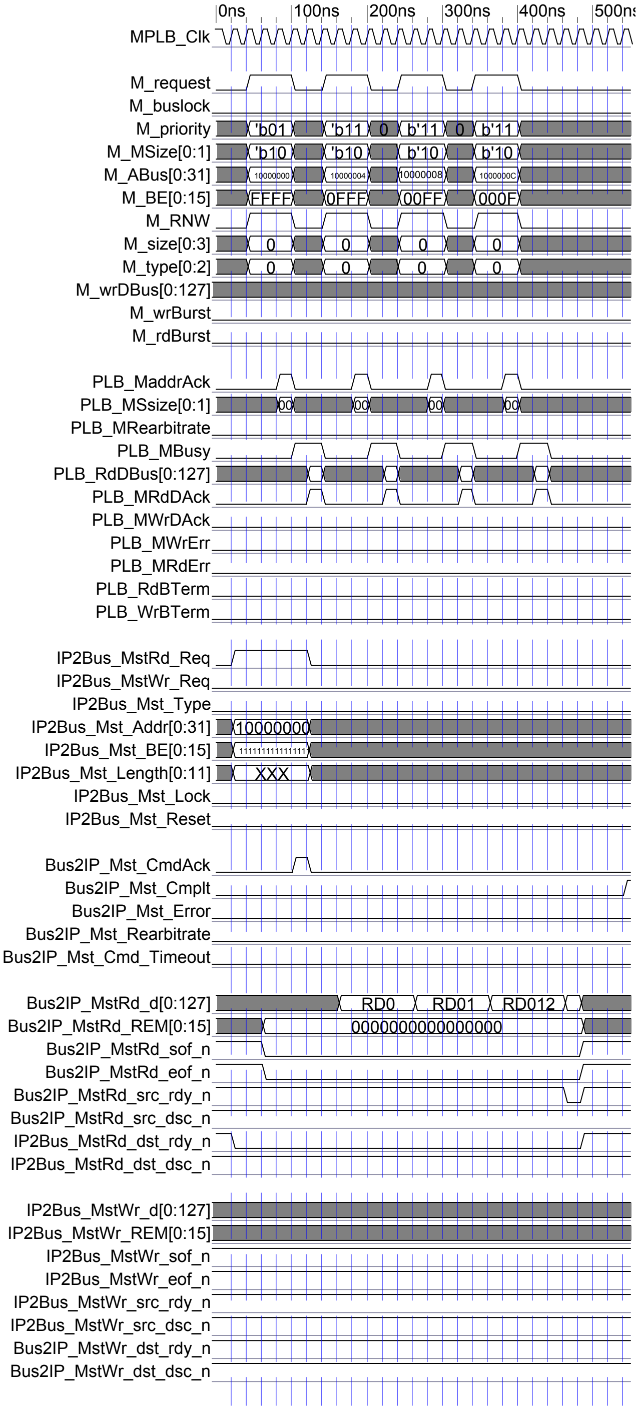

Figure 11: **Single Data Beat Read Resulting in Conversion Cycles**

Write Conversion Cycle

A single data beat write operation that requires Conversion Cycles is shown in Figure 12. The Master's Native Data Width is set to 128 and it attempts to write 13 bytes to a 32-bit Slave. This scenario requires the Master to initiate 3 Conversion Cycles to complete the needed data transfer.

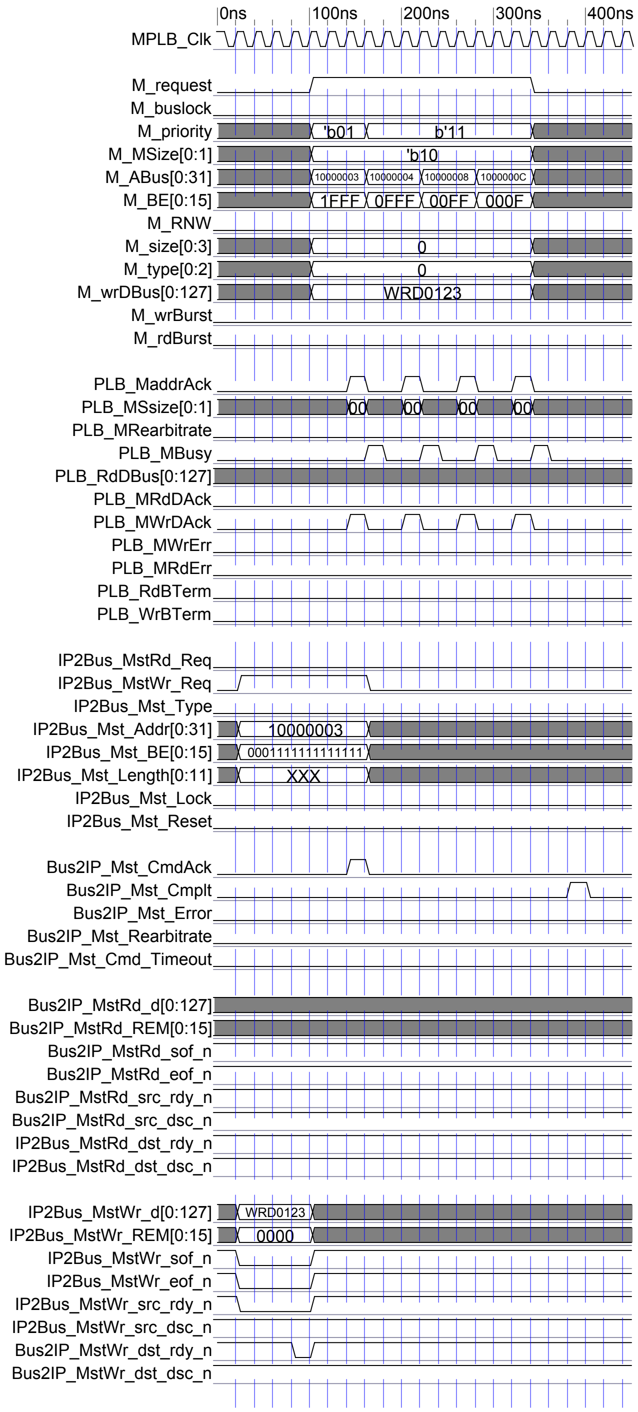

Figure 12: **Single Data Beat Write Resulting in Conversion Cycles**

Burst Read With Burst Length Expansion

A Fixed Length Burst read of 4 Quad-words that requires Burst length Expansion is shown in Figure 13. The Master′s Native Data Width is set to 128 and it attempts to burst read 4 quad words from a 32-bit Slave. This scenario requires the Master (and the target Slave) to expand the required number of PLB data beats from 4 to 16 to complete the requested data transfer. The Master will collect 4 words from the PLB (that is all the 32-bit Slave provides per data beat) and then transfers a full quad word via the Read LocalLink Interface. This behavior repeats until the transfer is completed.

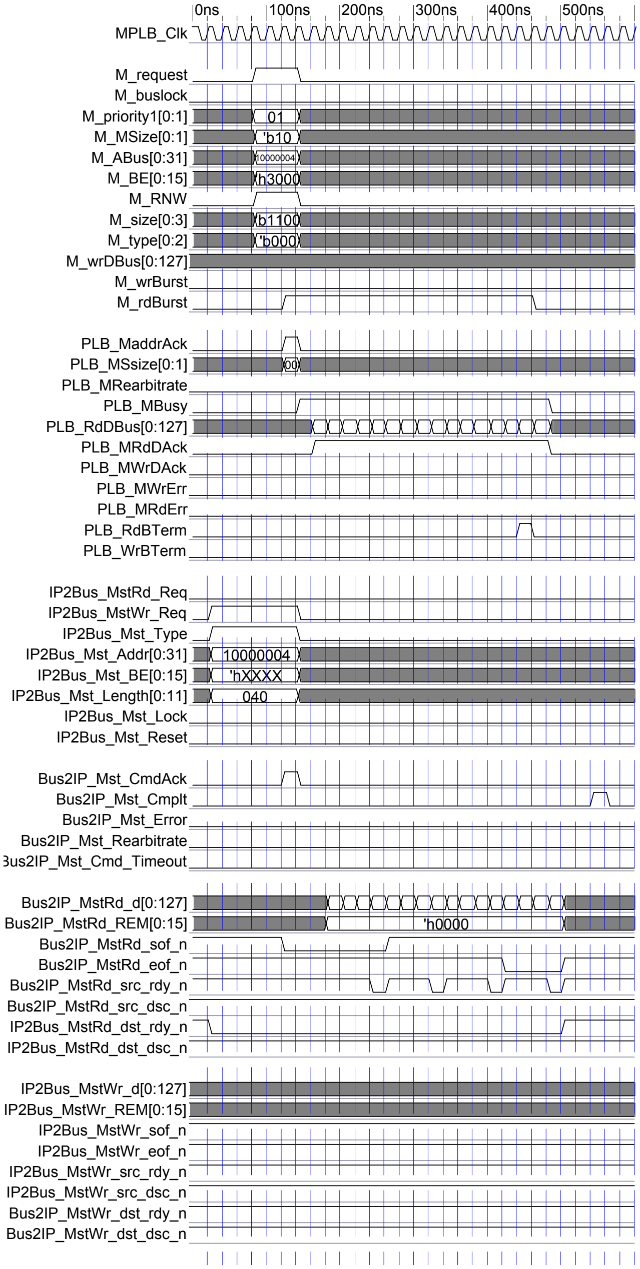

Figure 13: **Burst Read Resulting in Burst Length Expansion**

Burst Write With Burst Length Expansion

A Fixed Length Burst write of 4 Quad-words that requires Burst length Expansion is shown in Figure 14. The Master′s Native Data Width is set to 128 and it attempts to burst write 4 quad words to a 32-bit Slave. This scenario requires the Master (and the target Slave) to expand the required number of PLB data beats from 4 to 16 to complete the requested data transfer. The Master will collect a quad word from the Write LocalLink and then output 4 data beats (1 word per data beat) onto the PLB (that is all the 32-bit Slave will consume per data beat). This behavior repeats until the transfer is completed.

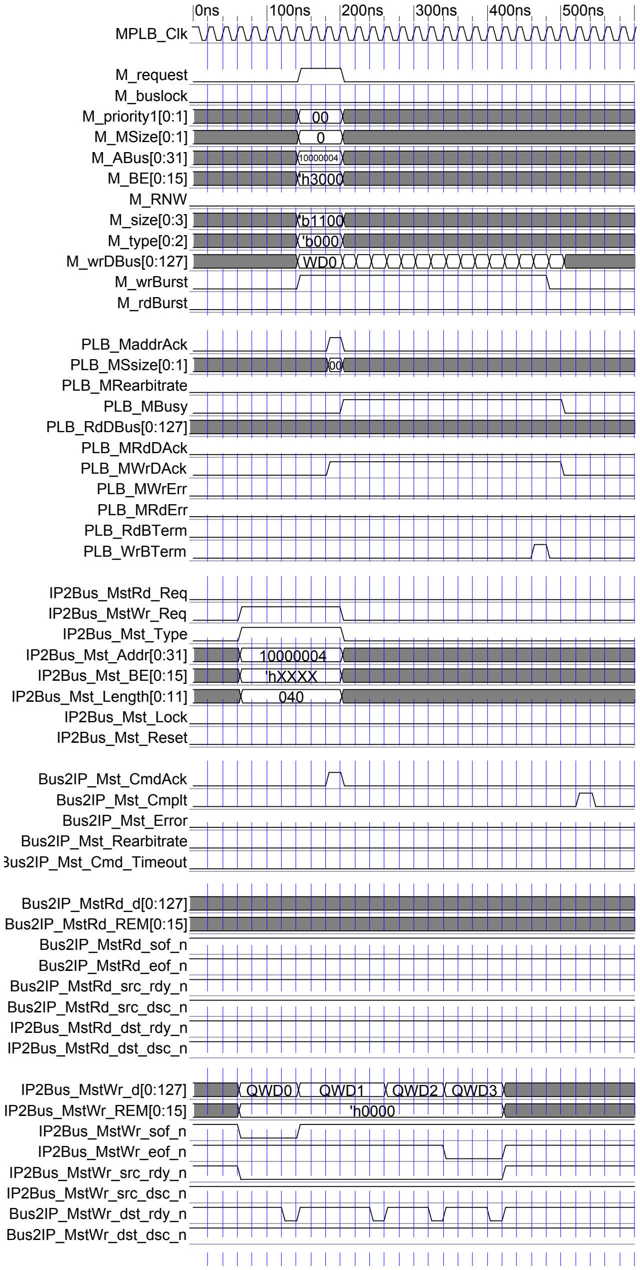

Figure 14: **Burst Write Resulting in Burst Length Expansion**

Register Descriptions

The PLBV46 Master Burst has no user accessible registers.

User Application Topics

The Command Interface

The PLBV46 Master Burst is controlled by the Client IP Logic via the command interface. The command interface provides the mechanism to request a data transfer and the status of the transfer. The actual data transfer is performed on the Read and Write LocalLink interfaces. The command interface consists of the following:

- Inputs to the Master Service (from IP Client)
    - IP2Bus_MstRd_Request (Read Request
    - IP2Bus_MstWr_Request (Write Request)
    - IP2Bus_Mst_Addr(0:31) (Starting PLB address)
    - IP2Bus_Mst_Length(0:11) (Transfer Length limit in bytes)
    - IP2Bus_Mst_BE(0:3) (Byte Enable designator for single data beat requests)
    - IP2Bus_Mst_Type (specifies single or fixed length burst request)
    - IP2Bus_Mst_Lock (PLB Bus Lock qualifier)
    - IP2Bus_Mst_Reset (Forces a synchronous reset of the Master logic)
- Status outputs from the Master to the IP Client
    - Bus2IP_Mst_CmdAck (Initial request to the PLB has been Command Acknowledge
    - Bus2IP_Mst_Cmplt (Command Completion indication)
    - Bus2IP_Mst_Error (Command Error)
    - Bus2IP_Mst_Rearbitrate (Command PLB Rearbitrate)
    - Bus2IP_Mst_Timeout (Request has timed out on the PLB)

The Command Interface protocol requires that the IP Client drive the IP2Bus_MstRd_Request or the IP2Bus_MstWr_Request and the associated qualifiers until the Bus2IP_Mst_CmdAck is received from the Master. Upon the receipt of the Bus2IP_Mst_CmdAck, the IP Client must deassert the request and optionally the associated qualifiers. If a PLB Timeout occurs during the initial PLB Address Phase of a request, the Bus2IP_Mst_CmdAck assertion will not occur. Instead the Master will reply with the Bus2IP_Mst_Timeout status asserted in conjunction with assertion of the Bus2IP_Mst_Cmplt status and the Bus2IP_Mst_Error. This is an indication to the IP Client that the address of the request did not match any assigned address range of a PLB Slave or the type of request was not support by the target Slave in the system(i.e. A single data beat only Slave will not respond to a burst request). If a Bus2IP_Mst_CmdAck is asserted by the Master, then the Master has successfully negotiated a PLB Address Phase with the starting address and command qualifiers and the corresponding data phase is in progress. When the PLB Data Phase completes, the Master will assert the Bus2IP_Mst_Cmplt signal. If a Data Phase error is received from the PLB Slave by the Master, the Bus2IP_Mst_Error status will be also be asserted when the Bus2IP_Mst_Cmplt is asserted. The duration of the Bus2IP_Mst_Cmplt assertion is one PLB clock period.

Transfer Length Limitations

IP2Bus_Mst_Length(0:11) signal in the Master's Command Interface qualifiers allows the user to specify a read or write burst transfer length in bytes (length is ignored for Single data beat commands). The length specified must be in increments of the Native Data Width of the Master (C_MPLB_DWIDTH/8) and cannot exceed a 12 bit representation. So the following **maximum** transfer lengths are allowed:

- C_MPLB_DWIDTH = 32, then maximum length is 4092 bytes (4096 - 4)
- C_MPLB_DWIDTH = 64, then maximum length is 4084 bytes (4096 - 8)
- C_MPLB_DWIDTH = 128, then maximum length is 4080 bytes (4096 - 16)

PLB Burst operations require a minimum of 2 data beats (otherwise Single data beat transfers must be used). Thus, the minimum transfer length for burst transfers is dependent upon the Native Data Width of the Master. The following **minimum** transfer lengths are required:

- C_MPLB_DWIDTH = 32, then minimum length is 8 bytes [(32/8) * 2]
- C_MPLB_DWIDTH = 64, then minimum length is 16 bytes [(64/8) * 2]
- C_MPLB_DWIDTH = 128, then minimum length is 32 bytes [(128/8) * 2]

Considerations for Request Spawning

One important aspect of the Master operation must be kept in mind by the User is the ability of the Master to spawn multiple child PLB requests when mechanizing a single request from the Client IP. This can occur during burst requests when the requested transfer length specified by the IP2Bus_Mst_Length qualifier exceeds 16 data beats times the native byte width of the Master. For a 32-bit Master, this boundary is 64 bytes. Thus, the Bus2IP_Mst_CmdAck sent by the Master is a result of the completion of the Address Phase of the initial request posted to the PLB by the Master. Ensuing Address Phases of spawned requests will not have an associated Bus2IP_Mst_CmdAck reply to the IP Client. It is also possible that an Bus2IP_Mst_Timeout can occur after the receipt of Bus2IP_Mst_CmdAck if spawned request address is outside of the mapped address range of a PLB Slave. This would result in a PLB Timeout condition that in an error status assertion on the status signals of the Command Interface.

LocalLink Interface Considerations when Using PLBV46 Master Burst

Special LocalLink interface requirements are in play with the PLBV46 Master Burst usage. The design does not utilize FIFOs to isolate the IP Client from the PLB operations. This approach minimizes resource utilization in the Master but requires the IP Client to adhere to some operational restrictions during LocalLink transfers. The PLBv46 Master Burst ignores the IP2Bus_MstWr_MEM inputs. The user should drive these inputs to logic 0.

PLB Command Hold Off

The Master Burst will not initiate a transfer request onto the PLB until the User IP has the associated LocalLink interface in the ready state. This hold off will result in the Bus2IP_Mst_CmdAck not being asserted if the User IP logic does not put the associated LocalLink in the ready state. For a read operation, this means that the IP2Bus_MstRd_dst_rdy_n signal is asserted, signalling that the IP is ready to receive the read data. For a write, the IP2Bus_MstWr_src_rdy_n signal must be asserted and the first write data value must be present on the IP2Bus_MstWr_d(0:n) bus.

Transfer Throttling

The main rule that must be followed is the limitation that once a LocalLink transfer is started, the IP Client is not allowed to throttle the LocalLink transfer by de-asserting the applicable ready signal or asserting the discontinue signal. The reason this is required is that without the FIFO's, the PLB Master will directly couple PLB Data phase protocol to the LocalLink transfer protocol with only minor translation. One of the rules of PLB is that a PLB Master is not allowed to throttle a data phase operation once it starts. Thus, since the PLB and LocalLink are almost directly coupled, the "no throttling" rule gets pushed to the LocalLink transfer and to the IP Client.

Conversely, the PLB Slave device is allowed to throttle the PLB Data phase at any time. With the PLB and the LocalLink directly coupled during the data phase, a LocalLink transfer may be throttled at any time by the Master side as a result of PLB Slave throttling. This is generally not a hard environment for the IP Client on the Read LocalLink interface. However, in the write LocalLink direction, the IP Client logic must be able to provide the next sequential data beat of data to transfer when a PLB Slave has throttled the write transfer and then stops throttling. This may require design forethought especially if the IP Client is sourcing data from a memory element that has a read latency (such as BRAM) or a FIFO that is reading ahead and has to recover at the throttle condition.

Transfer Termination

All LocalLink transfers must terminate with the assertion of the eof_n delimiter and the simultaneous assertion of the src_rdy_n and the dst_rdy_n signals. In the case of the Read LocalLink interface, the Master will assert the last read data value from the PLB on the Bus2IP_MstRd_d(0:n), and assert Bus2IP_MstRd_src_rdy_n and the Bus2IP_MstRd_eof_n. This state will continue until the IP asserts the IP2Bus_MstRd_dst_rdy_n (this should already be asserted). In the case of the Write LocalLink, the User logic must assert the last write data value for the PLB on the IP2Bus_MstWr_d(0:n), and assert IP2Bus_MstWr_src_rdy_n and the IP2Bus_MstWr_eof_n. The Master will assert the Bus2IP_MstWr_dst_rdy_n when the data beat has been consumed by the PLB. Note that for a single data beat transfer, the LocalLink transfer will only be one data beat with the simultaneous assertion of the sof_n in conjunction with the other termination signaling.

When the Master recognizes that the LocalLink transfer has completed, the BUS2IP_Mst_CmdCmplt will be asserted for one PLB clock cycle, indicating the Master has finished all processing for the command and is ready for the next command.

PLB Rearbitration

The User Logic should ignore the Bus2IP_Mst_Rearbitrate status reply. This signal has no operational ramifications for the User logic.

Conversion Cycle Operations

Conversion Cycles are required if the Master attempts to read or write data from/to a PLB Slave if the following conditions apply:

- The Master is requesting a Single Data Beat transfer.
- The Slave has a Native Data Width that is narrower than the Master's Native Data Width. The Slave will report it's Native Data Width to the Master via the Sl_Ssize(0:1) output signals during the PLB clock period that includes the assertion of the Sl_AddrAck signal. This is independent of the PLB data width.
- The Master's assertion of the M_BE(0:n) signals indicates an attempt to transfer more data bytes than the Slave can read or write in a single data beat or the alignment of the asserted BE cross the Native Data Width boundary of the target Slave.

Burst Length Expansion

Burst length Expansion is required when a Master attempts to perform a Fixed Length Burst operation where the requested data transfer width (M_size(0:3)) is wider than the Native Data Width of the target Slave. In this case, both the Master and the Slave must automatically adjust the number of data beats required on the PLB to transfer the requested data quantity at the Slave's Native Data Width per data beat.

Conversion Cycle and Burst Length Expansion Logic Inclusion

Conversion Cycle and Burst Length Expansion support logic is automatically included in the Master's implementation if the parameter C_MPLB_SMALLEST_SLAVE is assigned a value that is less than the assigned value for the C_MPLB_NATIVE_DWIDTH parameter and the parameter C_INHIBIT_CC_BLE_INCLUSION is left at the default value of 0. However, this logic is resource intensive and can significantly increase the size of the Master and decrease its Fmax capability. If the User can guarantee that the Master will never access a target Slave in a manner that will require Conversion Cycles or Burst Length Expansion, then the automatic inclusion of the Conversion Cycle logic can be overridden by setting the C_INHIBIT_CC_BLE_INCLUSION parameter to a value of 1.

IP Master Bus Locking

This Master does not currently support PLB Bus Lock. User must tie the IP2Bus_Mst_Lock input port to logic low.

Design Implementation

Target Technology

The intended target technology is a Spartan or Virtex FPGA.

Device Utilization and Performance Benchmarks

Since the PLBV46 Master Burst is a module that will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section should be considered engineering estimates. As the PLBV46 Master Burst is combined with other pieces of the User FPGA design, the utilization of FPGA resources and timing will vary from the results reported here.

The resource utilization of this version of the PLBV46 Master Burst is shown in Table 4 for currently changeable parameter configurations. The design resource utilization numbers are taken from the resource utilization section of the Xilinx ISE MAP report that is created by the MAP tool. Default synthesis and implementation properties were utilized.

The PLBV46 Master Burst benchmarks are shown in the following table for a Virtex-4 xc4vlx200-11ff1513 FPGA.

Table 4: **PLBV46 Master Burst FPGA Performance and Resource Utilization Benchmarks**

Parameter Values[1]			Device Resources			f_{MAX}[2]
C_MPLB_DWIDTH	C_MPLB_NATIVE_DWIDTH	C_MPLB_SMALLEST_SLAVE	Slices	Slice Flip-Flops	4-input LUTs	f_{MAX}[2] (MHz)
128	128	32	376	194	645	150
128	128	64	376	194	645	150
128	128	128	221	122	379	231
128	64	32	306	177	505	205
128	64	64	182	111	301	239
128	32	32	161	107	263	239
64	64	32	308	177	505	205
64	64	64	182	111	301	239
64	32	32	161	107	263	239
32	32	32	161	107	263	239

Note:

1. C_MPLB_AWIDTH fixed at 32 and C_INHIBIT_CC_BLE_INCLUSION fixed at 0. The setting of C_FAMILY is for Virtex-5 device family.
2. Fmax represents the maximum estimated frequency of the PLBV46 Master Burst in a standalone configuration as reported by ISE XST. The actual maximum frequency will depend on the entire system and may be greater or less than what is recorded in this table.

Specification Exceptions

The following High Level PLB features are **not** supported by the PLBV46 Master Burst design.

- Bus Slave
- Split Bus Transactions
- Command Abort
- Indeterminate Length Bursts
- Cachelines
- Bus Locking

Reference Documents

The following documents contain reference information important to understanding the PLBV46 Master Burst design.

1. IBM CoreConnect128-Bit Processor Local Bus, Architectural Specification (v4.6).
2. Xilinx SP026 PLBV46 Interface Simplifications.

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.

Revision History

Date	Version	Revision
9/21/06	1.0	Initial Xilinx release.
01/03/07	1.1	Added native data Width options of 64 and 128 bits; added Conversion Cycle and Burst Length Expansion functionality to supported features
7/20/07	1.2	Added SP026 to Reference Documents List; removed Bus Lock as a supported feature.
01/07/08	1.3	Corrected minor issues with various timing diagrams; added additional text in the User Applications Topics; clarified some Features statements
4/21/08	1.4	Added Automotive Spartan-3E, Automotive Spartan-3A, Automotive Spartan-3, and Automotive Spartan-3A DSP support.
5/14/08	1.5	Updates to reflect v1_01_a of the core.
12/02/08	1.6	Added clarifications for LocalLink signals and PLBV46 Master Burst usage per CR478853. Corrected a signal name in Table 1 from Bus2IP_MstRd_dst_dsc_n to Bus2IP_MstWr_dst_dsc_n.
5/26/10	1.7	Incorporated CR547389; added clarification about the IP2Bus_MstWr_REM not being used by this core.
6/23/10	1.8	Updated to add Virtex-6 and Spartan-6 as supported devices.
12/14/10	2.0	Text updates per CR 567030.