LogiCORE IP AXI Master Burst (axi_master_burst) (v1.00.a)

DS844 June 22, 2011 **Product Specification**

Introduction

The AXI Master Burst is a continuation of the Xilinx family of AXI4-compatible LogiCORE™ IP products. It provides a bidirectional interface between a User IP core and the AXI4 interface standard. This version of the AXI Master Burst has been optimized for bus mastering operations consisting of burst transactions.

Features

- Compatible with 32, 64, and 128-bit AXI4
- Parameterizable data width of Client IP Interface (IPIC) to 32, 64, or 128 bits
- Supports AXI4 Read and Write data bursts of 16, 32, 64, 128, and 256 maximum data beats
  - Transfer width is equal to the parameterized IPIC data width
  - Automatic AXI4 4K byte address boundary crossing protection
- The User interface consists of a Legacy Command/Status interface and Read and Write LocalLink interfaces for the data transactions.

  LocalLink transactions can be up to 1,048,575 bytes in length (parameterizable) with transfer read data width equal to the LocalLink data width. The AXI Master Burst automatically breaks up long transaction requests into multiple burst transactions on the AXI4 equal to the parameterized maximum burst length.

LogiCORE IP Facts Table						
Core Specifics						
Supported Device Family(1)	Virtex-6, Spartan-6					
Supported User Interfaces	AXI4 IPIC Master					
	Resources				**Frequency**	
	LUTs	**FFs**	**Slices**	**Block RAMs**	**Max. Freq.**	
	See Table 6 and Table 7					
Provided with Core						
Documentation	Product Specification					
Design Files	VHDL					
Example Design	Not Provided					
Test Bench	Not Provided					
Constraints File	Not Provided					
Simulation Model	Not Provided					
Tested Design Tools						
Design Entry Tools	EDK 13.2, XPS					
Simulation	Mentor Graphics ModelSim(2)					
Synthesis Tools	XST					
Support						
Provided by Xilinx, Inc.						

1. For a complete listing of supported devices, see IDS Embedded Edition Derivative Device Support.
2. For the supported version of the tool, see the ISE Design Suite 13: Release Notes Guide.

Applications

The AXI Master Burst provides a AXI4 mastering capability that has the legacy IPIC User interface suitable for updating to AXI4 those legacy plbv46 designs that used the plbv46_master_burst module.

Functional Description

The AXI Master Burst is designed to provide a User with a quick way to implement a light-weight mastering interface between User logic and AXI4. Figure 1 shows a block diagram of the AXI Master Burst. The port references and groupings are detailed in Table 1. The design is parameterizable to transaction data in 32, 64, and 128-bit widths for AXI4 read and write transactions Transaction request protocol between the AXI4 and the User Logic is provided by the IPIC command and Status Adapter Block. The primary data transport function is provided by the Read and Write Controller.

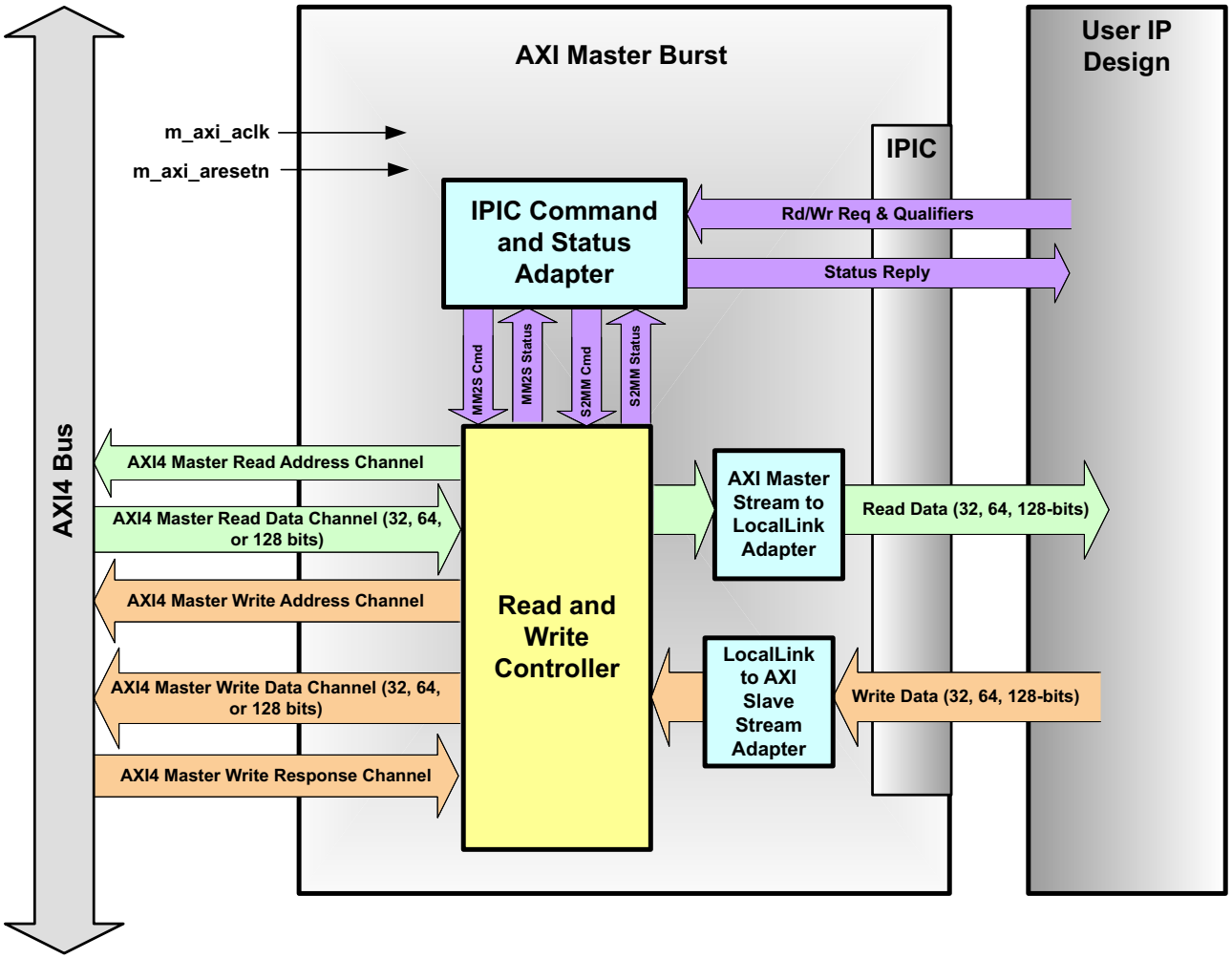

Figure 1: **AXI Master Burst Block Diagram**

Typical System Interconnect

The AXI Master Burst helper core is designed to be instantiated in a User IP design as a helper core. A typical use case is shown in Figure 2. The AXI Master Burst allows the User IP to access AXI4 slaves via the AXI4 Interconnect.

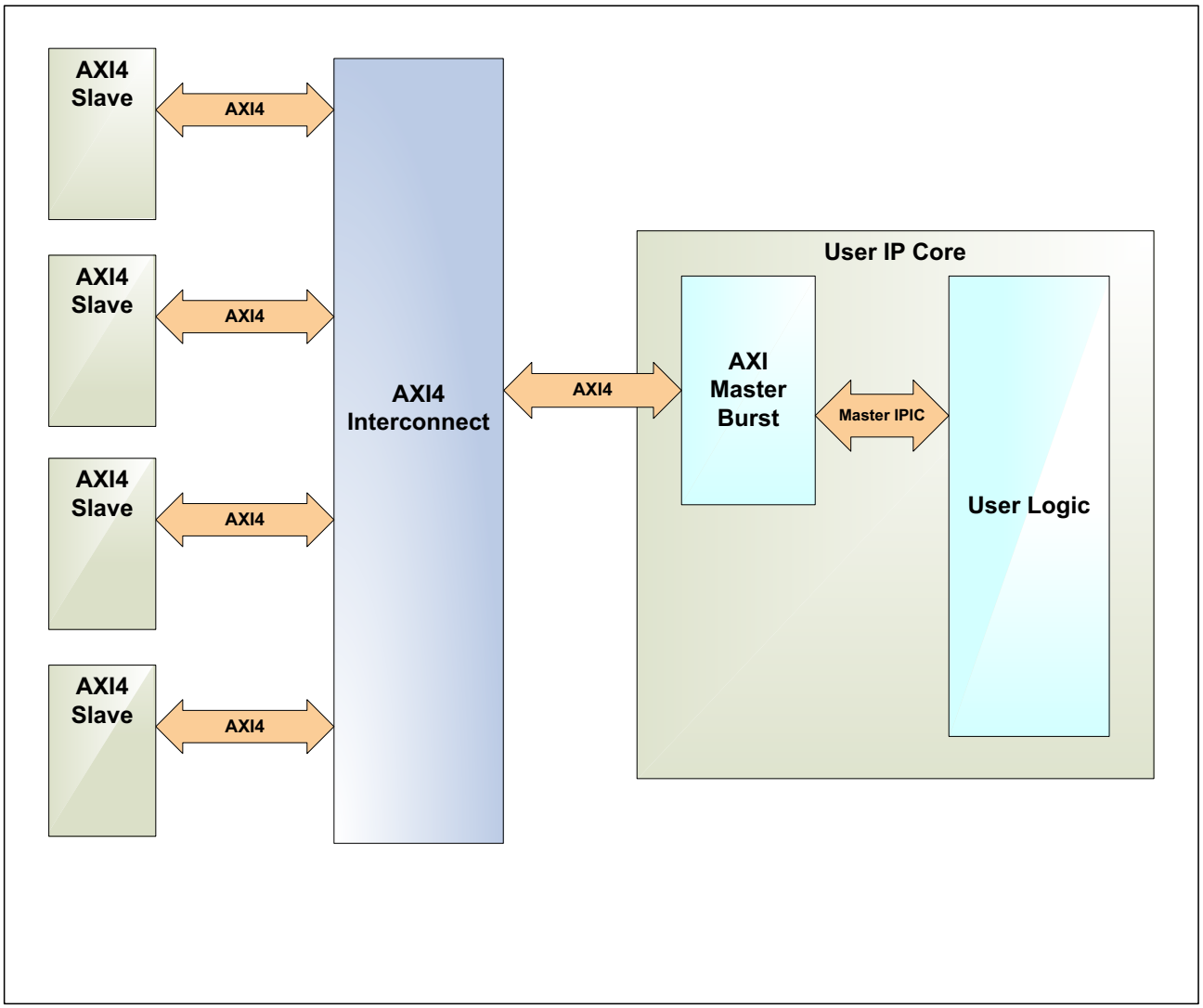

Figure 2: **Typical System Configuration Using AXI Master Burst**

I/O Signals

The AXI Master Burst signals are described in Table 1.

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

Signal Name	Interface	Signal Type	Init Status	Description
System Signals				
m_axi_aclk	Clock	I		**AXI Master Burst Clock.**
m_axi_aresetn	Reset	I		**AXI Master Burst Reset.** When asserted low, the AXI Master Burst core is put into hard reset. This signal must be synchronous to m_axi_aclk.
Master Detected Error Discrete				
md_error	Discrete Out	O		**AXI Master Burst Master Detected Error.** Active high master detected error output discrete. This bit is sticky when set and is only cleared by a hardware reset.
AXI4 Master Read Address Channel				
m_axi_arready	m_axi	I		**AXI Master Burst Read Address Channel Read Address Ready.** Indicates target is ready to accept the read address. • 1 = Target read to accept address. • 0 = Target not ready to accept address.
m_axi_arvalid	m_axi	O	0	**AXI Master Burst Read Address Channel Read Address Valid.** Indicates if m_axi_araddr is valid. • 1 = Read Address is valid. • 0 = Read Address is not valid.
m_axi_araddr(C_M_AXI_ADDR_WIDTH-1: 0)	m_axi	O	zeros	**AXI Master Burst Read Address Channel Address Bus.** The starting address for the requested read transaction.
m_axi_arlen(7:0)	m_axi	O	zeros	**AXI Master Burst Read Address Channel Burst Length.** This qualifier specifies the requested AXI Read transaction length In data beats - 1.
m_axi_arsize(2:0)	m_axi	O	zeros	**AXI Master Burst Read Address Channel Burst Size.** Indicates the data transaction width of each burst data beat. • 000b = Not Supported by AXI Master burst. • 001b =Not Supported by AXI Master burst. • 010b = 4 bytes (32-bit wide burst). • 011b = 8 bytes (64-bit wide burst). • 100b = 16 bytes (128-bit wide burst). • 101b = Not Supported by AXI Master burst. • 110b = Not Supported by AXI Master burst. • 111b = Not Supported by AXI Master burst.
m_axi_arburst(1:0)	m_axi	O	zeros	**AXI Master Burst Read Address Channel Burst Type.** Indicates type of burst. • 00b = FIXED - Not supported. • 01b = INCR - Incrementing address. • 10b = WRAP - Not supported. • 11b = Reserved.
m_axi_arprot(2:0)	m_axi	O	000b	**AXI Master Burst Read Address Channel Protection.** This is always driven with a constant output of 000b.

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

Signal Name	Interface	Signal Type	Init Status	Description
m_axi_arcache(3:0)	m_axi	O	0011b	**AXI Master Burst Read Address Channel Cache.** This is always driven with a constant output of 0011b.
AXI4 Master Read Data Channel				
m_axi_rready	m_axi	O	0	**AXI Master Burst Read Data Channel Ready.** Indicates the read channel is ready to accept read data. • 1 = Is ready. • 0 = Is not ready.
m_axi_rvalid	m_axi	I		**AXI Master Burst Read Data Channel Data Valid.** Indicates m_axi_rdata is valid. • 1 = Valid read data. • 0 = Not valid read data.
m_axi_rdata(C_M_AXI_DATA _WIDTH-1: 0)	m_axi	I		**AXI Master Burst Read Data Channel Read Data**. Read data bus for the requested read transaction.
m_axi_rresp(1:0)	m_axi	I		**AXI Master Burst Read Data Channel Response.** Indicates results of the read transaction. • 00b = OKAY - Normal access has been successful. • 01b = EXOKAY - Not supported. • 10b = SLVERR - Slave returned error on transaction. • 11b = DECERR - Decode error, transaction targeted unmapped address.
m_axi_rlast	m_axi	I		**AXI Master Burst Read Data Channel Last.** Indicates the last data beat of a burst transaction. • 0 = Not last data beat. • 1 = Last data beat.
AXI4 Master Write Address Channel				
m_axi_awready	m_axi	I		**AXI Master Burst Write Address Channel Write Address Ready.** Indicates target is ready to accept the write address. • 1 = Target ready to accept address. • 0 = Target not ready to accept address.
m_axi_awvalid	m_axi	O	0	**AXI Master Burst Write Address Channel Write Address Valid.** Indicates if m_axi_awaddr is valid. • 1 = Write Address is valid. • 0 = Write Address is not valid.
m_axi_awaddr (C_M_AXI_ADDR_ WIDTH-1: 0)	m_axi	O	zeros	**AXI Master Burst Write Address Channel Address Bus.** The starting address for the requested write transaction.
m_axi_awlen(7:0)	m_axi	O	zeros	**AXI Master Burst Write Address Channel Burst Length.** This qualifier specifies the requested AXI Write transaction length In data beats - 1.

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

Signal Name	Interface	Signal Type	Init Status	Description
m_axi_awsize(2:0)	m_axi	O	zeros	**AXI Master Burst Write Address Channel Burst Size.** Indicates the data transaction width of each burst data beat. • 000b = Not Supported by AXI Master burst. • 001b =Not Supported by AXI Master burst. • 010b = 4 bytes (32-bit wide burst). • 011b = 8 bytes (64-bit wide burst). • 100b = 16 bytes (128-bit wide burst). • 101b = Not Supported by AXI Master burst. • 110b = Not Supported by AXI Master burst. • 111b = Not Supported by AXI Master burst.
m_axi_awburst(1:0)	m_axi	O	zeros	**AXI Master Burst Write Address Channel Burst Type.** Indicates type of burst. • 00b = FIXED - Not supported. • 01b = INCR - Incrementing address. • 10b = WRAP - Not supported. • 11b = Reserved.
m_axi_awprot(2:0)	m_axi	O	000b	**AXI Master Burst Write Address Channel Protection.** This is always driven with a constant output of 000b.
m_axi_awcache(3:0)	m_axi	O	0011b	**AXI Master Burst Write Address Channel Cache.** This is always driven with a constant output of 0011b
AXI4 Master Write Data Channel				
m_axi_wready	m_axi	I		**AXI Master Burst Write Data Channel Ready.** Indicates the SG Write Data Channel target slave is ready to accept write data. • 1 = Target slave is ready. • 0 = Target slave is not ready.
m_axi_wvalid	m_axi	O	0	**AXI Master Burst Write Data Channel Data Valid.** Indicates the Write Data Channel has a valid data beat on the bus. • 1 = Valid write data. • 0 = Not valid write data.
m_axi_wdata (C_M_AXI_DATA_ WIDTH-1: 0)	m_axi	O	zeros	**AXI Master Burst Write Data Channel Write Data Bus.**
m_axi_wstrb (C_M_AXI_DATA_ WIDTH/8 - 1: 0)	m_axi	O	zeros	**AXI Master Burst Write Data Channel Write Strobe Bus.**
m_axi_wlast	m_axi	O		**AXI Master Burst Write Data Channel Last.** Indicates the last data beat of a burst transaction. • • 1 = Last data beat. • • 0 = Not last data beat.

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

Signal Name	Interface	Signal Type	Init Status	Description
AXI4 Master Write Response Channel				
m_axi_bresp(1:0)	m_axi	I		**AXI Master Burst Write Response Channel Response.** Indicates results of the write transaction. • 00b = OKAY - Normal access has been successful. • 01b = EXOKAY - Not supported. • 10b = SLVERR - Slave returned error on transaction. • 11b = DECERR - Decode error, transaction targeted unmapped address.
m_axi_bvalid	m_axi	I		**AXI Master Burst Write Response Channel Response Valid.** Indicates response, m_axi_bresp, is valid. • 1 = Response is valid. • 0 = Response is not valid.
m_axi_bready	m_axi	O	0	**AXI Master Burst Write Response Channel Ready**. Indicates source is ready to receive response. • 1 = Ready to receive response. • 0 = Not ready to receive response.
IPIC Command Interface Signals				
ip2bus_mstrd_req	IPIC	I		**AXI Master Burst Read Request.** Active high read request initiation control signal
ip2bus_mstwr_req	IPIC	I		**AXI Master Burst Write Request.** Active high write request initiation control signal
ip2bus_mst_addr(C_M_AXI_ADDR_WIDTH -1:0)	IPIC	I		**AXI Master Burst Address.** Address to be used for the specified read or write command
ip2bus_mst_be((C_NATIVE_DATA_WIDTH/8) -1: 0)	IPIC	I		**AXI Master Burst Byte Enables.** Input command qualifiers (active high byte enables) used to indicate the valid bytes for the specified read or write transaction. This input is only used for requests when the IP2Bus_Mst_Type qualifier is set to '0' for a single data beat. ***Note:*** If ip2bus_mst_be is set to all zeros for a single beat request, an internal error is induced in the AXI Master Burst.
ip2bus_mst_length(C_LENGTH_WIDTH-1:0)	IPIC	I		**AXI Master Burst Transaction Length.** Input command qualifier specifying the length of the requested transaction in bytes. The max value that can be specified is 2^n-1where n is the value assigned to the C_LENGTH_WIDTH parameter. This is only used for Burst type transactions and a value of zero is not allowed. ***Note:*** If the ip2bus_mst_length is set to all zeros for a burst request, an internal error is induced in the AXI Master Burst.
ip2bus_mst_type	IPIC	I		**AXI Master Burst Transaction Type.** Input command qualifier specifying the type of transaction being requested by the User logic. • 0 = Single Data Beat • 1 = Fixed Length Burst
ip2bus_mst_lock	IPIC	I		**AXI Master Burst Lock.** This input command qualifier is ignored by the AXI Master Burst.

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

Signal Name	Interface	Signal Type	Init Status	Description
ip2bus_mst_reset	IPIC	I		**AXI Master Burst Reset.** Active high reset input used to reset all of the AXI Master Burst logic. ***Note:*** This input should not be asserted after a transaction command has been posted to the Command Interface and before the transaction completion (as indicated by the assertion of the Bus2IP_Mst_Cmplt output status signal) has occurred.
bus2ip_mst_cmdack	IPIC	O	'0'	**AXI Master Burst Command Acknowledge.** Active high signal indicating that the Command Request (Read or Write) has been posted to and accepted by the Read/Write Controller.
bus2ip_mst_cmplt	IPIC	O	'0'	**AXI Master Burst Command Complete.** Active high signal indicating the requested transaction has completed by the Read/Write Controller and the associated status bits are valid to sample.
bus2ip_mst_error	IPIC	O	'0'	**AXI Master Burst Error.** Active high signal indicating an error was encountered by the Read/Write Controller during the requested transaction. This signal is valid when Bus2IP_Mst_Cmplt is asserted.
bus2ip_mst_rearbitrate	IPIC	O	'0'	**AXI Master Burst Rearbitrate.** Not part of AXI4. This signal is always set to '0'.
bus2ip_mst_timeout	IPIC	O	'0'	**AXI Master Burst Timeout.** Not part of AXI4. This signal is always set to '0'.
IPIC Read LocalLink Data Interface Signals (Note: Legacy IPIC is Big Endian)				
bus2ip_mstrd_d(C_NATIVE_DATA_WIDTH - 1:0)	IPIC	O	zeros	**AXI Master Burst Read LocalLink Data**. The Read LocalLink data output bus.
bus2ip_mstrd_rem((C_NATIVE_DATA_WIDTH/8])- 1:0)	IPIC	O	ones	**AXI Master Burst Read LocalLink Remainder**. REM output presented in active low mask format.
bus2ip_mstrd_sof_n	IPIC	O	'1'	**AXI Master Burst Read LocalLink Start of Frame**. Active low signal indicating the starting data beat of a Read LocalLink packet transaction.
bus2ip_mstrd_eof_n	IPIC	O	'1'	**AXI Master Burst Read LocalLink End of Frame**. Active low signal indicating the ending data beat of a Read LocalLink packet transaction.
bus2ip_mstrd_src_rdy_n	IPIC	O	'1'	**AXI Master Burst Read LocalLink Source Ready**. Active low signal indicating that the data value asserted on the Bus2IP_MstRd_d output bus is valid and ready for transaction.
bus2ip_mstrd_src_dsc_n	IPIC	O	'1'	**AXI Master Burst Read LocalLink Source Discontinue**. Active low signal indicating that the Read LocalLink Source (Master) needs to discontinue the transaction. This is only asserted by the AXI Master Burst if an internal error is encountered by the MM2S side of the Read/Write Controller. The bus2ip_mstrd_src_dsc_n is asserted (in conjunction with bus2ip_mstrd_eof_n) until the LocalLink data beat is completed with acceptance by the destination via assertion of ip2bus_mstrd_dst_rdy_n.

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

Signal Name	Interface	Signal Type	Init Status	Description
ip2bus_mstrd_dst_rdy_n	IPIC	I		**AXI Master Burst Read LocalLink Destination Ready**. Active low input signal indicating that the LocalLink destination (User logic) is ready for a data transaction beat. ***Note:*** The AXI Master Burst does not issue any read requests on the AXI Read Address Channel until the ip2bus_mstrd_dst_rdy_n is asserted low for at least 1 clock period after the IPIC read command has been issued.
ip2bus_mstrd_dst_dsc_n	IPIC	I		**AXI Master Burst Read LocalLink Destination Discontinue**. Active low signal indicating that the Read LocalLink Destination (User logic) needs to discontinue the transaction. This is not supported by the AXI Master Burst. User logic should tie this signal to a constant logic high.
IPIC Write Data Interface Signals (Note: Legacy IPIC is Big Endian)				
ip2bus_mstwr_d(C_NATIVE_DATA_WIDTH - 1:0)	IPIC	I		**AXI Master Burst Write LocalLink Data.** Write Data input
ip2bus_mstwr_rem([C_NATIVE_DATA_WIDTH/8] - 1:0)	IPIC	I		**AXI Master Burst Write LocalLink Remainder**. REM input presented in active low mask format.
ip2bus_mstwr_src_rdy_n	IPIC	I		**AXI Master Burst Write LocalLink Source Ready**. Active low input signal indicating that the data and qualifiers asserted on the Write LocalLink bus are valid (from the source) and ready for a transaction data beat. ***Note:*** The AXI Master Burst does not issue any write requests on the AXI Write Address Channel until the ip2bus_mstwr_src_rdy_n is asserted low for at least 1 clock period after the IPIC write command has been issued.
ip2bus_mstwr_src_dsc_n	IPIC	I		**AXI Master Burst Write LocalLink Source Discontinue**. Active low input signal indicating that the Write LocalLink Source (User Logic) needs to discontinue the transaction. This is not supported by the AXI Master Burst and should be tied to logic '1' by the User logic.
ip2bus_mstwr_sof_n	IPIC	I		**AXI Master Burst Write LocalLink Start of Frame**. Active low input signal indicating the starting data beat of a Write LocalLink packet transaction.
ip2bus_mstwr_eof_n	IPIC	I		**AXI Master Burst Write LocalLink End of Frame**. Active low input signal indicating the ending data beat of a Write LocalLink packet transaction.

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

Signal Name	Interface	Signal Type	Init Status	Description
bus2ip_mstwr_dst_rdy_n	IPIC	O	'1'	**AXI Master Burst Write LocalLink Destination Ready.** Active low output signal indicating that the AXI Master Burst is ready to accept a data transaction on the Write LocalLink.
bus2ip_mstwr_dst_dsc_n	IPIC	O	'1'	**AXI Master Burst Write LocalLink Destination Discontinue.** Active low output signal indicating that the AXI Master Burst needs to discontinue Write LocalLink operations. This is only asserted by the AXI Master Burst if an internal error is encountered by the S2MM side of the Read/Write Controller. The bus2ip_mstwr_dst_dsc_n is asserted until the LocalLink data beat is completed. The bus2ip_mstwr_dst_dsc_n is asserted until the LocalLink data beat is completed. The Master Burst still expects the Write LocalLink to be terminated with a ip2bus_mstwr_eof_n assertion in the next LocalLink data beat.

Design Parameters

The AXI Master Burst Design Parameters are listed and described in Table 2.

Table 2: **AXI Master Burst Design Parameter Description**

Parameter Name	Allowable Values	Default Values	VHDL Type	Feature/Description
AXI Master Burst General Parameters				
C_FAMILY	virtex6, spartan6	virtex6	String	Specifies the target FPGA family
AXI Master Burst AXI4 Parameters				
C_M_AXI_ADDR_WIDTH	32	32	integer	Address width (in bits) of AXI4 Interface. This is currently fixed at 32 bits.
C_M_AXI_DATA_WIDTH	32, 64, 128, 256	32	integer	Data width (in bits) of AXI4 Interface.
C_MAX_BURST_LEN	16, 32, 64, 128, 256	16	integer	Specifies the maximum number of data beats to use for each AXI transaction initiated by the AXI Master Burst
C_ADDR_PIPE_DEPTH	1-14	1	integer	Specifies the depth of the address pipelining the AXI Master Burst will support when submitting transaction requests to the AXI Address Channels.
C_NATIVE_DATA_WIDTH	32, 64, 128	32	integer	Data width (in bits) of LocalLink Data Interface. The value assigned must be less than or equal to the value assigned to the C_M_AXI_DATA_WIDTH parameter
C_LENGTH_WIDTH	12-20	12	integer	Specifies the width (in bits) of the IPIC ip2bus_mst_length input port. The value limits the maximum number of bytes to be transacted that can be specified by the user on the ip2bus_mst_length input port.

Parameter Descriptions

C_FAMILY

- **Type:** string
- **Allowed Values:** Spartan®-6 and Virtex®-6 FPGAs
- **Definition:** Indicates the target FPGA device family for the design
- **Description:** This parameter is set by the EDK tools to a value reflecting the FPGA device family selected for the EDK project.

C_M_AXI_ADDR_WIDTH

- **Type:** Integer
- **Allowed Values:** 32 (default = 32)
- **Definition:** Bit width of the AXI Read and AXI Write Address Channels on the AXI Master Burst AXI4 interface.
- **Description:** This integer parameter is used to size the Read Address and Write Address Channels of the AXI4 AXI Master Burst interface. The EDK tool suite assigns this parameter a fixed value of 32.

C_M_AXI_DATA_WIDTH

- **Type:** Integer
- **Allowed Values:** 32, 64, 128, or 256 (default = 32)
- **Definition:** Bit width of the AXI Read and AXI Write Data Channels on the AXI Master Burst AXI4 interface
- **Description:** This integer parameter is used to size the Read Data and Write Data Channels of the AXI4 AXI Master Burst interface.

C_MAX_BURST_LEN

- **Type:** Integer
- **Allowed Values:** 16, 32, 64, 128, or 256 (default = 16)
- **Definition:** This parameter limits the burst length requested by AXI Master Burst on the AXI4 data transport interface.

C_NATIVE_DATA_WIDTH

- **Type:** Integer
- **Allowed Values:** 32, 64, or 128 (default = 32)
- **Definition:** Defines the bit width of the LocalLink Read and Write data channels
- **Description:** This integer parameter is used to size the Read LocalLink Data and Write LocalLink Data Channels of the AXI4 AXI Master Burst IPIC interface. The value assigned must be less than or equal to the value assigned to the C_M_AXI_DATA_WIDTH parameter.

C_ADDR_PIPE_DEPTH

- **Type:** Integer
- **Allowed Values:** 1-14 (default = 1)
- **Definition:** Sets the address pipeline limit used by AXI Master burst for posting requests on the AXI Address Channels
- **Description:** The effective address pipelining on the AXI4 Read and Write Address Channels will be the value assigned plus 2. If the value assigned is 1, the effective address pipelining will be 2.

C_LENGTH_WIDTH

- **Type:** Integer
- **Allowed Values:** 12 to 20 (default = 12)
- **Definition:** Sets the bit width of the IPIC ip2bus_mst_length command qualifier.
- **Description:** The bit width allows a maximum of 2^n-1 bytes to be specified for transaction per command submitted by the User on the IPIC Command interface.
  - 12 bits = 4,095 bytes max per command
  - 13 bits = 8,191 bytes max per command
  - 14 bits = 16,383 bytes max per command
  - 15 bits = 32,767 bytes max per command
  - 16 bits = 65,535 bytes max per command
  - 17 bits = 131,071 bytes max per command
  - 18 bits = 262,143 bytes max per command
  - 19 bits = 524,287 bytes max per command
  - 20 bits = 1,048,575 bytes max per command

Parameter - I/O Signal Dependencies

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

Parameter Name	Affects Port	Depends on Parameter	Relationship Description
C_M_AXI_DATA_WIDTH	m_axi_rdata m_axi_wdata m_axi_wstrb		The value assigned to the parameter sets the vector width of the affected port.
C_M_AXI_ADDR_WIDTH	ip2bus_mst_addr m_axi_awaddr m_axi_araddr		The value assigned to the parameter sets the vector width of the affected port.
C_NATIVE_DATA_WIDTH	ip2bus_mst_be ip2bus_mstwr_d bus2ip_mstrd_d ip2bus_mstwr_rem bus2ip_mstrd_rem		The value assigned to the parameter sets the vector width of the affected port.
C_NATIVE_DATA_WIDTH		C_M_AXI_DATA_WIDTH	The value assigned to C_NATIVE_DATA_WIDTH must be less than or equal to the value assigned to C_M_AXI_DATA_WIDTH
C_LENGTH_WIDTH	ip2bus_mst_length		The value assigned to the parameter sets the vector width of the affected port.

Clocking

AXI Master Burst utilizes a single clock for logic synchronization. This clock is input on the `m_axi_aclk` input port. All interfaces for the core are required to be synchronized to this clock. The AXI Master Burst has been simulation tested with an `m_axi_aclk` frequency range of 10 MHz to 200 MHz. Actual Fmax achieved in a hardware implementation can vary. See the section Performance.

Resets

An active low reset assertion on the AXI Master Burst `m_axi_aresetn` input resets the entire AXI Master Burst core. This is considered a hardware reset and there are no graceful completions of AXI4 transactions in progress. A hardware reset initializes all AXI Master Burst internal logic to power on conditions. It is required that the `m_axi_aresetn` input is synchronous to the `m_axi_aclk` master clock input and is asserted for the minimum number of clocks stated in Table 4. The table also indicates the stabilization time for AXI Master Burst outputs reacting to a reset condition.

Table 4: **Reset Assertion/Deassertion Stabilization Times**

Description	Value	Applicable Signal
Minimum assertion time	8 clocks (m_axi_aclk)	axi_resetn input
Reset assertion to output signals in reset state (maximum)	3 clocks (m_axi_aclk)	All output signals
Reset deassertion to normal operation state (maximum)	3 clocks (m_axi_aclk)	All output signals

The input signal `Bus2IP_Mst_Reset` should not be asserted after a transaction command has been posted to the AXI Master Burst command interface and before it has completed with the assertion of the `Bus2IP_Mst_Cmplt`. To do so can cause the Master to violate the AXI4 protocol and hang the AXI4 system connected to the Master.

Performance

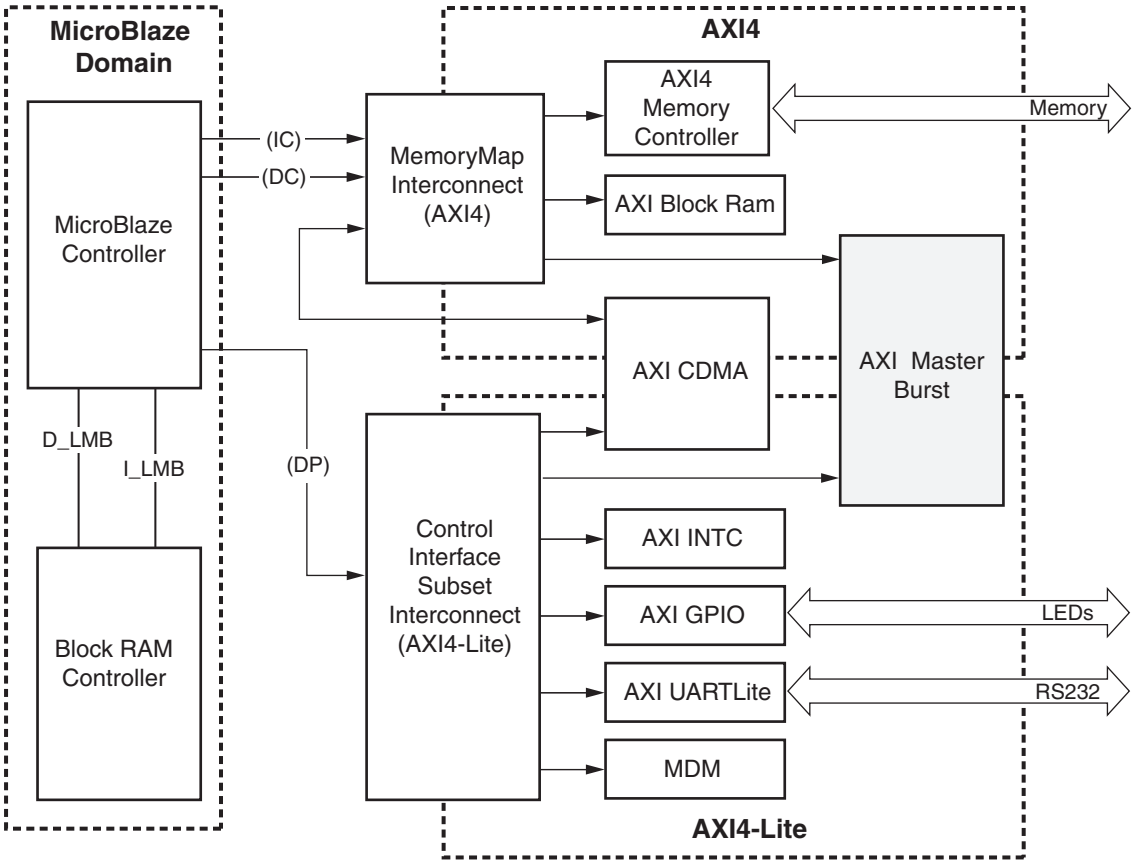

Figure 3: **Virtex-6 and Spartan-6 FPGA System Configuration Diagram**

The target FPGA was filled with logic to drive the LUT and block RAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target F_{MAX} numbers are shown in Table 5.

Table 5: **System Performance**

Target FPGA	Target F_{MAX} (MHz)		
	AXI4	AXI4-Lite	MicroBlaze
xc6slx45t (1)	90	120	80
xc6vlx240t (2)	135	180	135

Notes:

1. Spartan-6 FPGA LUT utilization: 70%; Block RAM utilization: 70%; I/O utilization: 80%; MicroBlaze™ processor not AXI4 interconnect; AXI4 interconnect configured with a single clock of 120 MHz.
2. Virtex-6 FPGA LUT utilization: 70%; Block RAM utilization: 70%; I/O utilization: 80%.

Throughput

The AXI Master Burst performs AXI4 Burst transactions of 1 to 16, 1 to 32, 1 to 64, 1 to 128, or 1 to 256 data beats per AXI4 request depending on the setting of the C_MAX_BURST_LEN parameter. The AXI Master Burst breaks up the parent IPIC command (of up to 1,048,575 bytes, see C_LENGTH_WIDTH parameter) into these smaller AXI4 transactions (child commands). In addition, the AXI4 4k address boundary crossing protection is automatically performed by the AXI Master Burst by ensuring generated transaction requests do not cause a 4k byte address boundary crossing. The AXI Master Burst does not add any transaction bubbles between spawned child command data transactions as long as the User logic does not throttle the LocalLink interface and the AXI Interconnect/Target AXI Slave can keep up on the AXI address, AXI data, and in the case of writes, the AXI response channels.

AXI Address Channel Request Posting Hold-off

The AXI Master Burst is designed such that it does not begin committing transaction requests in the pertinent AXI Address Channel until the User Logic indicates it is ready via LocalLink ready signaling for at least one clock period after the IPIC input command has been accepted. This is done to optimize the use of the AXI Interconnect and Target AXI Slave by the system. Otherwise, it is possible to stall the AXI Interconnect and Target Slave with committed transaction requests from the Master if the associated data transactions do not occur because the User Logic is not ready to provide write data or accept read data (depending on the type of command).

Transaction Timing Examples

The next section shows timing relationships for AXI4 and the IPIC interface signals during read and write transactions. Actual timing relationships can vary depending on AXI handshaking and LocalLink handshaking.

Single Data Beat Read Operation

A single beat read cycle is shown in Figure 4. The diagram shows the AXI Slave accepting the read address and qualifiers in one clock cycle and presenting the read data in the next clock cycle.

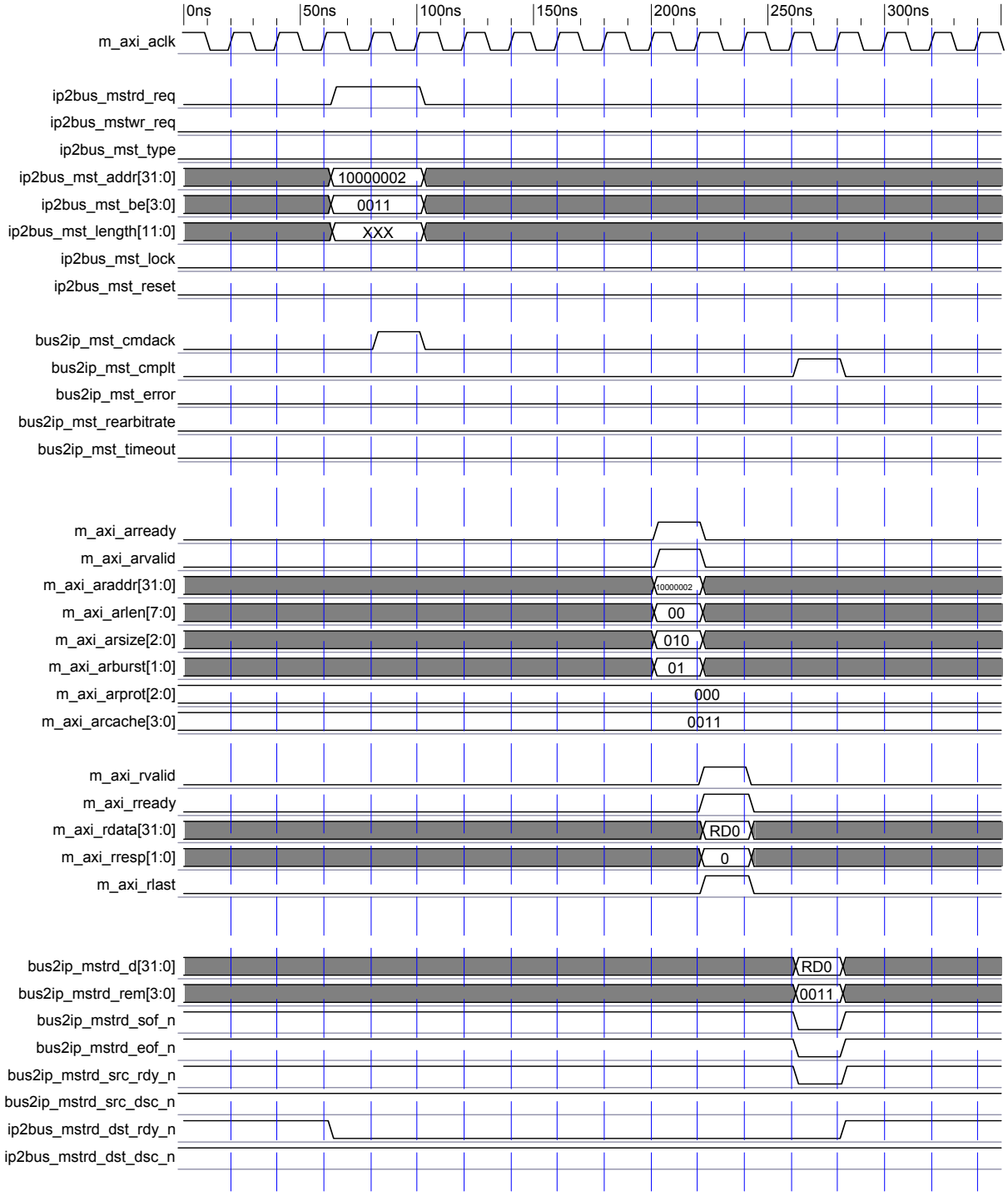

Figure 4: **Example Single Beat Read Transaction Timing**

Single Data Beat Write Operation

A typical single beat write cycle is shown in Figure 5.

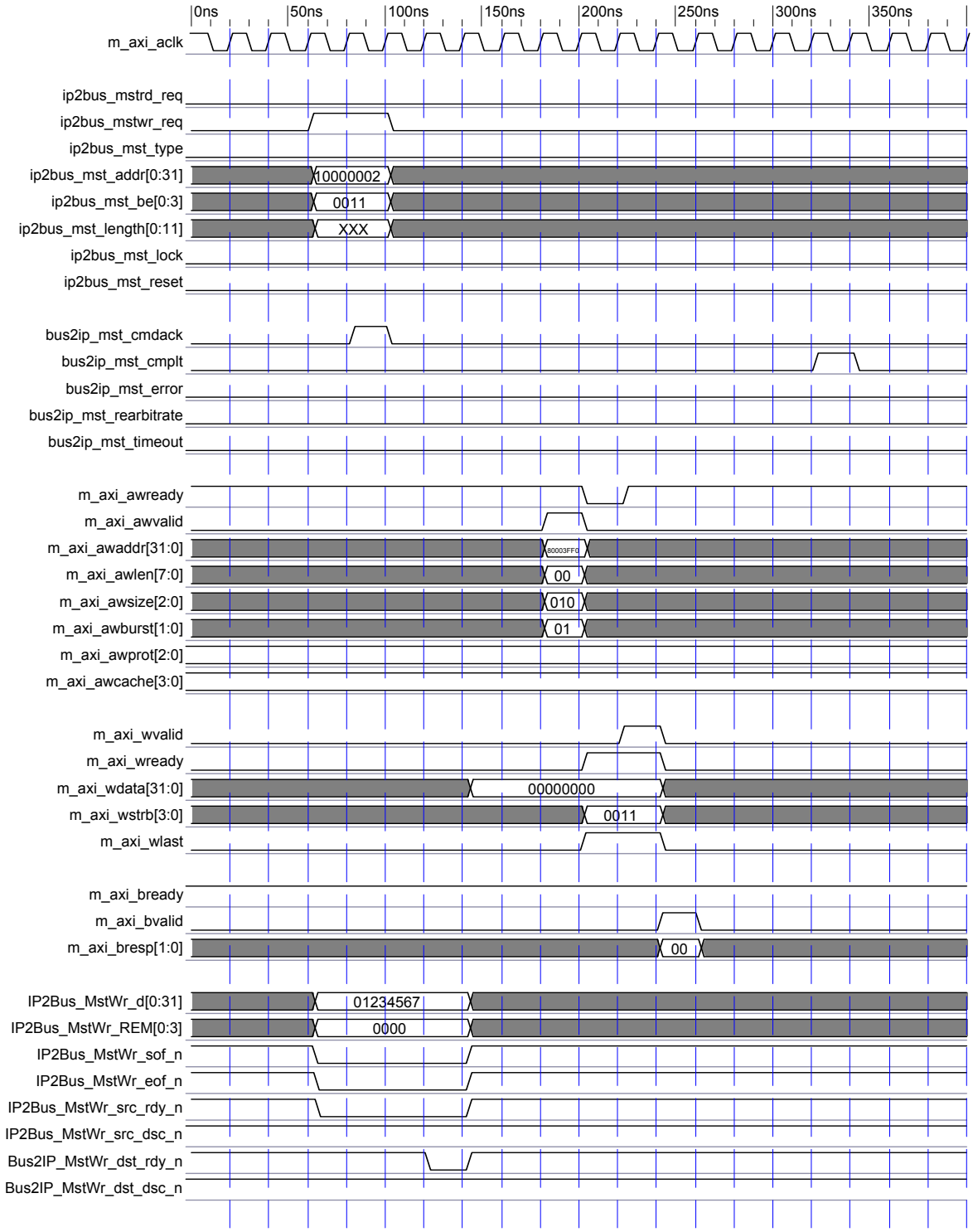

Figure 5: **Example Single Beat Write Transaction Timing**

Single Data Beat Read Operation with AXI Read Data Channel Reported Error

A single data beat Read transaction with a Slave reported error is shown in Figure 6. The AXI Read Data Channel response error is captured, reported on the IPIC Status Channel, and the Master's `md_error` output is asserted and held. The assertion of `md_error` is cleared by the `ip2bus_mst_reset` input from the IPIC Command interface. The `m_axi_aresetn` would clear the `md_error` if it were asserted.

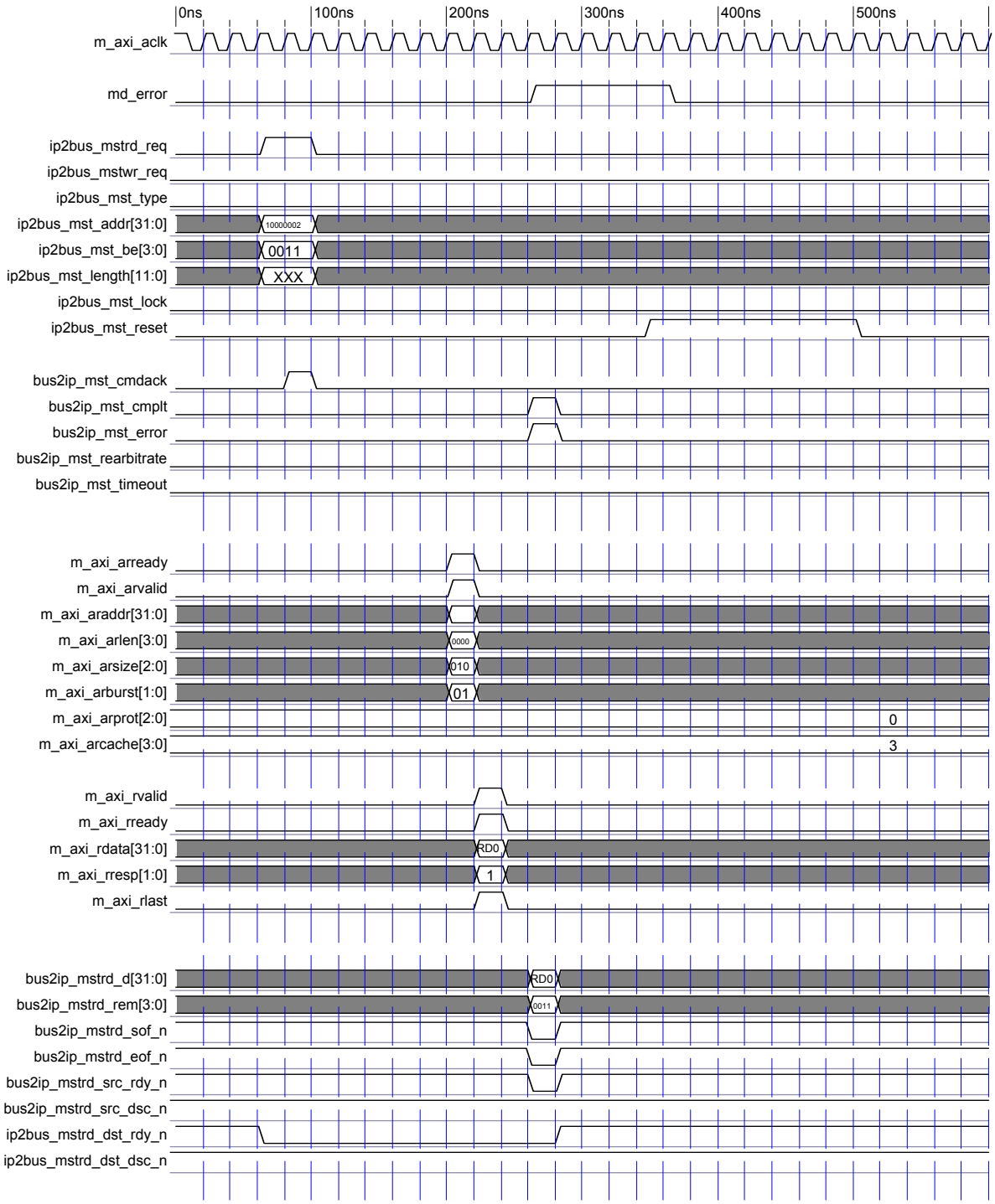

Figure 6: **Example Single Beat Read Transaction Timing With Error**

Single Data Beat Write Operation with AXI Response Channel Reported Error

A single beat write transaction is shown in Figure 7. The AXI Write Response Channel response error is captured, reported on the IPIC Status Channel, and the Master's `md_error` output is asserted and held. The assertion of `md_error` is cleared by the `ip2bus_mst_reset` input from the IPIC Command interface. The `m_axi_aresetn` would clear the `md_error` if it were asserted.

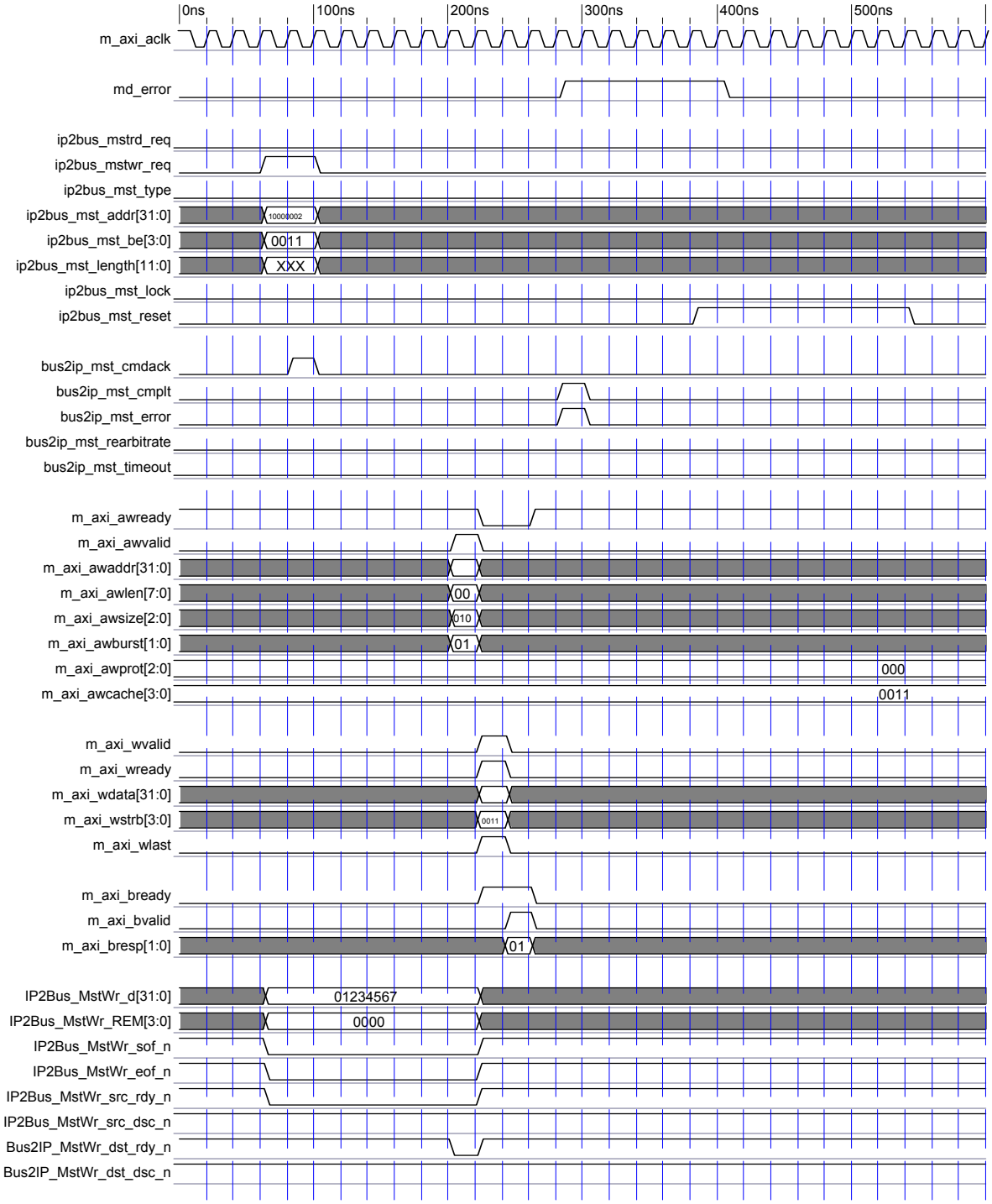

Figure 7: **Example Single Beat Write Transaction Timing With Error**

Burst Read Transaction

A burst Write transaction of 80 bytes is shown in Figure 10. This example is for a AXI Master Burst configured for a 32-bit native data width of 32-bits and a maximum allowed burst length of 16 data beats per AXI4 transaction. The command length of 80 bytes requires the Master to break the transaction up into two AXI4 transactions, one of 16 data beats and one of four data beats.

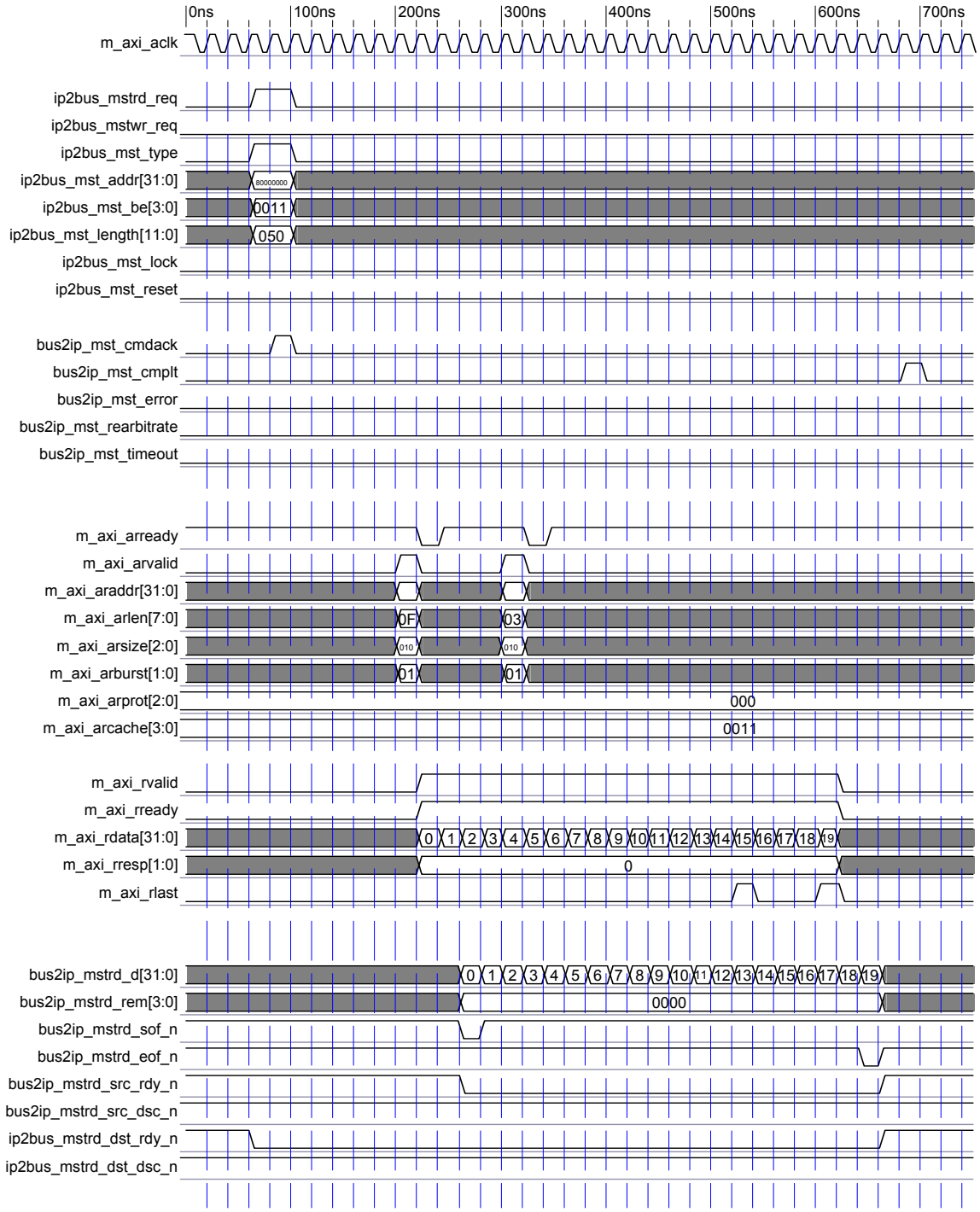

Figure 8: **Example Burst Read Transaction Timing**

Burst Read Discontinue

The AXI Master burst issues a discontinue on the Read LocalLink if an internal error is encountered during the read transaction. This is normally caused by the User logic setting the `ip2bus_mst_length` qualifier to a value of zero on the IPIC Command interface during a read command assertion. LocalLink requires all transactions to complete with an EOF assertion, even during a discontinue. Figure 9. shows an example of the AXI Master Burst issuing a discontinue on a Read burst transaction as the result of an internal error. The Read LocalLink is terminated early with the EOF assertion by the source.

If the LocalLink is not terminated correctly by the destination, the AXI Master Burst does not assert the `bus2ip_mst_cmplt` *status signal.*

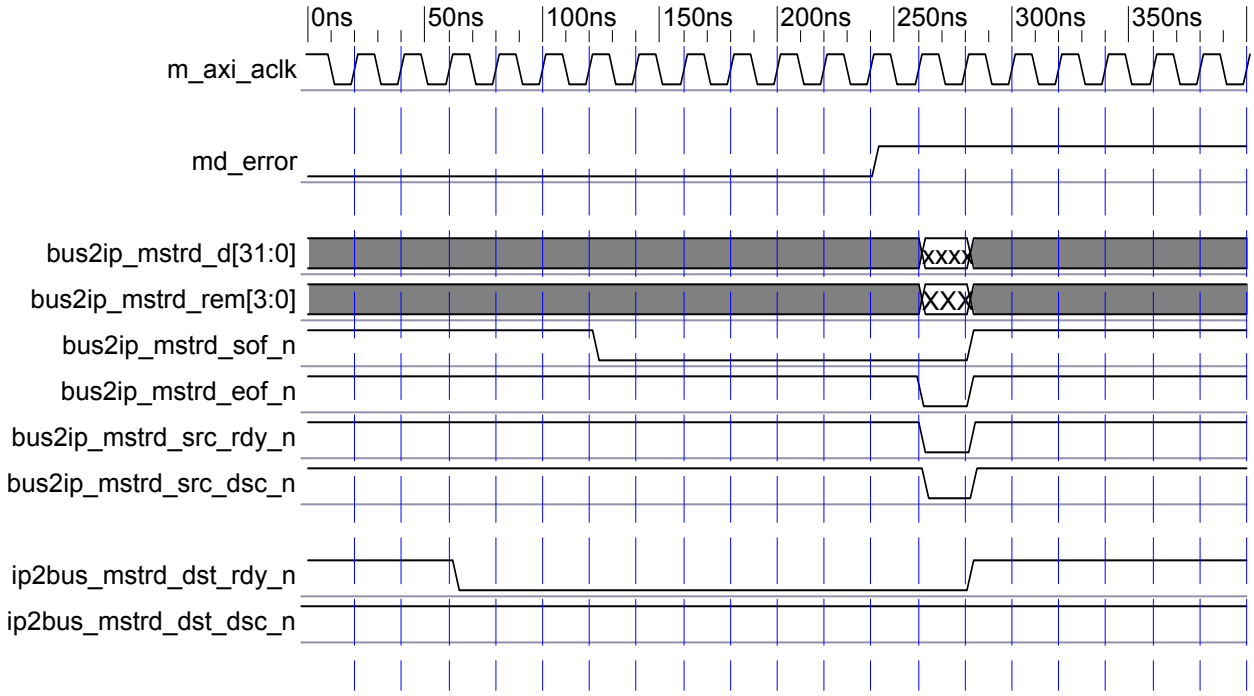

Figure 9: **Example Burst Read Discontinue Timing**

Burst Write Transaction

A burst Write transaction of 80 bytes is shown in Figure 10. This example is for a AXI Master Burst configured for a 32-bit native data width and a maximum allowed burst length of 16 data beats per AXI4 transaction. The command length of 80 bytes requires the Master to break the transaction up into two AXI4 transactions, one 16 data beats and one four data beats.

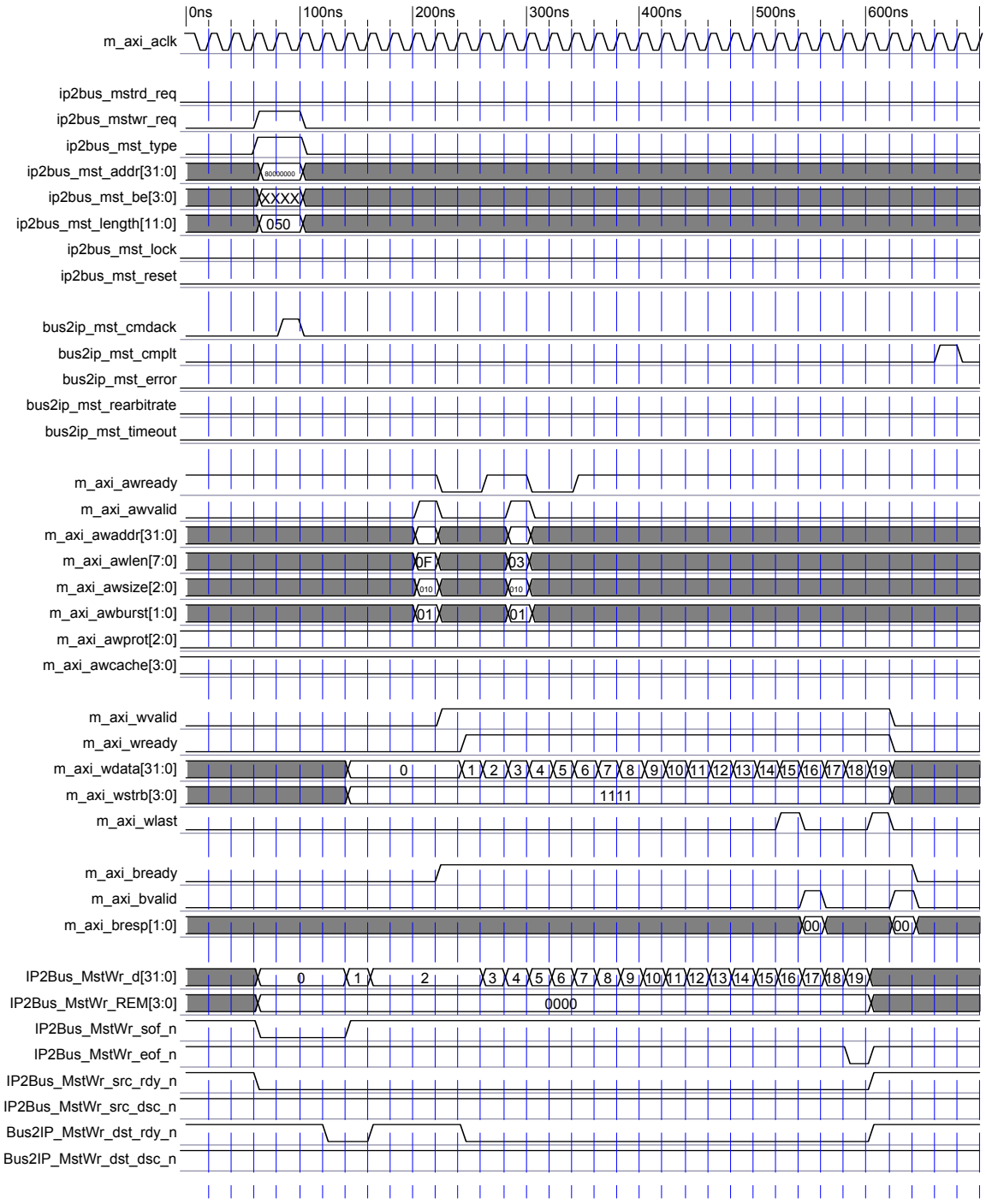

Figure 10: **Example Burst Write Transaction Timing**

Burst Write Discontinue

The AXI Master burst issues a discontinue on the Write LocalLink if an internal error is encountered during the write transaction. This is normally caused by the User logic setting the `ip2bus_mst_length` qualifier to a value of zero on the IPIC Command interface during a write command assertion. LocalLink requires all transactions to complete with an EOF assertion, even during a discontinue. Figure 11. shows an example of the AXI Master Burst issuing a discontinue on a Write burst transaction as the result of an internal error. The Write LocalLink is terminated early with the EOF assertion by the source after the `bus2ip_mstwr_dst_dsc_n` assertion is detected.

If the LocalLink is not terminated correctly by the source, the AXI Master Burst does not assert the `bus2ip_mst_cmplt` *status signal.*

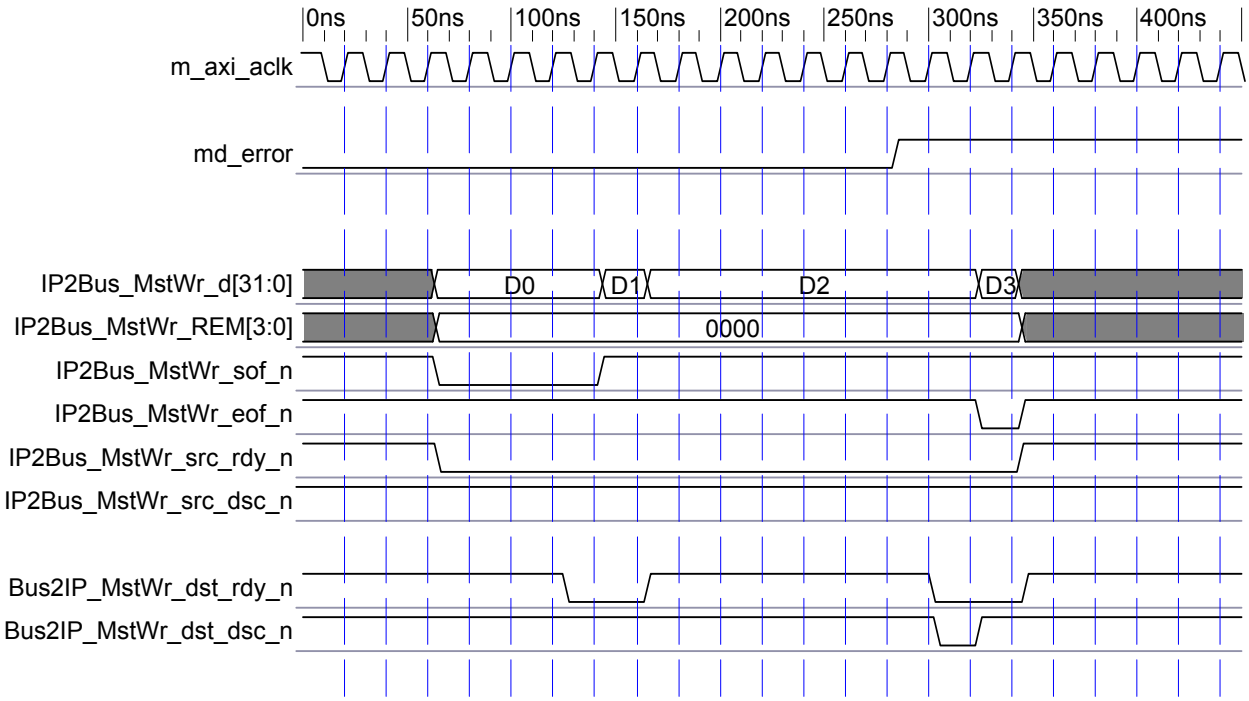

Figure 11: **Example Burst Write Discontinue Timing**

Resource Utilization

Resource utilization numbers for the AXI Master Burst core are shown for the Spartan-6 FPGA family in Table 6 and for the Virtex-6 FPGA family in Table 7. These values have been generated using the Xilinx® EDK and ISE® tools for version 13.2.

Table 6: **Spartan-6 FPGA Resource Estimates**

C_M_AXI_DATA_WIDTH	C_M_AXI_ADDR_WIDTH	C_NATIVE_DATA_WIDTH	C_MAX_BURST_LEN	C_ADDR_PIPE_DEPTH	C_LENGTH_WIDTH	Slices	Slice Reg	Slice LUTs	Block RAM
32	32	32	16	4	20	281	588	526	0
128	32	128	256	4	20	450	1299	833	1

Table 7: **Virtex-6 FPGA Resource Estimates**

C_M_AXI_DATA_WIDTH	C_M_AXI_ADDR_WIDTH	C_NATIVE_DATA_WIDTH	C_MAX_BURST_LEN	C_ADDR_PIPE_DEPTH	C_LENGTH_WIDTH	Slices	Slice Reg	Slice LUTs	Block RAM
32	32	32	16	4	20	307	603	501	0
128	32	128	256	4	20	454	1319	801	1

Support

Xilinx provides technical support for this LogiCORE IP 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 IP modules and software, contact your local Xilinx sales representative.

Reference Documents

The AXI4 AMBA® AXI Protocol Version: 2.0 Specification contains reference information important to understanding the design.

List of Acronyms

Acronym	Spelled Out
AXI	Advanced eXtensible Interface
EDK	Embedded Development Kit
EOF	End Of Frame
FF	Flip-Flop
FPGA	Field Programmable Gate Array
I/O	Input/Output
IP	Intellectual Property
IPIC	Intellectual Property Interface Connection
LUT	Lookup Table
MHz	Mega Hertz
R/W	Read/Write
RAM	Random Access Memory
HW	Hardware
SG	Scatter Gather
SW	Software
VHDL	VHSIC Hardware Description Language (VHSIC an acronym for Very High-Speed Integrated Circuits)
XPS	Xilinx Platform Studio (part of the EDK software)
XST	Xilinx Synthesis Technology

Revision History

The this table shows the revision history for this document:

Date	Version	Description of Revisions
06/22/2011	1.0	Initial Xilinx Release

Notice of Disclaimer