PLB Central DMA Controller (v1.00a)

DS493 March 12, 2007 **Product Specification**

Introduction

The PLB Central DMA Controller provides simple Direct Memory Access (DMA) services to peripherals and memory devices on the PLB. The controller transfers a programmable quantity of data from a source address to a destination address without processor intervention.

Features

- Provides a single physical channel of Direct Memory Access between a source address and a destination address
- Provides programmable registers for source address, destination address, transfer length and data size (32-bit or 64-bit data transfer)
- Supports setting up of source and destination addresses as fixed or increasing, as the DMA operation progresses (for supporting keyhole memory devices)
- Supports 32-bit aligned word size transfers and 64-bit aligned double word size transfers
- Supports PLB burst transfers

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex™-II Pro, Virtex-4	
Version of Core	plb_central_dma	v1.00a
Resources Used		
Slices	See Table 14 & Table 15	
LUTs		
FFs		
Block RAMs		
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	ISE 8.1i or later	
Verification	N/A	
Simulation	ModelSim SE/EE 6.0 or later	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

Functional Description

The block diagram for the PLB Central DMA Controller is shown in Figure 1. The core is comprised of the primary modules:

- Slave Attachment Module
- Master Attachment Module
- Memory Buffer.

The modules are described in the following sections.

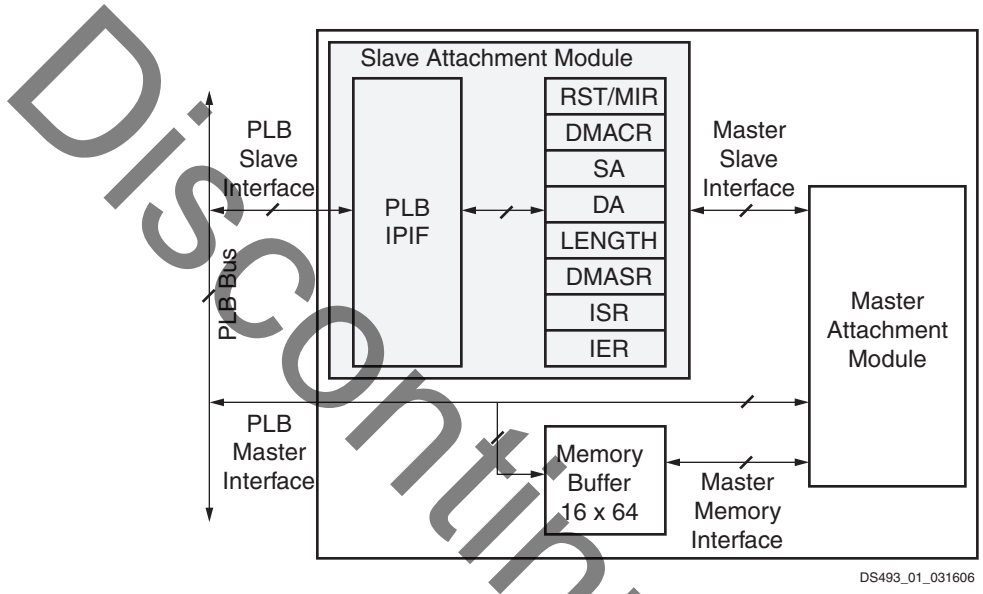

Figure 1: **PLB Central DMA Block Diagram**

Slave Attachment

- The Slave Attachment module performs the following operations:
- Interfaces with the PLB using the PLB IPIF interface
- Reads and writes the PLB Central DMA controller registers through the PLB slave interface
- Modifies the source address, destination address, length, DMA status and interrupt status registers with inputs from Master Attachment module
- Generates interrupts based on DMA done and DMA error condition, coming from Master Attachment module

Master Attachment

The Master Attachment module performs the following operations:

- Provides PLB master interface (without using the IPIF interface)
- Controls read and write transactions to transfer the data specified in the Length register from the source address to destination address
- Handles conditions like a PLB slave terminating the burst transaction and a bus time-out condition
- Updates the Source Address, Destination Address, Length and DMA Status Registers during the

DMA transfer

Memory Buffer

PLB Central DMA core contains of a 16X64 internal data buffer, which performs the following:

- Supports the PLB burst transfers to speed up the DMA operation
- 32-bit transfers based on the present source and destination address. The mirroring operation (for the 32-bit transfers) is done before the data is put into the internal data buffer, such that the mirrored data is put on the PLB interface during the write cycle

PLB Central DMA Controller Operation

The PLB Central DMA controller operates on the PLB as a master/slave device. It responds as a slave when its registers are being read and written. As a master, it initiates read and write transactions when a DMA operation has been started.

The operation of the PLB Central DMA is initiated by writing values into the following DMA registers. For more information, see the PLB Central DMA Controller Register Descriptions section.

- DMA Control Register (DMACR): The contents of this register set the main parameters for the DMA transfer as follows (Refer to Table 7):
  - Source Increment (SINC): SINC should be set to '0', if and only if the Source Address register is written with a *keyhole* address such that a single address is associated with a sequence of data[1]. If the source address should increment for each data transferred, SINC should set to '1'.
  - Destination Increment (DINC): DINC should be set to '0' if and only if the Destination Address is written with a *keyhole* address such that a single address is associated with a sequence of data[2]. If the destination address should increment for each data transferred, DINC should set to '1'.
  - DMA Transfer Data Size (DSIZE): It is set to 4 bytes or 8 bytes, to select the data size to be used for individual bus transfers.
- LENGTH register (LENGTH): The number of bytes to transfer is written into this register. Writing of this register is the event that starts the DMA operation, so it must be done last. (Refer to Table 10 for more information on this register)

The data transfer starts by reading data from the source address into a internal data buffer followed by the transfer of data from the internal buffer to the destination address. This repeats until all the data is moved. The status registers get updated as the DMA operation progresses.

1. An example of a keyhole address is a memory mapped FIFO that maps as an element at a single address but can consume or produce an endless sequence of data. A variation on a keyhole address is a *wide keyhole* where a number of consecutive addresses map to the element. When SINC = '0', PLB Central DMA will perform all read transactions to the same address. It is the responsibility of the corresponding slave to impose a keyhole behavior. For bursts, this implies that the slave will not increment the keyhole address during the burst or will make the keyhole at least as wide as the burst size. It is not feasible to use SINC = '0' with a non-keyhole slave such as a memory. Even though the DMA Controller would deliver a non-incremented address during the address phase of PLB transactions, the memory would increment it during bursts. (See also DINC, which is subject to the same considerations when set to '0'.)
2. See the above note for SINC = '0'. The case DINC = '0' has the same considerations but for the destination address.

While it is moving data as the bus master, the PLB Central DMA Controller attempts to move data efficiently. However, the PLB Central DMA Controller has the following built-in limitations:

- The PLB Central DMA Controller does not support simultaneous read and write transactions. This means each datum will cross the bus twice, once while being read from the source address into the internal data buffer of the PLB Central DMA Controller, and once while being written from the internal data buffer to the destination address. (No *fly by* mode).
- This core does not support conversion cycles, where a 64-bit master accesses a 32-bit slave with requested bytes on both the lower and upper 32-bits of the 64-bit data bus.
- This core does not issue transfer abort as this condition does not arise during DMA transfer.
- Each bus transaction moves only the amount of data given by the DSIZE register. For example, if DSIZE is four bytes, then each bus transaction moves only a word.
- When the LENGTH register has odd number of bytes, the number of bytes transferred are greater than the odd number mentioned in the LENGTH register and are a multiple of DSIZE. All these DSIZE multiple of bytes are written into the internal data buffer during the read cycle, but during the write cycle, byte-enables are issued such that only the valid bytes of data is written into the memory. For example, if DSIZE is 1000 and the LENGTH register is programmed to be 264 bytes (33 double words), the PLB Central DMA controller will finish the transfer with a single data beat after performing 2 fixed length bursts of 16 double data beats each.

The status of the DMA operation is available in the DMA Status Register (DMASR). The DMABSY bit equal to 1 represents that a DMA operation is underway. When equal to 0, it represents the completion of a DMA transfer. Alternatively, the DMA Done (DD) interrupt can be used to detect when a DMA operation is complete.

If an error condition is detected during a bus transfer, the DMA operation will be aborted at its current point of progress. The error is reported through the DMA Error (DE) interrupt condition and the DMA Bus Error (DBE) status bit.

Burst Behavior of the PLB Central DMA Controller

The PLB Central DMA supports both read and write bursts if it needs to read or write two or more sequential memory locations.

For read transfers, the burst transfer is completed by the slave device in the data acknowledge phase of the last data transfer. This is followed by the negation of the M_rdBurst signal for read burst operation.

For write transfers, the burst transfer is completed by the slave device in the data acknowledge phase of the last data transfer. This is followed by the negation of the M_wrBurst signal for write burst operation.

In case of early burst terminates by PLB slaves during reads and writes, the PLB Central Controller resumes the transaction from the point where it was terminated.

Interrupt and Error Condition Descriptions

DMA Interrupt Conditions

Interrupt conditions, which are established by the occurrence of interrupt events, are stored in the interrupt status register of the channel (see Table 12). Interrupt conditions can be reported, cleared and enabled.

- Reporting: Port signal IP2INTC_Irpt is active if and only if either of the interrupt conditions occur

- Clearing: Active interrupt conditions are cleared by writing a value to the interrupt status register with a 1 in the bit position to be cleared
- Enabling: Interrupts are enabled by setting the corresponding bit in the interrupt enable register (see Table 13)

Error Conditions

A DMA operation proceeds until it is complete or until it is aborted due to an error condition detected on the bus.

If completion is due to an error, the corresponding DMA BUS Error (DBE) bit of the DMA Status Register (DMASR) will be set. Additionally, the DMA Error (DE) interrupt condition is enabled. The final values of length, source address and destination address will reflect the partial completion status of the DMA operation based on the fact that there is unwritten data present in the internal data buffer of the controller.

PLB Central DMA Controller Design Parameters

To allow the user to obtain a PLB Central DMA Controller that is uniquely tailored for their system, certain features are parameterizable in the PLB Central DMA Controller design. This allows the user to have a design that only utilizes the resources required by their system and runs at the best possible performance. The features that are parameterizable in the PLB Central DMA Controller are shown in Table 1.

Table 1: **PLB Central DMA Controller Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
PLB Central DMA Controller Features					
G1	PLB address width	C_PLB_AWIDTH	32	32	integer
G2	PLB data width	C_PLB_DWIDTH	64	64	integer
G3	The base address for the DMA registers	C_BASEADDR	Valid address(1)	-	std_logic_vector
G4	The high address for the DMA registers	C_HIGHADDR	Valid address (1)	-	std_logic_vector
G5	User ID code that appears in MIR register	C_USER_ID_CODE	0 - 255	5	integer
G6	Set access type for registers SA(2), DA(2), DMACR(2) and IER(2). Reduces LUT count when set to 0.	C_READ_OPTIONAL_REGS	0 = The registers are write only 1 = The registers are read/write	1	integer
G7	Number of masters on the system	C_PLB_NUM_MASTERS	2 - 16	8	integer
G8	Set the width of the master identifier array	C_PLB_MIDWIDTH	1 - 4	3	integer

Table 1: **PLB Central DMA Controller Design Parameters *(Contd)***

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G46	Number of bits required to encode the number of PLB Masters	C_PLB_MID_WIDTH	1 - log2 (C_PLB_NUM _MASTERS)	2	integer

Notes:
1. The C_HIGHADDR should be such that the range for C_HIGHADDR - C_BASEADDR must be a power of 2 and greater than or equal to 0x3F. The C_BASEADDR value must be a multiple of 0x40.
2. Refer to Table 4

Allowable Parameter Combinations

The following are a list of allowable parameter combinations:

- The value of C_PLB_NUM_MASTERS should be equal to $2^{C\_PLB\_MIDWIDTH}$

PLB Central DMA Controller I/O Signals

Table 2 provides a summary of all PLB Central DMA Controller input/output (I/O) signals, the interfaces under which they are grouped and a brief description of the signals.

Table 2: **PLB Central DMA Controller I/O signal Descriptions**

Port	Signal Name	Interface	I/O	Initial State	Description
PLB Master Signals					
P1	PLB_Clk	PLB	I	-	PLB Clock
P2	PLB_Rst	PLB	I	-	PLB Reset
P3	M_ABus[0:C_PLB_ AWIDTH - 1]	PLB	O	0	Master address bus
P4	M_BE[0:C_PLB_ DWIDTH/8 - 1]	PLB	O	0	Master byte enables
P5	M_BusLock	PLB	O	0	Master bus lock
P6	M_wrDBus[0:C_PLB_ DWIDTH - 1]	PLB	O	0	Master write data bus
P7	M_request	PLB	O	0	Master bus request
P8	M_RNW	PLB	O	0	Master read not write
P9	M_priority[0:1]	PLB	O	0	Master bus request priority
P10	M_rdBurst	PLB	O	0	Master burst read transfer indicator
P11	M_type[0:2]	PLB	O	0	Master transfer type
P12	M_size[0:3]	PLB	O	0	Master transfer size
P13	M_wrBurst	PLB	O	0	Master burst write transfer indicator
P14	M_abort	PLB	O	0	Master abort bus request indicator
P15	M_compress	PLB	O	0	Master compressed data transfer indicator
P16	M_guarded	PLB	O	0	Master guarded transfer indicator

Table 2: **PLB Central DMA Controller I/O signal Descriptions *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P17	M_ordered	PLB	O	0	Master synchronize transfer indicator
P18	M_lockErr	PLB	O	0	Master lock error indicator
P19	M_MSize[0:1]	PLB	O	0	Master data bus size
P20	MPLB_MRdDBus[0:C_PLB_DWIDTH - 1]	PLB	I	-	PLB master read data bus
P21	MPLB_MBusy	PLB	I	-	PLB master slave busy indicator
P22	MPLB_MErr	PLB	I	-	PLB master slave error indicator
P23	MPLB_MWrBterm	PLB	I	-	PLB master terminate write burst indicator
P24	MPLB_MWrDAck	PLB	I	-	PLB master write data acknowledge
P25	MPLB_MAddrAck	PLB	I	-	PLB master address acknowledge
P26	MPLB_MRdBTerm	PLB	I	-	PLB master terminate read burst indicator
P27	MPLB_MRdDAck	PLB	I	-	PLB master read data acknowledge
P28	MPLB_MRearbitrate	PLB	I	-	PLB master bus rearbitrate indicator
P29	MPLB_MSSize[0:1]	PLB	I	-	PLB slave data bus size
P30	MPLB_MRdWdAddr[0:3]	PLB	I	-	PLB master read word address
		PLB Slave Signals			
P31	SPLB_ABus[0:C_PLB_AWIDTH - 1]	PLB	I	-	PLB address bus
P32	SPLB_type[0:2]	PLB	I	-	PLB transfer type
P33	SPLB_size[0:3]	PLB	I	-	PLB transfer size
P34	SPLB_abort	PLB	I	-	PLB abort bus request indicator
P35	SPLB_rdBurst	PLB	I	-	PLB burst read transfer indicator
P36	SPLB_wrBurst	PLB	I	-	PLB burst write transfer indicator
P37	SPLB_BE[0:C_PLB_DWIDTH/8 - 1]	PLB	I	-	PLB byte enables
P38	SPLB_wrDBus[0:C_PLB_DWIDTH - 1]	PLB	I	-	PLB write data bus
P39	SPLB_RNW	PLB	I	-	PLB read not write
P40	SPLB_PAValid	PLB	I	-	PLB primary address valid indicator
P41	SPLB_SAValid	PLB	I	-	PLB secondary address valid indicator
P42	SPLB_masterID[0:C_PLB_MIDWIDTH - 1]	PLB	I	-	PLB current master identifier
P43	SPLB_rdPrim	PLB	I	-	PLB secondary to primary read request indicator

Table 2: **PLB Central DMA Controller I/O signal Descriptions *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P44	SPLB_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P45	SPLB_busLock	PLB	I	-	PLB lock
P46	SPLB_MSize[0:1]	PLB	I	-	PLB master data bus size
P47	SPLB_compress	PLB	I	-	PLB compressed data transfer indicator
P48	SPLB_guarded	PLB	I	-	PLB guarded transfer indicator
P49	SPLB_ordered	PLB	I	-	PLB synchronize transfer indicator
P50	SPLB_lockErr	PLB	I	-	PLB lock error indicator
P51	SPLB_pendPri[0:1]	PLB	I	-	PLB pending request priority
P52	SPLB_reqpri[0:1]	PLB	I	-	PLB current request priority
P53	SPLB_pendReq	PLB	I	-	PLB pending bus request indicator
P54	Sl_addrAck	PLB	O	0	Slave address acknowledge
P55	Sl_SSize[0:1]	PLB	O	0	Slave data bus size
P56	Sl_MErr[0:C_PLB_NUM_MASTERS - 1]	PLB	O	0	Slave error indicator
P57	Sl_wait	PLB	O	0	Slave wait indicator
P58	Sl_rearbitrate	PLB	O	0	Slave rearbitrate bus indicator
P59	Sl_wrDAck	PLB	O	0	Slave write data acknowledge
P60	Sl_wrComp	PLB	O	0	Slave write transfer complete indicator
P61	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P62	Sl_rdWdAddr[0:3]	PLB	O	0	Slave read word address
P63	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
P64	Sl_rdComp	PLB	O	0	Slave read transfer complete indicator
P65	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
P66	Sl_MBusy[0:C_PLB_NUM_MASTERS - 1]	PLB	O	0	Slave busy indicator
P67	Sl_rdDBus[0:C_PLB_DWIDTH - 1]	PLB	O	0	Slave read data bus
DMA Signals					
P40	PLB_PAValid	PLB	I	-	PLB primary address valid indicator
P41	PLB_busLock	PLB	I	-	PLB lock
P42	PLB_masterID[0:C_PLB_MID_WIDTH - 1]	PLB	I	-	PLB current master indicator
P43	PLB_RNW	PLB	I	-	PLB read not write
P44	PLB_BE[0:C_PLB_DWIDTH/8 - 1]	PLB	I	-	PLB byte enables

Table 2: **PLB Central DMA Controller I/O signal Descriptions *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P45	PLB_size[0:3]	PLB	I	-	PLB transfer size
P46	PLB_type[0:2]	PLB	I	-	PLB transfer type
P47	PLB_MSize[0:1]	PLB	I	-	PLB master data bus size
P48	PLB_compress	PLB	I	-	PLB compressed data transfer indicator
P49	PLB_guarded	PLB	I	-	PLB guarded transfer indicator
P50	PLB_ordered	PLB	I	-	PLB synchronize transfer indicator
P51	PLB_lockErr	PLB	I	-	PLB lock error indicator
P52	PLB_abort	PLB	I	-	PLB abort bus request indicator
P53	PLB_ABus[0:C_PLB_AWIDTH - 1]	PLB	I	-	PLB address bus
P54	PLB_SAValid	PLB	I	-	PLB secondary address valid indicator
P55	PLB_rdPrim	PLB	I	-	PLB secondary to primary read request indicator
P56	PLB_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P57	PLB_wrDBus[0:C_PLB_DWIDTH - 1]	PLB	I	-	PLB write data bus
P58	PLB_wrBurst	PLB	I	-	PLB burst write transfer indicator
P59	PLB_rdBurst	PLB	I	-	PLB burst read transfer indicator
P60	Sl_addrAck	PLB	O	0	Slave address acknowledge
P61	Sl_wait	PLB	O	0	Slave wait indicator
P62	Sl_SSize[0:1]	PLB	O	0	Slave data bus size
P63	Sl_rearbitrate	PLB	O	0	Slave rearbitrate bus indicator
P64	Sl_MBusy[0:C_PLB_NUM_MASTERS - 1]	PLB	O	0	Slave busy indicator
P65	Sl_MErr[0:C_PLB_NUM_MASTERS - 1]	PLB	O	0	Slave error indicator
P66	Sl_wrDAck	PLB	O	0	Slave write data acknowledge
P67	Sl_wrComp	PLB	O	0	Slave write transfer complete indicator
P68	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P69	Sl_rdDBus[0:C_PLB_DWIDTH - 1]	PLB	O	0	Slave read bus
P70	Sl_rdWdAddr[0:3]	PLB	O	0	Slave read word address
P71	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
P72	Sl_rdComp	PLB	O	0	Slave read transfer complete indicator
P73	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer

Table 2: **PLB Central DMA Controller I/O signal Descriptions *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P68	IP2INTC_Irpt	PLB	O	0	DMA Interrupt
P75	PLB_Clk	PLB	I	-	PLB clock
P76	PLB_Rst	PLB	I	-	PLB reset

Parameter - Port Dependencies

The dependencies between the PLB Central DMA Controller design parameters and I/O signals are shown in Table 3.

Table 3: **Parameter-Port Dependencies**

Generic or Port	Parameter	Affects	Depends	Description
Design Parameters				
G1	C_PLB_AWIDTH	P3, P31	-	Affects the size of address bus
G2	C_PLB_DWIDTH	P4, P6, P20, P37, P38, P67	-	Affects the size of read, write data buses and also the byte enables for master and slave interfaces
G7	C_PLB_NUM_MASTERS	P56, P66	-	Affects the width of Sl_MErr and Sl_MBusy signals
G8	C_PLB_MIDWIDTH	P42	-	Affects the width of current master identifier signal
I/O Signals				
P3	M_ABus[0:C_PLB_AWIDTH - 1]	-	G1	WIdth of the master address bus depends on C_PLB_AWIDTH
P4	M_BE[0:C_PLB_DWIDTH/8 - 1]	-	G2	Width of the master byte enables depends on C_PLB_DWIDTH
P6	M_wrDBus[0:C_PLB_DWIDTH - 1]	-	G2	Width of the master write data bus depends on C_PLB_DWIDTH
P20	MPLB_MRdDBus [0:C_PLB_DWIDTH - 1]	-	G2	Width of the master read data bus depends on C_PLB_DWIDTH
P31	SPLB_ABus[0:C_PLB_AWIDTH - 1]	-	G1	WIdth of the slave address bus depends on C_PLB_AWIDTH
P37	SPLB_BE[0:C_PLB_DWIDTH/8 - 1]	-	G2	Width of the slave byte enables depends on C_PLB_DWIDTH
P38	SPLB_wrDBus[0:C_PLB_DWIDTH - 1]	-	G2	Width of the master write data bus depends on C_PLB_DWIDTH
P42	SPLB_masterID[0:C_PLB_MIDWIDTH - 1]	-	G8	Width of the PLB current master identifier array depends on C_PLB_MIDWIDTH

Table 3: **Parameter-Port Dependencies *(Contd)***

Generic or Port	Parameter	Affects	Depends	Description
P56	SI_MErr[0:C_PLB_NUM_MASTERS - 1]	-	G7	Width of the slave error indicator depends on C_PLB_NUM_MASTERS
P66	SI_MBusy[0:C_PLB_NUM_MASTERS - 1]	-	G7	Width of the slave busy indicator depends on C_PLB_NUM_MASTERS
P67	SI_rdDBus[0:C_PLB_DWIDTH - 1]	-	G2	Width of the slave read data bus depends on C_PLB_DWIDTH

PLB Central DMA Controller Register Descriptions

The PLB Central DMA Controller contains addressable registers as summarized in Table 4. Each register is addressable on a 32-bit boundary. The detailed information about these registers is provided in the following section. The register addresses are offset to the base address, C_BASEADDR.

Table 4: **PLB Central DMA Controller Register summary**

Grouping	Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
PLB Central DMA Controller	C_BASEADDR + 0[1]	RST	Write	NA	Software Reset Register
	C_BASEADDR + 0[1]	MIR	Read	Refer Table 6	Module Identification Register
	C_BASEADDR + 4	DMACR	R/W	80000008	DMA Control Register
	C_BASEADDR + 8	SA	R/W	00000000	Source Address
	C_BASEADDR + C	DA	R/W	00000000	Destination Address
	C_BASEADDR +10	LENGTH	R/W	00000000	DMA Length
	C_BASEADDR + 14	DMASR	R/W	00000000	DMA Status Register
	C_BASEADDR + 2C	ISR	Read /TOW[2]	00000000	Interrupt Status Register
	C_BASEADDR + 30	IER	R/W	00000000	Interrupt Enable Register

Notes:
1. Address shared by two unrelated functions
2. TOW = Toggle On Write. Writing a parameterizes to a bit position within the register causes the corresponding bit position in the register to toggle

Software Reset Register (RST)

The software Reset register (RST) is shown in Figure 2. It is a write only register addressed at an offset 0x0 from base address C_BASEADDR. The bit definitions of this register is as shown in Table 5.

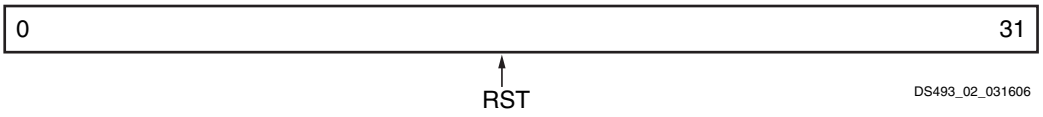

Figure 2: **Software Reset Register (RST)**

Table 5: **RST Register Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0:31	RST	Write	N/A	**Software Reset** A write of 0x0000000A causes reset of the PLB Central DMA controller. The address of the software Reset register is shared with the Module Identification Register (the Module Identification Register is a read only register)

Module Identification Register (MIR)

The Module Identification Register (MIR) is shown in Figure 3. It is a read only register addressed at an offset 0x0 from base address C_BASEADDR. The bit definitions of this register is as shown in Table 6.

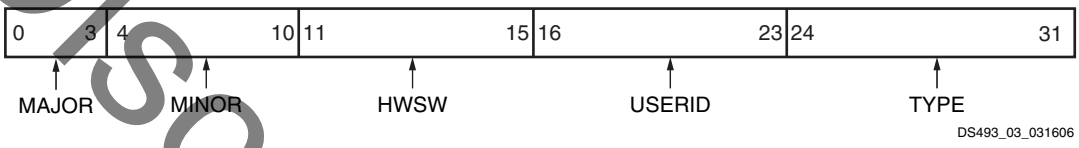

Figure 3: **Module Identification Register (MIR)**

Table 6: **MIR Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0:3	MAJOR	Read	0001	Major Revision Returns the major revision number for the core
4:10	MINOR	Read	0000000	Minor Revision Returns the minor revision number for the core
11:15	HWSW	Read	00000	Hardware/Software Compatibility Revision Returns a constant 0x0
16:23	USERID	Read	C_USER_ID_CODE	**User ID Code** Returns the 8-bit value of the C_USER_ID_CODE parameter
24:31	TYPE	Read	00000001	**DMA Channel Type** Returns a constant 0x1

DMA Control Register (DMACR)

The DMA Control Register (DMACR) is shown in Figure 4. It is a read/write register addressed at an offset 0x4 from base address C_BASEADDR. The bit definitions of this register is as shown in Table 7.

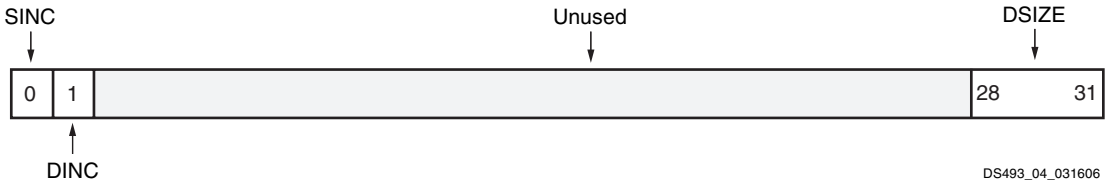

Figure 4: **DMA Control Register (DMACR)**

Table 7: **DMACR Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0	SINC	R/W	1	**Source Increment** Increment the source address by four (DSIZE) for each source word read. '1' = Increment the source address. '0' = Do not increment the source address. SINC = '0' is allowed if and only if the Source Address register is written with a *keyhole* address. See also footnote(1) on page 3.
1	DINC	R/W	0	**Destination Increment** Increment the destination address by four (DSIZE) for each destination word written. '1' = Increment the destination address '0' = Do not increment the destination address. DINC = '0' is allowed if and only if the Destination Address register is written with a *keyhole* address. See also footnote(2) on page 3.
2:27	Reserved			
28:31	DSIZE	R/W	1000	**DMA Transfer Data Size** Sets the size of the data used in each data transfer on the bus 0100 = Word (four bytes) 1000 = Double Word (eight bytes) Other values of DSIZE are invalid

Source Address Register (SA)

The Source Address register (SA) is shown in Figure 5. It is a read/write register addressed at an offset 0x8 from base address C_BASEADDR. The bit definitions of this register is as shown in Table 8.

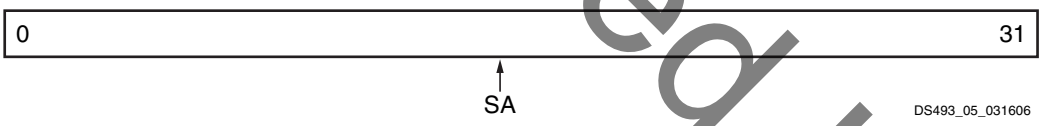

Figure 5: **Source Address Register (SA)**

Table 8: **SA Register Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0:31	SA	R/W	0	**Source Address** Source address for the current DMA operation. The address, in bytes, must be a multiple of the programmed DSIZE. The source address and destination address must align to the same DSIZE unit. When SINC = '1', as data is moved from the source address, this register updates to track the current source address. When SINC = '0', the source address remains constant at the programmed value. (See also the SINC field of the DMA Control Register).

Destination Address Register (DA)

The Destination Address register (DA) is shown in Figure 6. It is a read/write register addressed at an offset 0xC from base address C_BASEADDR. The bit definitions of this register is as shown in Table 9.

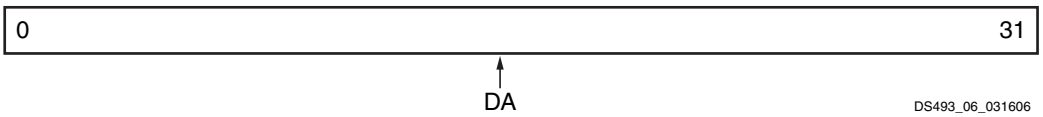

Figure 6: **Destination Address Register (DA)**

Table 9: **DA Register Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0:31	DA	R/W	0	**Destination Address** Destination address for the current DMA operation. The address, in bytes, must be a multiple of the programmed DSIZE. The source address and destination address must align to the same DSIZE unit. When DINC = '1', as data is moved to the destination address, this register updates to track the current destination address. When DINC = '0', the destination address remains constant at the programmed value. (See also the DINC field of the DMA Control Register.)

Length Register (LENGTH)

Length Register (LENGTH) is shown in Figure 7. The Length Register (LENGTH) is read/write register addressed at an offset 0x10 from base address C_BASEADDR. The bit definitions of this register is as shown in Table 10.

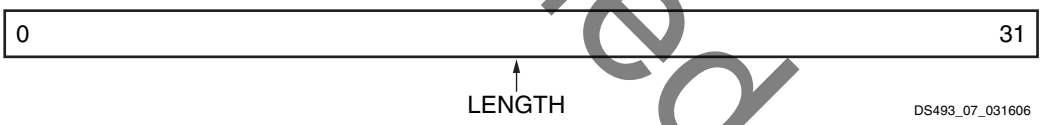

Figure 7: **Length Register (LENGTH)**

Table 10: **LENGTH Register Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0:31	LENGTH	R/W	0	Length of the DMA Transfer This parameter passes information into and out of a DMA operation. The DMA operation starts by writing into this register and this register is written after Source Address and Destination Address registers are written, During the DMA operation, this register has the number of bytes of the DMA transfer yet to be transferred. It should be noted that the number of actual data transfers is dependent on the value of the DSIZE field of the DMA Control Register, i.e. if DSIZE = "1000", then each data transfer is Double Word (eight bytes). When the value of Length is < DSIZE, one more data transfer of DSIZE will occur.

DMA Status Register (DMASR)

The DMA Status Register (DMASR) is shown in Figure 8. It is a read only register addressed at an offset 0x14 from base address C_BASEADDR. The bit definitions of this register is as shown in Table 11.

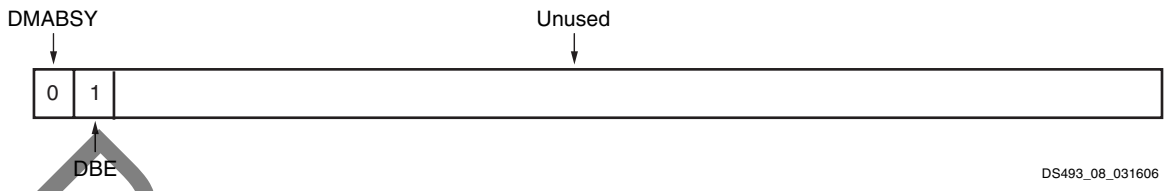

Figure 8: **DMA Status Register (DMASR)**

Table 11: **DMASR Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0	DMABSY	Read	0	DMA Busy 0 = DMA operation is not in progress 1 = DMA operation is in progress
1	DBE	Read	0	DMA Bus Error 0 = No DMA bus error 1 = DMA bus error
2:31	Reserved			

Interrupt Status Register (ISR)

The Interrupt Status Register (ISR) is shown in Figure 9. It is a read/toggle on write register addressed at an offset 0x2C from base address C_BASEADDR. The bit definitions of this register is as shown in Table 12..

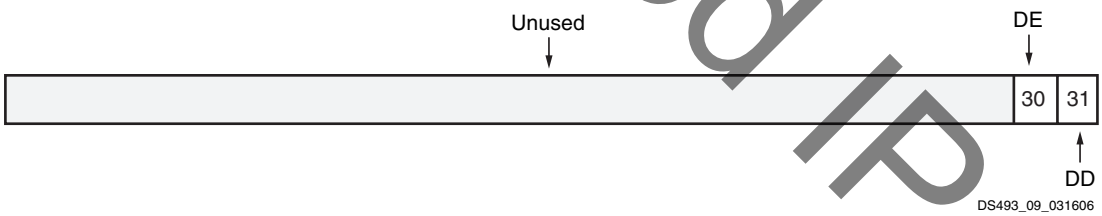

Figure 9: **Interrupt Status Register (ISR)**

Table 12: **ISR Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0:29	Reserved			
30	DE	R/TOW(1)	0	DMA Error 0 = DMA error has not occurred 1 = DMA error has occurred
31	DD	R/TOW(1)	0	DMA Done 0 = DMA operation is not done 1 = DMA operation is done

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

Interrupt Enable Register (IER)

The Interrupt Enable Register (IER) is shown in Figure 10. It is a read/write register addressed at an offset 0x30 from base address C_BASEADDR. The bit definitions of this register is as shown in Table 13.

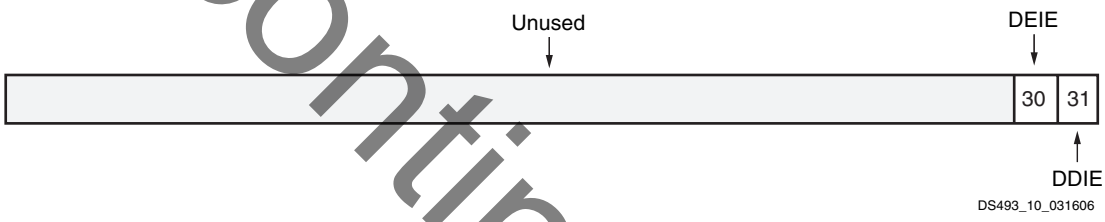

Figure 10: **Interrupt Enable Register (IER)**

Table 13: **IER Bit Definitions**

Bits	Name	Core Access	Reset Value	Description
0:29	Reserved			
30	DEIE	R/W	0	DMA Error Interrupt Enable Interrupt enable bit for DMA error. The interrupt pin would be driven only when this bit is set 0 = Interrupt is not enabled, but the DMA error bit in the DMA status register gets updated when an error occurs 1 = Interrupt is enabled
31	DDIE	R/W	0	DMA Done Interrupt Enable Interrupt enable bit for DMA done. The interrupt pin would be driven only when this bit is set 0 = Interrupt is not enabled, but the DMA done bit in the DMA status register gets updated when DMA operation is done 1 = Interrupt is enabled

Timing Diagrams

The following diagrams illustrate the PLB Central DMA operation for various read and write transactions of different lengths.

1. Read and write transactions of a 32-bit aligned DMA transfer of length 8 bytes are shown in Figure 11 and Figure 12.

2. Read and write transactions of a 64-bit aligned DMA transfer of length 8 bytes are shown in Figure 13 and Figure 14.

3. Read and write transactions of a 32-bit aligned DMA transfer of length 32 bytes are shown in Figure 15 and Figure 16.

4. Read and write transactions of a 64-bit aligned DMA transfer of length 32 bytes are shown in Figure 17 and Figure 18.

5. Read and write transactions of a time-out case in a 64-bit aligned DMA transfer of length 8 bytes are shown in Figure 19 and Figure 20.

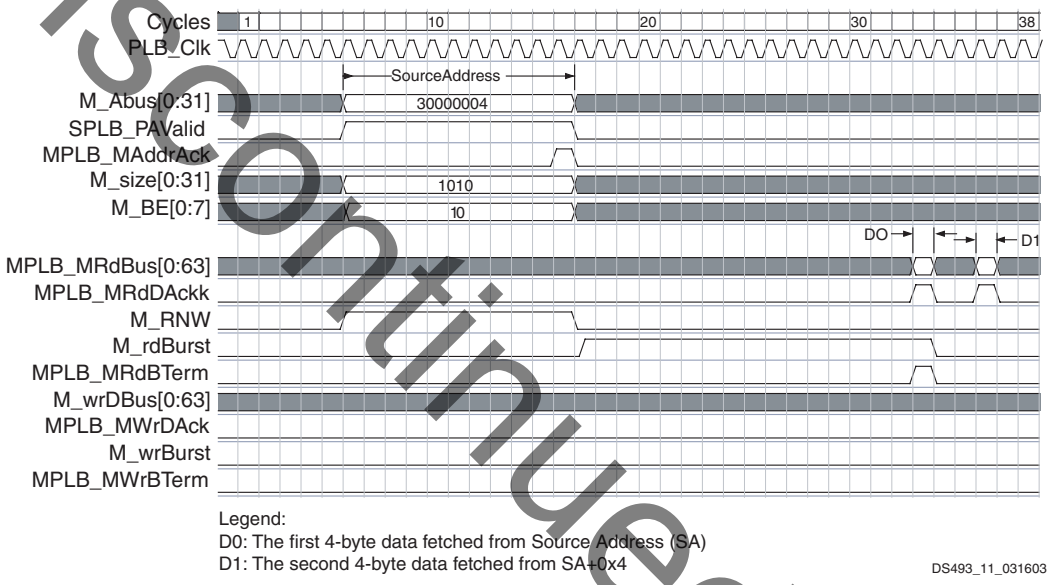

Figure 11: **Read Transaction for 32-bit Aligned DMA Transfer of Length 8**

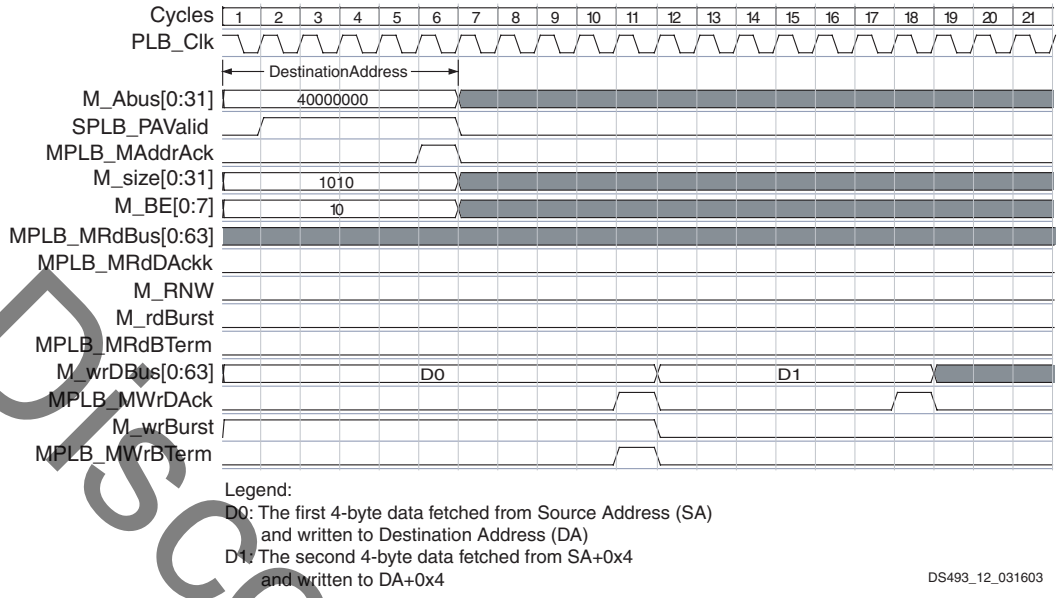

Figure 12: **Write Transaction for 32-bit Aligned DMA Transfer of Length 8**

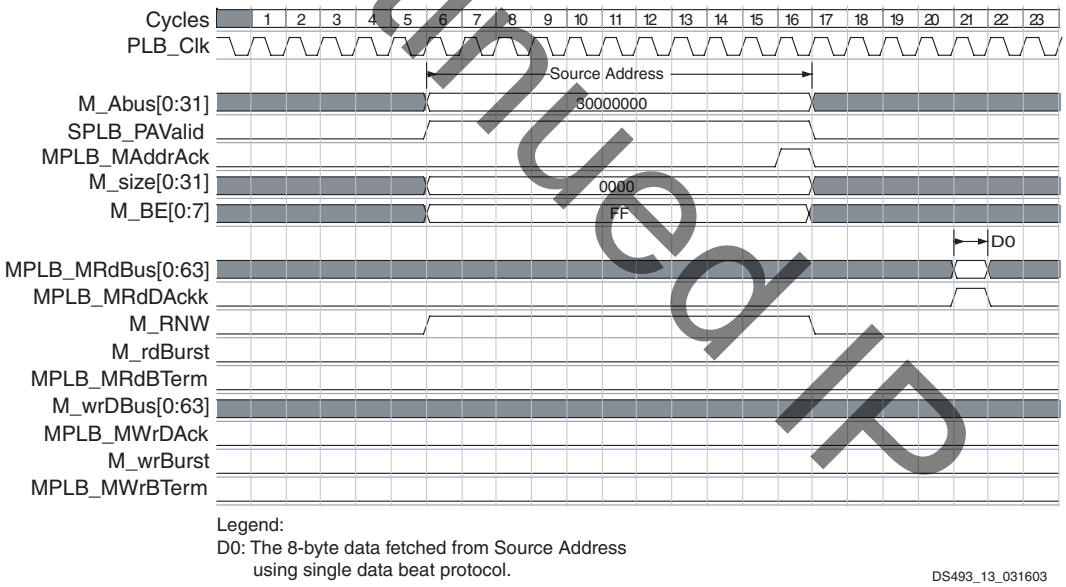

Figure 13: **Read Transaction for 64-bit Aligned DMA Transfer of Length 8**

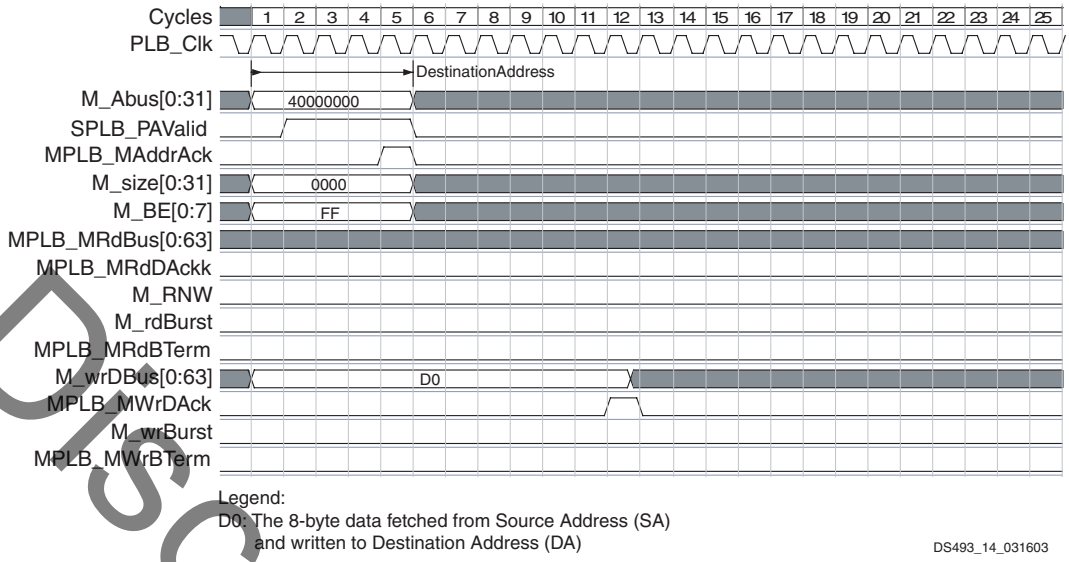

Figure 14: **Write Transaction for 64-bit Aligned DMA Transfer of Length 8**

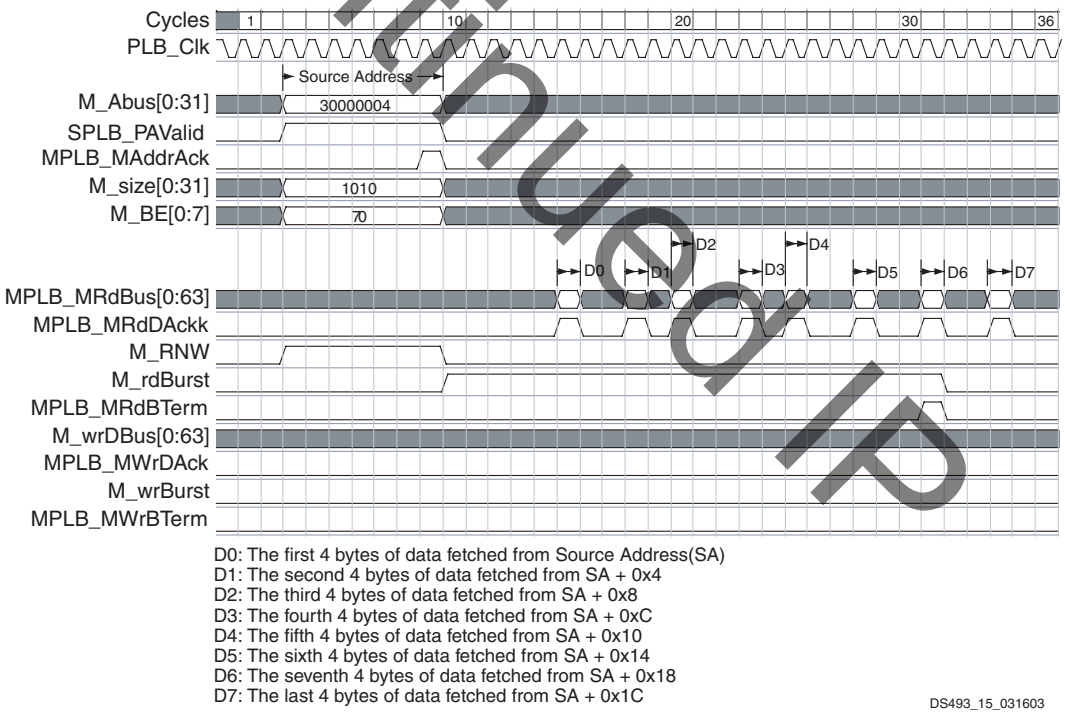

Figure 15: **Read Transaction for 32-bit Aligned DMA Transfer of Length 32**

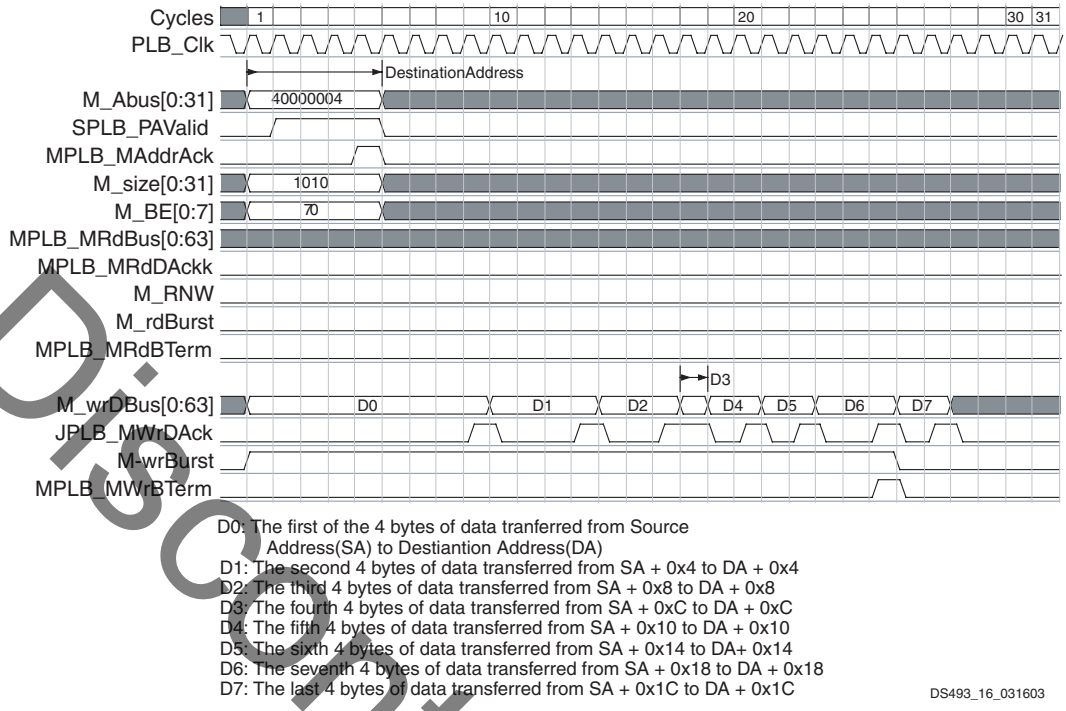

Figure 16: **Write Transaction for 32-bit Aligned DMA Transfer of Length 32**

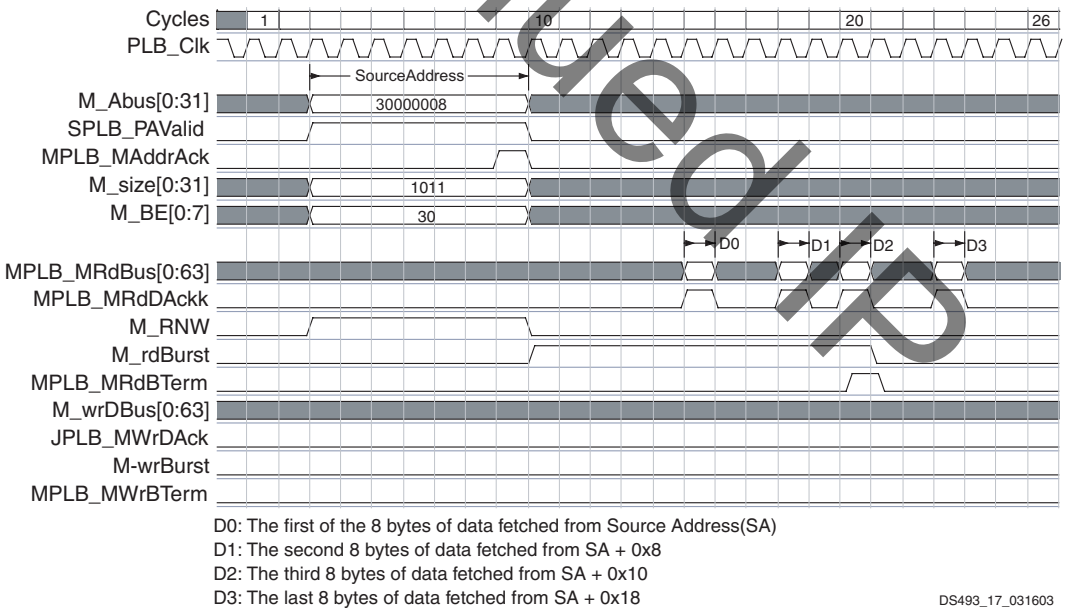

Figure 17: **Read Transaction for 64-bit Aligned DMA Transfer of Length 32**

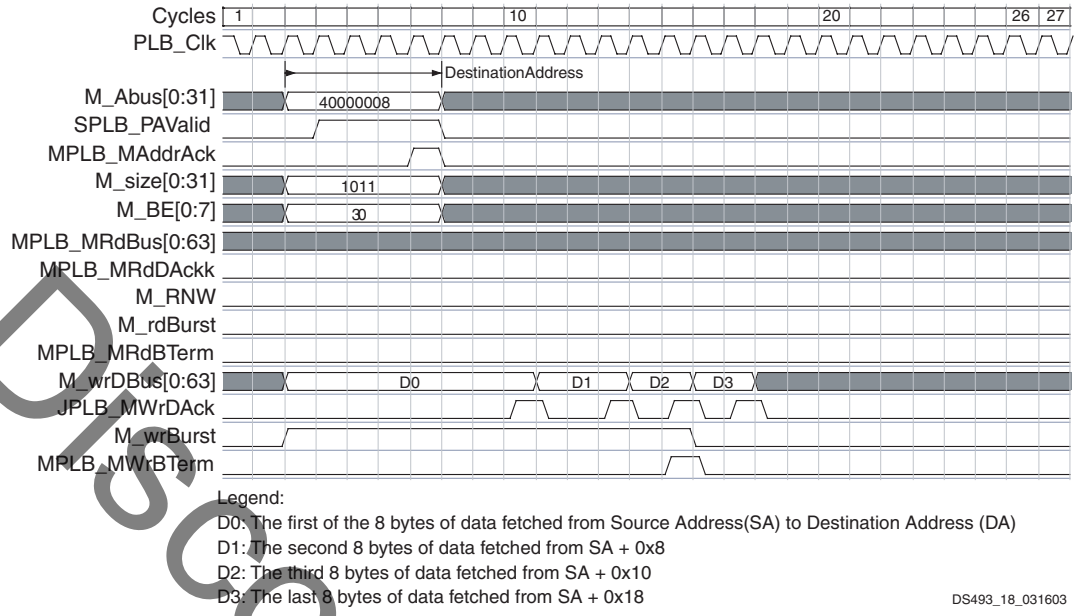

Figure 18: **Write Transaction for 64-bit Aligned DMA Transfer of Length 32**

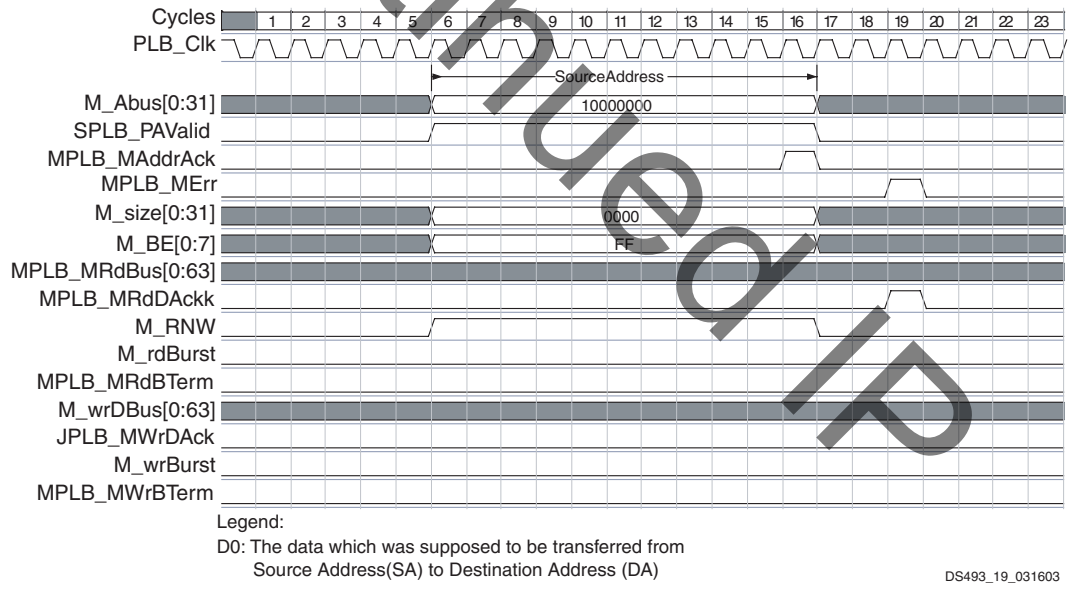

Figure 19: **Read Transaction in the Time-out Case**

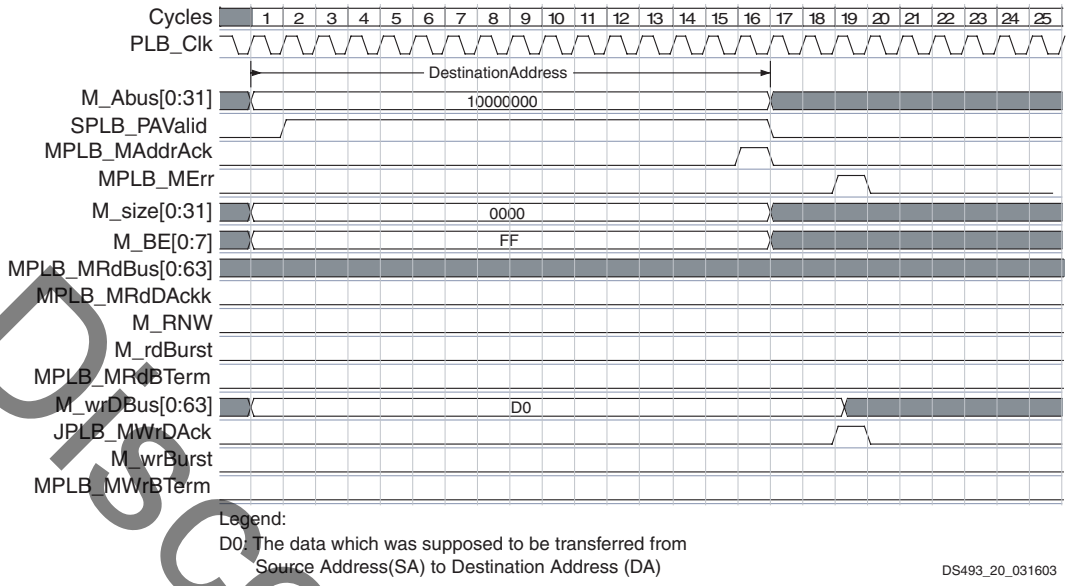

Figure 20: **Write Transaction in the Time-out Case**

Design Implementation

Target Technology

The intended target technologies are the Virtex-II Pro and Virtex-4 device.

Device Utilization and Timing

The PLB Central DMA may be parameterized so that all registers are readable. Alternatively, the Source Address, Destination Address, DMA Control and Interrupt Enable registers may be configured as write-only, to save resources. If the C_READ_OPTIONAL_REGS parameter is set to 1, the Source Address, Destination Address, DMA Control and Interrupt Enable registers are configured as read/write registers. If the C_READ_OPTIONAL_REGS parameter is set to 0, these registers are write-only, resulting in a reduction of logic utilization.

The PLB_Clk is capable of running at 100 MHz.

Performance Benchmarks

Table 14: **Performance and Resource Utilization for PLB Central DMA Controller (Virtex-II Pro -6)**

Parameter Values	Device Resources			f_{MAX} (MHz)
C_READ_OPTIONAL_REGS	Slices	Slice Flip- Flops	4-input LUTs	f_{MAX}
1	374	171	573	100.7
0	367	170	531	102.2

Table 15: **Performance and Resource Utilization for PLB Central DMA Controller (Virtex-4 -10)**

Parameter Values	Device Resources			f_{MAX} (MHz)
C_READ_OPTIONAL_REGS	Slices	Slice Flip- Flops	4-input LUTs	f_{MAX}
1	389	171	573	115.5
0	364	170	531	123.7

Reference Documents

(1) *64-Bit IBM Processor Local Bus (PLB) Architecture Specification Version 3.5*

(2) *PLB IPIF Design Specification DS458, Revision v1.00.f*

Revision History

Date	Version	Revision
12/20/04	1.0	Initial release
03/13/06	1.1	Updated for 32-bit address alignment when DSIZE = 4. Added timing diagrams.
03/21/06	1.2	Converted to new DS template; figures updated to graphic standards.
03/12/07	1.3	Modified the descriptions of SINC, DINC, Source Address and Destination Address.