

Introduction

The OPB Central DMA Controller provides simple Direct Memory Access (DMA) services for peripherals and memory devices on the OPB bus. The controller moves 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 transfer length, source address, destination address, and dataword size
- Byte, halfword, and word data sizes supported
- Supports "keyhole" addressing (word data size, only)
- Provides fast internal data buffer to support OPB burst transfers

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	QPro™-R Virtex™-II, QPro Virtex-II, Spartan™-II, Spartan-II-E, Spartan-3, Spartan-3E, Virtex, Virtex-II, Virtex-II Pro, Virtex-4, Virtex-E	
Version of Core	opb_central_dma	v1.00c
Resources Used		
	Min	Max
Slices	187	215
LUTs	307	357
FFs	271	282
Block RAMs	0	0
Provided with Core		
Documentation	Product Specification	
Design File Formats	N/A	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	N/A	
Design Tool Requirements		
Xilinx Implementation Tools	N/A	
Verification	N/A	
Simulation	N/A	

DMA Operation

A DMA operation is set up and started by writing values into the following DMA registers:

- **DMACR.** The bits of the DMA Scatter Gather Control Register are set to values for the desired operation:
 - **SINC.** If the Source Address is a "keyhole" register—a single address associated with a sequence of values—SINC is set to 0. If the Source Address should increment for each byte transferred, SINC is set to 1. Please note that the non-incrementing option (SINC = 0) is supported only for DSIZE = 4.
 - **DINC.** Is set under the same considerations as SINC, but for the Destination Address.
 - **DSIZE.** Is set to 1, 2, or 4 to select the data size, in bytes, to be used for individual bus transfers
- **SA.** The source address for the transfer is written to the Source Address register. (Please see [Table 7 on page 8](#) for alignment restrictions.)
- **DA.** The destination address for the transfer is written to the Destination Address register. (Please see [Table 8 on page 8](#) for alignment restrictions.)
- **LENGTH.** The number of bytes to transfer is written to the Length register. Writing of this register is the event that starts the DMA operation, so it must be done last¹. (Please see [Table 9 on page 9](#) for padding behavior when LENGTH is not a multiple of DSIZE.)

Once started, the DMA operation proceeds by reading source-address data into the internal data buffer, then writing the data from the internal buffer to the destination address. This repeats until all data is moved. The registers update as the DMA operation progresses.

The status of the DMA operation is available in the DMA Status Register (DMASR). The DMABSY bit is '1' while a DMA operation is underway and returns to '0' upon completion. 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 is aborted at its current point of progress. The error is reported through the DMA Error (DE) interrupt condition and the DMA Bus Error (DBE) or DMA Bus Timeout (DBT) status bit.

More detail is provided in the descriptions that follow. (Register descriptions, [page 5](#); Interrupt usage, [page 11](#); and Error conditions, [page 11](#))

Operation from the OPB Viewpoint

The Central DMA Controller operates on the OPB as a master/slave device. It responds as a slave when its registers are being read and written and it initiates as a master when a DMA operation has been started.

While it is moving data as the bus master, the DMA Controller attempts to move data efficiently. Nevertheless, there are these built-in limitations:

- Each datum will cross the bus twice, once while being read from the source into the DMA Controller's buffer and once while being written from the buffer to the destination. (No "fly by" mode.)
- The DMA Controller's ability to move a datum each clock is limited by the response behavior of the source and destination slaves. (More on this later in the discussion of burst behavior.)
- Each bus transaction moves only the amount of data given by DSIZE. For example, if DSIZE is one, then each bus transaction moves only a byte.

Burst Behavior

The OPB does not have explicit burst modes of operation. However, it is possible to achieve burst-like performance during DMA operations if the source or destination slave devices are capable of response on consecutive cycles. The pace of OPB transactions is determined by the addressed slave in all cases. The duration of a transaction is from the assertion of OPB_select until the assertion of OPB_xferAck. The slave is allowed to cause assertion of OPB_xferAck on the same cycle that the master causes assertion of OPB_select, so a transaction lasting one cycle is possible. Once the slave, possibly following a setup latency, begins to assert immediate acknowledges and the master immediately issues the next transaction, burst performance is achieved.

1. The other registers can be written, if they need to change, in any order.

Sequential-Address Transactions. The OPB_seqAddr signal of the OPB is helpful in achieving burst-like performance. This signal may be asserted by the master if it is known that the next transaction will be the successor address of the current transaction and in the same direction, read or write. This "advance information" can be key to a slave being able to attain a state that allows it to acknowledge immediately, particularly for reads.

The Central DMA Controller always asserts OPB_seqAddr if it will be moving two or more DSIZE units to or from the buffer. It also respects the OPB protocol requirement to assert another signal, OPB_busLock, when it asserts OPB_seqAddr. Bus locking is necessary because otherwise the master cannot guarantee in a multi-master environment that the next bus transactions will be its successor transaction. The DMA Controller also respects a Xilinx-imposed requirement that the bus-locked sequence always have a final transaction for which OPB_seqAddr is negated.¹

As a final point, please note that the DMA Controller also asserts OPB_seqAddr for transactions where the source or destination is not incrementing (SINC=0 or DINC=0) and a constant rather than sequential address is presented to the bus. This is a technical violation of the OPB protocol, but not one that affects correct operation of current EDK devices. This behavior is under review and may result in changes to a future version of the DMA Controller.

DMA Block Diagram

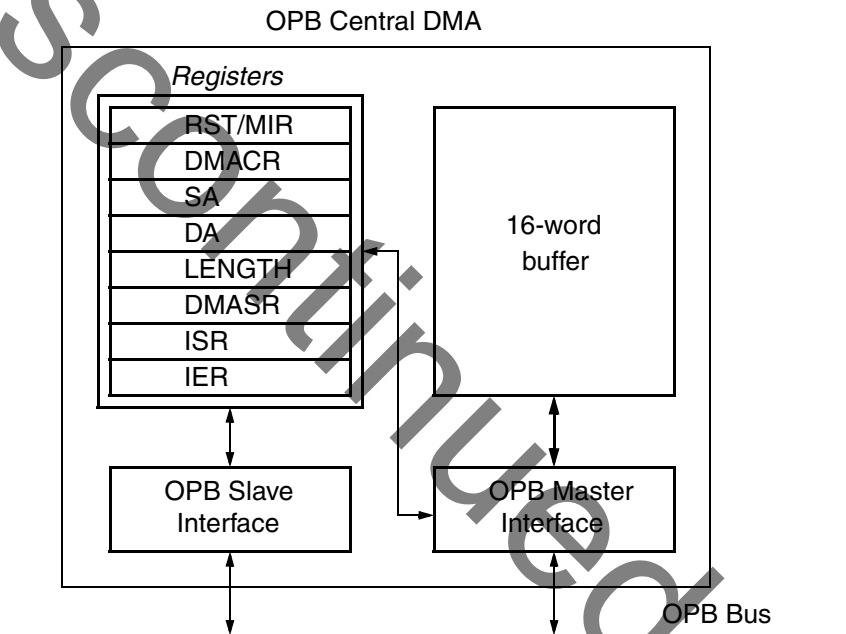


Figure 1: OPB Central DMA Block Diagram

OPB Central DMA Controller Parameters

To allow you to obtain a OPB Central DMA Controller that is uniquely tailored for your system, certain features can be parameterized. This allows you to have a design that only utilizes the resources required. The features that can be parameterized in the Xilinx OPB Central DMA Controller design are shown in the following table:

1. The reason for this requirement is to give slaves an advance warning that a "burst" is ending. The early negation of OPB_seqAddr also allows arbitration for the next master to begin one cycle earlier.

Table 1: OPB Central DMA Controller Design Parameters

Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
OPB address width	C_OPB_AWIDTH	32	32	integer
OPB data width	C_OPB_DWIDTH	32	32	integer
The base address for the DMA registers	C_BASEADDR ⁽¹⁾	A valid address, with unused leftmost bits set to zero	zero	std_logic_vector(0 to 31)
The high address for the DMA registers	C_HIGHADDR	A valid address, with unused leftmost bits set to zero	zero	std_logic_vector(0 to 31)
User ID code that appears in MIR register	C_USER_ID_CODE	0 - 255	0	integer
Sets access type for registers SA, DA, DMACR, and IER. Reduces LUT count when set to 0.	C_READ_OPTIONAL_REGS	0 = SA, DA, DMACR, and IER are write only 1 = SA, DA, DMACR, and IER are read/write	0	integer

Notes:

1. C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR + 1.

OPB Central DMA Controller I/O Signals

The I/O signals for the OPB Central DMA Controller are listed in Table 2. The interfaces referenced in this table are shown in Figure 1 in the OPB Central DMA block diagram.

Table 2: OPB Central DMA I/O Signals

Signal Name	MSB:LSB	I/O	Description
<i>Master-Side signals</i>			
MOPB_Clk		I	OPB Clk (Master side)
MOPB_Rst		I	OPB Reset (Master side)
M_ABus	0:C_OPB_AWIDTH – 1	O	Master Address Bus
M_BE	0:C_OPB_DWIDTH/8 – 1	O	Master Byte Enables
M_busLock	0:C_NUM_MASTERS-1	O	Master Buslock
M_DBus	0:C_OPB_DWIDTH – 1	O	Master Databus
M_request		O	Master Request
M_RNW		O	Master Read/ Not Write
M_select		O	Master Select
M_seqAddr		O	Master Sequential Address
MOPB_DBus	0:C_OPB_DWIDTH – 1	I	OPB Data Bus
MOPB_errAck		I	OPB Error Acknowledge
MOPB_MGrant		I	OPB Master Grant
MOPB_retry		I	OPB Retry
MOPB_timeout		I	OPB Timeout
MOPB_xferAck		I	OPB Transfer Acknowledge
<i>Slave-Side signals</i>			

Table 2: OPB Central DMA I/O Signals (Continued)

Signal Name	MSB:LSB	I/O	Description
SOPB_Clk		I	OPB Clk (Master side)
SOPB_Rst		I	OPB Reset (Master side)
SOPB_ABus	0:C_OPB_AWIDTH – 1	I	OPB Address Bus
SOPB_BE	0:C_OPB_DWIDTH/8 – 1	I	OPB Byte Enables
SOPB_DBus	0:C_OPB_DWIDTH – 1	I	OPB Data Bus
SOPB_RNW		I	OPB Read/ Not Write
SOPB_select		I	OPB Select
SOPB_seqAddr		I	OPB Sequential Address
SI_DBus	0:C_OPB_DWIDTH – 1	O	Slave Data Bus
SI_errAck		O	Slave Error Acknowledge
SI_retry		O	Slave Retry
SI_toutSup		O	Slave Timeout Suppress
SI_xferAck		O	Slave Transfer Acknowledge
<i>DMA signals</i>			
DMA_Req		I	DMA Request (reserved for future use)
DMA_Ack		O	DMA Acknowledge (reserved for future use)
DMA_Interrupt		O	DMA Interrupt

OPB Central DMA Controller Register Descriptions

The OPB Central DMA Controller contains addressable registers for read/write operations as summarized in [Table 3](#) and shown in detail in the register descriptions that follow. The register addresses are relative to the base address, C_BASEADDR.

Each register is addressable on a 32-bit boundary.

Table 3: OPB Central DMA Controller Registers

Register Name	Abbrev	Address (as an offset from C_DMA_BASEADDR)	Access
Channel Reset	RST	0x0 ⁽¹⁾	Write
Module Identification Register	MIR		Read
DMA Control Register	DMACR	0x4	Read/Write
Source Address	SA	0x8	Read/Write
Destination Address	DA	0xC	Read/Write
DMA Length	LENGTH	0x10	Read/Write
DMA Status Register	DMASR	0x14	Read
Interrupt Status Register	ISR	0x2C	Read/TOW ⁽²⁾
Interrupt Enable Register	IER	0x30	Read/Write

Notes:

1. Address shared by two un-related functions
2. TOW means Toggle On Write: toggle each bit position to which a "1" is written.

Software Reset Register (RST)

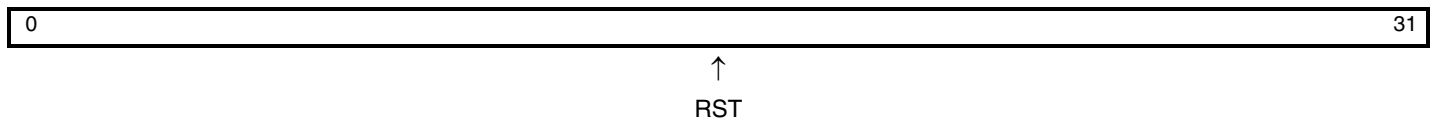


Figure 2: RST

Table 4: Software Reset Register (RST) - Write Only - Address offset 0x0

Bits	Name	Description	Reset Value
0:31	RST	Software Reset (write only) A write of 0x0000000A causes a reset of the OPB Central DMA. The address of the software reset is shared with the Module Identification Register, which is read only.	NA

Module Identification Register (MIR)

)

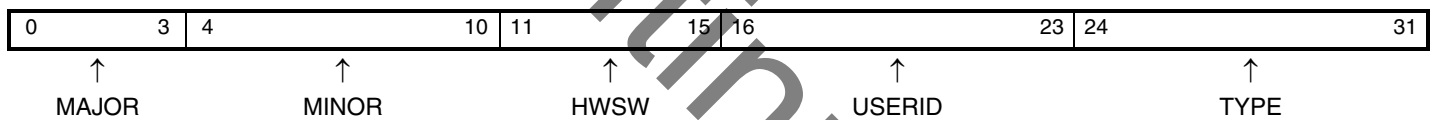


Figure 3: MIR

Table 5: Module Identification Register (MIR) - Read Only - Address offset 0x0

Bits	Name	Description	Reset Value
0:3	MAJOR	Major Revision Returns the major revision number for the core.	0001
4:10	MINOR	Minor Revision Returns the minor revision number for the core.	0000000
11:15	HWSW	Hardware/Software Compatibility Revision Returns a constant 0.	00010
16:23	USERID	User ID Code Returns the 8-bit value of the C_USER_ID parameter.	C_USER_ID
24:31	TYPE	DMA Channel Type Returns a constant 1.	00000001

DMA Control Register (DMACR)

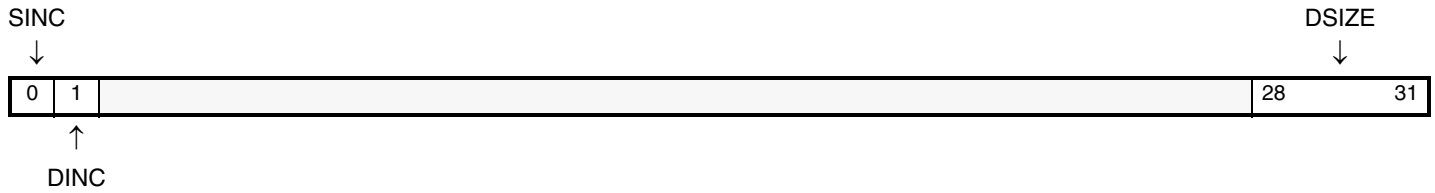


Figure 4: DMACR

Table 6: DMA Control Register (DMACR) - Address offset 0x4

Bits	Name	Description	Reset Value
0	SINC	Source Increment Increment the source address by DSIZE for each data transfer on the bus. The source address can be set to not increment for use with keyhole address devices such as FIFOs. 0 Do not increment the source address (allowed only if DSIZE = 4) 1 Increment the source address	1
1	DINC	Destination Increment Increment the destination address by DSIZE for each data transfer on the bus. The destination address can be set to not increment for use with keyhole address devices such as FIFOs. 0 Do not increment the destination address (allowed only if DSIZE = 4) 1 Increment the destination address	0
2:27	Reserved		
28:31	DSIZE	DMA Transfer Data Size Sets the size of the data used in each data transfer on the bus. 0001 Byte 0010 Halfword (two bytes) 0100 Word (four bytes) Other values of DSIZE are invalid	0100

Source Address Register (SA)

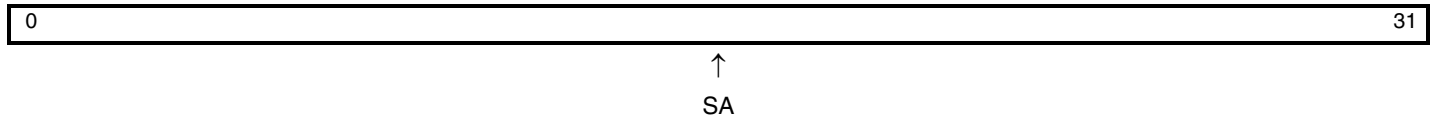


Figure 5: SA

Table 7: SA Register (SA) - Address offset 0x8

Bits	Name	Description	Reset Value
0:31	SA	Source Address Source address for the current DMA operation. The address, in bytes, must be a multiple of the programmed DSIZE. Further, the source address and the destination address must align to the same DSIZE unit within a 32-bit word. The source address updates during the DMA transfer to indicate the current source address.	0

Destination Address Register (DA)

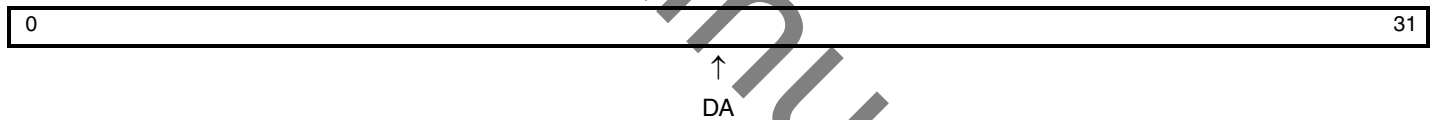


Figure 6: DA

Table 8: DA Register (DA) - Address offset 0xC

Bits	Name	Description	Reset Value
0:31	DA	Destination Address Destination address for the current DMA operation. The address, in bytes, must be a multiple of the programmed DSIZE. Further, the source address and the destination address must align to the same DSIZE unit within a 32-bit word. The destination address updates during the DMA transfer to indicate the current destination address.	0

Length Register (LENGTH)

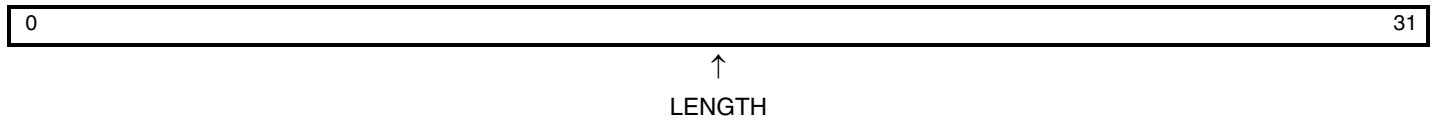


Figure 7: LENGTH

Table 9: Length Register (LENGTH) - Address offset 0x10

Bits	Name	Description	Reset Value
0:31	LENGTH	<p>Length of the DMA Transfer</p> <p>This parameter passes information into and out of a DMA operation. Prior to the operation it is set to the requested number of bytes to transfer. After the operation it represents the number, if any, of the requested bytes that did not transfer. Writing to this register starts the DMA operation, so it should be written last when setting up a DMA operation.</p> <p>If the programmed LENGTH is not a multiple of DSIZE, then additional "pad" bytes are transferred to extend the overall DMA transfer to the next multiple. Pad bytes are not subtracted in the final length calculation, so a successful DMA operation always terminates with LENGTH = 0.</p>	0

DMA Status Register (DMASR)

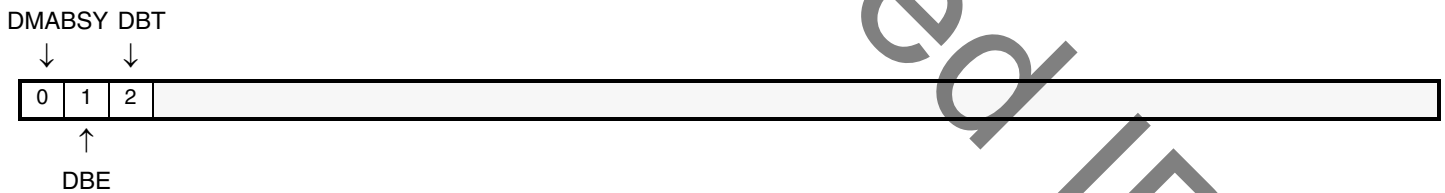


Figure 8: DMASR

Table 10: DMA Status Register (DMASR) - Address offset 0x14

Bits	Name	Description	Reset Value
0	DMABSY	<p>DMA Busy</p> <p>0 DMA operation is not in progress 1 DMA operation is in progress</p>	0
1	DBE	<p>DMA Bus Error</p> <p>0 No DMA Bus Error 1 DMA Bus Error</p>	0

Table 10: DMA Status Register (DMASR) (Continued)- Address offset 0x14

Bits	Name	Description	Reset Value
2	DBT	DMA Bus Timeout 0 No DMA Bus Timeout 1 DMA Bus Timeout	0
3:31	Reserved		

Interrupt Status Register (ISR)

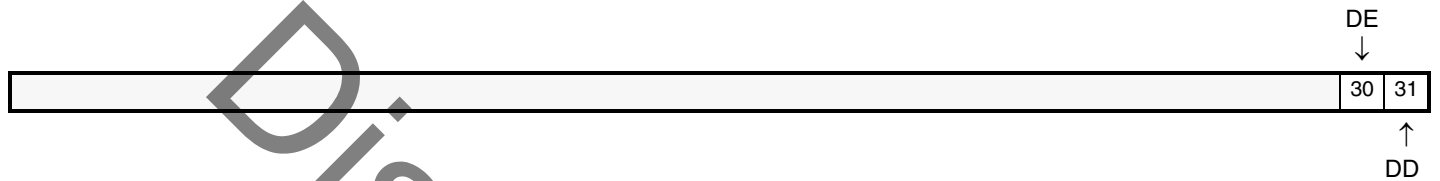


Figure 9: ISR

Table 11: Interrupt Status Register (ISR) - Address offset 0x2C

Bits	Name	Description	Reset Value
0:29	Reserved		
30	DE	DMA Error 0 DMA error has not occurred 1 DMA error has occurred	0
31	DD	DMA Done 0 DMA operation is not done 1 DMA operation is done	0

Interrupt Enable Register (IER)

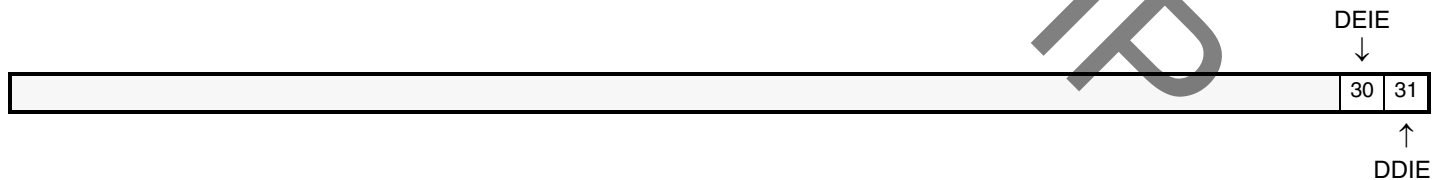


Figure 10: IER

Table 12: Interrupt Enable Register (IER) - Address offset 0x30

Bits	Name	Description	Reset Value
0:29	Reserved		

Table 12: Interrupt Enable Register (IER) (Continued)- Address offset 0x30

Bits	Name	Description	Reset Value
30-31		Interrupt-enable bits for the interrupts defined in the ISR. The interrupt enable is in the same bit position as the corresponding interrupt status bit. 0 Interrupt is not enabled 1 Interrupt is enabled	0

OPB Central DMA Controller Interrupt Descriptions

DMA Interrupt Conditions

Table 13 enumerates the interrupt conditions for the central DMA.

Table 13: OPB Central DMA Controller Interrupt Conditions

Interrupt	Description
DMA Done	DMA operation is complete (with or without error)
DMA Error	DMA operation is complete with error

Reporting, Clearing and Enabling Interrupts

Interrupt conditions, which are established by the occurrence of *interrupt events*, are stored in the channel's Interrupt Status Register (see Table 11). Interrupt conditions can be reported, cleared, and enabled.

- Reporting: Port signal DMA_Interrupt is active if and only if either of the interrupt conditions is active and enabled.
- Clearing: Active interrupt conditions are cleared by writing a value to the Interrupt Status Register with a '1' in the bit position to be cleared¹.
- Enabling: Interrupts are enabled by setting the corresponding bit in the Interrupt Enable Register (see Table 12). The interrupt enable bit controls only whether an active interrupt will cause DMA_Interrupt to assert, not whether the interrupt condition will be made active in the Interrupt Status Register by the occurrence of the interrupt event.

Error Conditions

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

If completion is due to an error or timeout, the corresponding DMA BUS Error (DBE) or DMA BUS Timeout (DBT) bit of the DMA Status Register will be set. Additionally, the DMA Error (DE) interrupt condition will become active. The final values of LENGTH, SA and DA will reflect the partial-completion status of the DMA operation, taking into account that unwritten data may be present in the controller's internal data buffer.

Resource Utilization

The OPB Central DMA may be parameterized so that all registers are readable, or, to save resources, the SA, DA, DMACR, and IER registers may be configured as write-only. If the C_READ_OPTIONAL_REGS parameter is set to '1', then the SA, DA, DMACR, and IER registers are read/write. If the C_READ_OPTIONAL_REGS parameter is set to '0', then these registers are write-only, resulting in a savings of approximately 50 LUTs.

1. If a '1' is written to the bit position of an interrupt that is not active, the interrupt active bit is set. This is a behavior that can be useful for software testing.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/27/03	1.0	Initial release
07/10/03	1.1	Update for EDK Granite
10/29/03	1.2	Added description of (1) restrictions related to DSIZE, (2) transfer with pad, and (3) behavior on bus error. Other minor changes.
02/04/04	1.3	Corrected some signal directions in the I/O signals list.
03/16/04	1.4	Described required mutual alignment of SA and DA to the same DSIZE unit within a word. Added description of operation from OPB viewpoint. Updated the MIR SWHW field reset value to 00010 (Corresponding to the "c" in v1.00.c).
8/19/04	1.5	Updated for Gmm; updated trademarks and supported device family listing.
9/10/04	1.5.1	In Table 2, removed "reserved for future use" from DMA_interrupt signal description.
2/4/05	1.6	Added restriction that non-incrementing address can be used only if data is programmed to move in four-byte units.
4/4/05	1.7	Updated for EDK 7.1.1 SP1 release; updated trademarks and supported device listing.
12/1/05	1.8	Added Spartan-3E to supported device listing.