Channelized Direct Memory Access and Scatter Gather (v1.00a)

DS440 February 25, 2010 **Product Specification**

Introduction

This specification is for a DMA Scatter Gather controller which can scale up to a relatively large number of channels (hundreds). Many concepts from an earlier Xilinx DMA Scatter Gather controller (see Reference [1]) are retained and some are dropped or altered to cater to the needs of a channelized implementation and the use of block RAM to store channel state.

An example of a communications controller that needs a high number of DMASG channels is an HDLC controller with multiple physical channels and multiple TDM (time-division multiplexed) subchannels per physical channel. The present DMASG controller is specified with such a system, or similar one, in mind.

The Channelized Direct Memory Access and Scatter Gather controller is packaged as a service within a Xilinx IPIF bus-connection block.

Features

- Available as a service in selected IPIFs
- Supports up to 16 physical channels
- Supports up to 64 subchannels per physical channel
- Block RAM implementation limits resource usage as number of channels scales up
- Supports packet SG, simple SG and simple DMA
- Status FIFO interface for synchronization with communications controller
- Event FIFO for reporting interrupts
- Supports optional Interrupt coalescing

LogiCORE™ IP Facts		
Core Specifics		
Supported Device Family	Spartan®-3, Spartan-3E, Virtex®-4	
Version of core	chan_dma_sg	v1.00a
Resources Used		
	Min	Max
I/O	N/A	N/A
LUTs	830	960
FFs	330	370
Block RAMs	1	8
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	None	
Verification	N/A	
Instantiation Template	VHDL Wrapper	
Design Tool Requirements		
Xilinx Implementation Tools	ISE® v11.1	
Verification	Mentor Graphics ModelSim v6.4b and above	
Simulation	Mentor Graphics ModelSim v6.4b and above	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

Channelized DMASG Controller Overview

Definitions

Direct memory access (DMA) allows for a bounded number of sequential data transfers to take place between regions in the address space (typically between memory and an I/O device) without processor management of individual transfers. The processor sets up the *DMA operation* by specifying the number of bytes to be moved and the source and destination addresses.

Scatter gather (SG) allows a *sequence* of DMA operations to be pre-specified by software and performed automatically without further processor intervention. The processor prepares the DMA operations in a system of buffers and their associated *Buffer Descriptors*. The SG automation hardware processes the Buffer Descriptors and performs the DMA operations specified therein through activation of the DMA hardware.

Often it is useful to consider that one or more DMA operations combine to compose a higher-level unit of data. An example of such a unit is a packet or frame[1] of data in a communications protocol. This specification addresses issues associated with the handling of packets and allows packets to be distributed across one or more buffers.

This document specifies four types of DMA channels as shown in Table 1. Support for packet SG channels in high numbers is the primary motivation for developing the Channelized DMASG Controller.

Table 1: **Types of Channelized DMA [SG] Channels**

0	Simple DMA	A single DMA operation
1	Simple SG	A sequence of simple DMA operations performed automatically
2	Packet-transmit SG	Simple SG plus delimiting, status reporting, and optional interrupt coalescing for outbound packets
3	Packet-receive SG	Simple SG plus delimiting, status reporting, and optional interrupt coalescing for inbound packets

Distributed DMA Architecture

A common DMA organization is *centralized DMA* in which devices share a common *DMA controller.* The DMA controller serves as an intermediate station between the device and memory. The DMA controller operates as a bus master.

In contrast, this specification provides for a *distributed DMA* organization in which dedicated DMA and SG automation hardware are present in the device. At the same time, the specification does not preclude the possibility of a centralized DMA implementation or of a distributed DMA controller in one device having channels that service a non-local device.

The following table contrasts centralized and distributed DMA.

[1] This document will use the term packet but the term frame is also widely used.

Table 2: **Comparison of Distributed and Centralized DMA**

Aspect	Centralized DMA	Distributed DMA
Hardware resources	Centralized in the DMA controller	Distributed in the I/O devices
Device as bus master and/or slave	Device needs to have slave capability.	Device needs to have master and slave capability.
Bus transactions	Two per datum moved.	One per datum moved.[1]
Alternate usage	Can be used as a memory-to-memory copy engine.	
Software	The number of channels is typically fixed. If this number is fewer than the number of devices using DMA, software must manage the channels as a shared resource.	No mismatch in number of channels vs. number of channels needed.

1. However, if the device and the memory are on different buses connected by a bus bridge, distributed DMA requires two bus transactions per datum, one for each bus. If the centralized DMA controller is implemented as a specialized bus bridge, it needs the same two transactions per datum.

Operation

Next, the operation of the Channelized DMASG Controller is described, first in terms of simple DMA operations, then in terms of scatter-gather DMA operations. The descriptions make reference to the registers defined in "Channelized DMASG Controller Register Descriptions" on page 23. The reader is encouraged to reference ahead to that information as needed.

Simple DMA Operation

When used under programmed IO, a DMA operation for a channel 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[1], then SINC is set to 0. If the Source Address should increment for each byte transferred, SINC is set to 1.
  - DINC. Set under the same considerations as SINC, but for the Destination Address.
  - SLOCAL. If the Source Address is local[2] this bit is set to 1, otherwise to 0.
  - DLOCAL. Set under the same considerations as SLOCAL, but for the Destination Address.
  - L. Not used under simple DMA.
  - SGS. Not used for simple-DMA channels. Set to 1 when SG channels are used for simple DMA.
- SA. The source address for the transfer is written to the Source Address register.
- DA. The destination address for the transfer is written to the Destination Address register.

[1] A keyhole register is a single address associated with a sequence of values. An example is a FIFO.

[2] Addresses can be either global to the entire address space or local to the device to which the DMA SG implementation is attached. A local address needs to have only as many low-order bits of the corresponding global address as make it unique within the device.

- LENGTH. The number of bytes to transfer. Writing of this register is the event that starts the DMA operation, so it must be done last [1].

The status of the DMA operation is available in the DMA Status Register, DMASR. The DMABSY bit becomes a '1' when the DMA engine starts to process the channel, which may be delayed from the time that the LENGTH register is written. Because of this delay, the recommended way to poll for completion is to wait for the LENGTH register to become zero, rather than to wait for DMABSY to become '0, because a false completion could be detected before DMA even starts. Alternatively, interrupts on DMA Done or DMA Error can be used to indicate completion. The DMA Bus Error (DBE) and DMA Bus Timeout (DBT) status bits indicate errors. The Last bit (L) is not used under simple DMA.

Figure 1 shows the operation of a DMA channel.

Scatter Gather Buffer Descriptors

Scatter Gather requires a common memory-resident data structure that holds the list of DMA operations to be performed. This list is shared by software and the SG hardware. The Buffer Descriptor shown in Table 3 is the basis for organizing the DMA operations as a linked list.

Table 3: **Buffer Descriptor**

SR	Packet Status Register
DMACR	DMA Control Register
SA	Source Address
DA	Destination Address
LENGTH	Length
DMASR	DMA Status Register
BDA	Buffer Descriptor Address (Link to next Buffer Descriptor)

Each field of the Buffer Descriptor is four bytes in length and corresponds to either one of the DMA SG registers or, in one case, to a Device Status Register. SG automation is the process of initializing a DMA operation from the fields of the buffer descriptor, starting the operation, recording the completion status to the Buffer Descriptor, and then following the link to the next buffer descriptor and repeating the process, which is described in more detail in the next section.

The SR field serves to report the status of a packet reception and is only written by SG automation for channels of type 3. The definition of the value is user-determined, with the exception that it must be non-zero. The non-zero requirement allows the SR field to serve as an indicator that a packet has been handled by hardware [2].

The Buffer Descriptors organized into a linked list are shown in Figure 2. Scatter Gather hardware will successively perform the DMA operations specified in the descriptors up to and including the descriptor with SGS=1 (all others in the figure are assumed to have SGS=0).

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

[2] Software must initialize the bit to 0 if it is to serve this purpose.

.

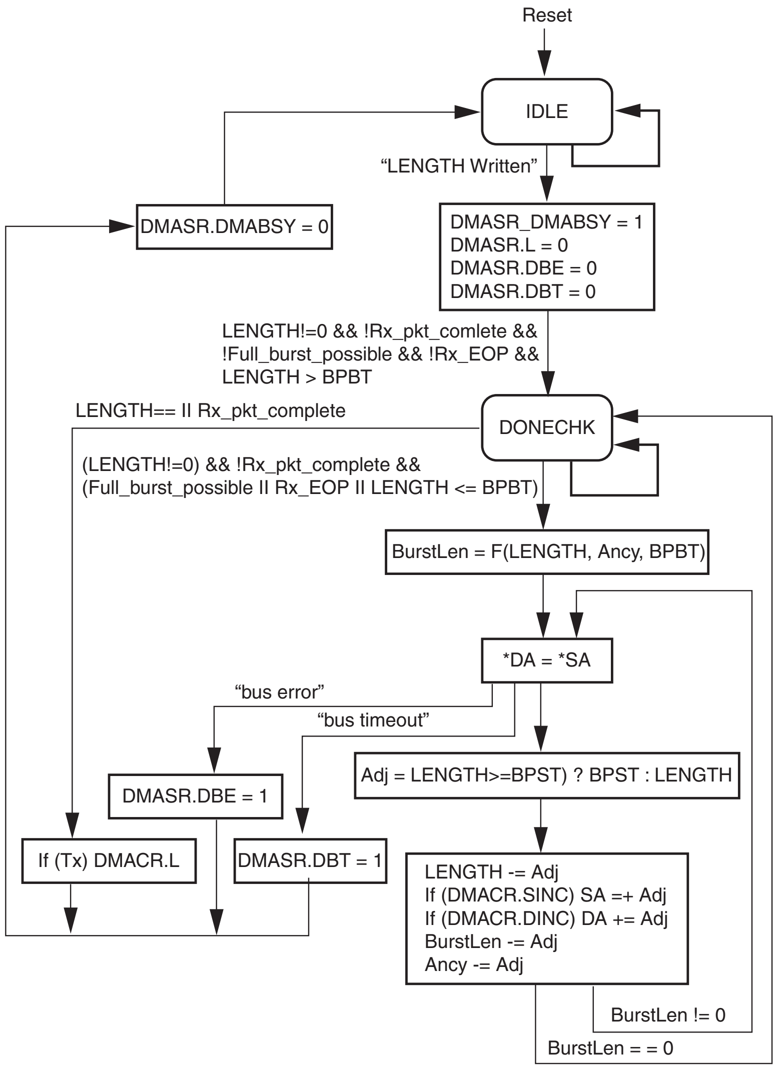

Reset
IDLE
"LENGTH Written"
DMASR.DMABSY = 0
DMASR_DMABSY = 1
DMASR.L = 0
DMASR.DBE = 0
DMASR.DBT = 0
LENGTH!=0 && !Rx_pkt_comlete &&
!Full_burst_possible && !Rx_EOP &&
LENGTH > BPBT
LENGTH== II Rx_pkt_complete
DONECHK
(LENGTH!=0) && !Rx_pkt_complete &&
(Full_burst_possible II Rx_EOP II LENGTH <= BPBT)
BurstLen = F(LENGTH, Ancy, BPBT)
*DA = *SA
"bus error"
"bus timeout"
Adj = LENGTH>=BPST) ? BPST : LENGTH
DMASR.DBE = 1
If (Tx) DMACR.L
DMASR.DBT = 1
LENGTH -= Adj
If (DMACR.SINC) SA =+ Adj
If (DMACR.DINC) DA += Adj
BurstLen -= Adj
Ancy -= Adj
BurstLen != 0
BurstLen = = 0

Figure 1: **DMA Operation**

Notes

1. C programming-language notation is used for operators such as assignment, comparison, pointer-dereference, etc.
2. SA, DA, LENGTH, DMASR: see Table 14 on page 26, Table 14 and Table 15; also, the same names are used for Buffer-Descriptor fields
3. BPST: The number of bytes per single transfer on bus
4. BPBT: The maximum number of bytes per burst transfer on bus
5. **Rx:** Channel type is 3 and SG active
6. **Tx:** Channel type is 2 and SG active
7. **Rx_EOP:** Rx end-of-packet indication has been given through the ancy.
8. Ancy: The vacancy of a Tx channel or the occupancy of an Rx channel. The controller keeps track of the ancy by adding to it byte counts reported through the status FIFO and subtracting from it bytes transferred via DMA.
9. Rx_pkt_complete: Rx && Ancy==0 && Rx_EOP
10. F: Function that determines the number of bytes to transfer on the next bus tenure
    For non Rx and non Tx, F = MIN(LENGTH, BPBT)
    For Rx or Tx, F = MIN(LENGTH, Ancy, BPBT)

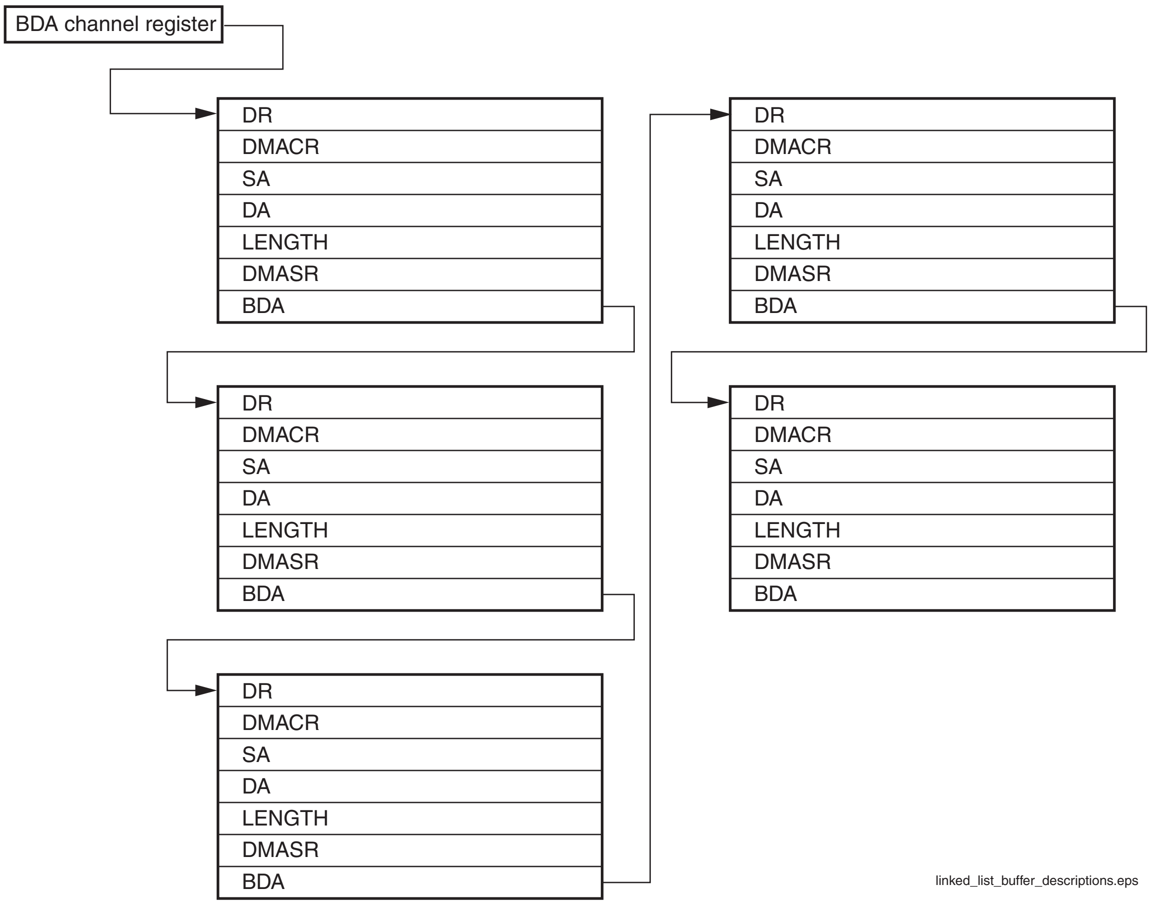
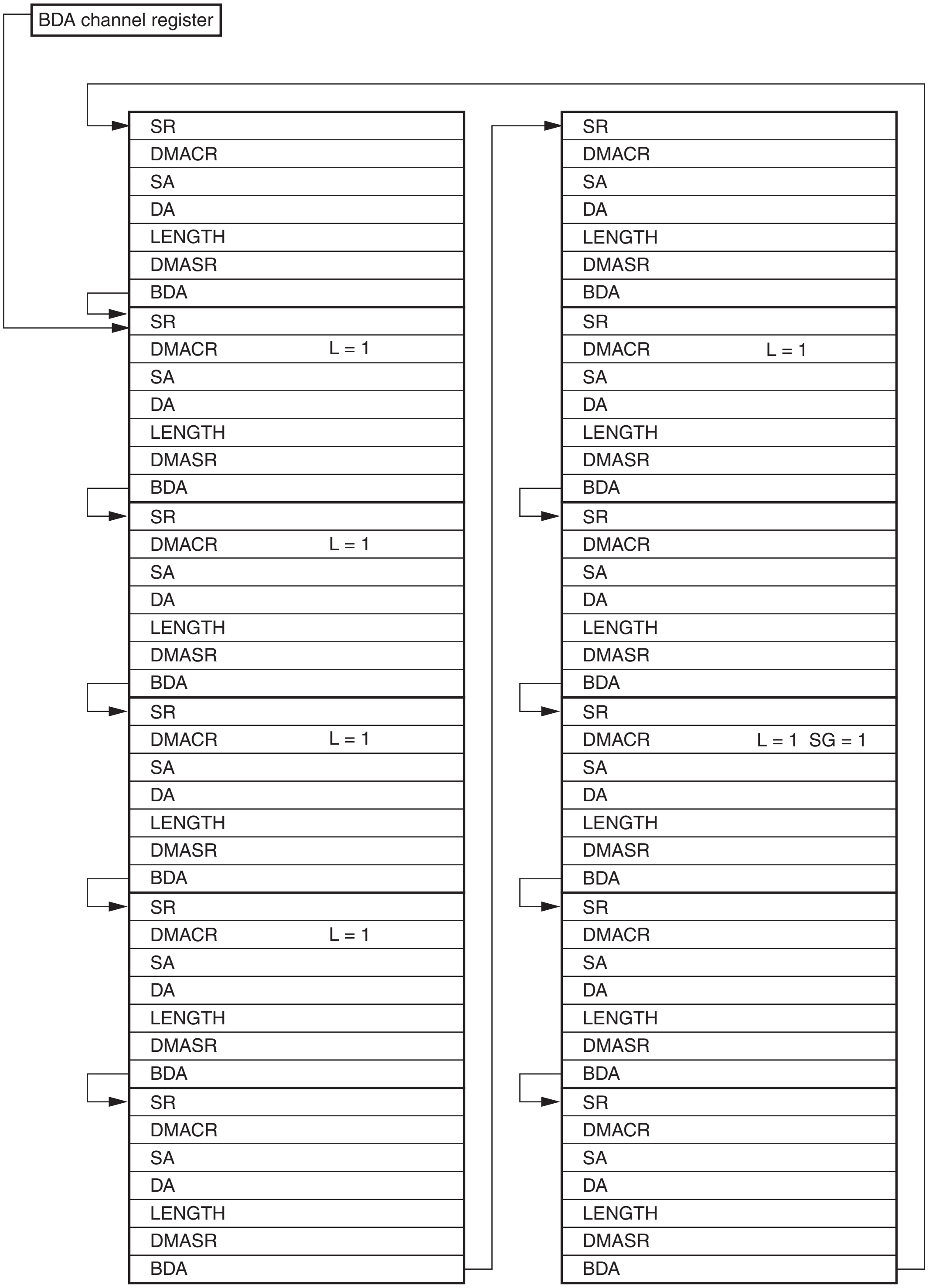


Figure 2: **Linked List of Buffer Descriptors**

Figure 3 shows Buffer Descriptors organized into a buffer ring. The buffer ring apparently is for a channel of type 2, as evidenced by L=1 tags (other Buffer Descriptors are assumed in the figure to have L=0). Some packets are specified by a single Buffer Descriptor and others by more than one consecutive Buffer Descriptor. The last ready packet is marked with SGS=1, giving a sentinel position in the ring. Also, even if Buffer Descriptors are contiguously allocated, they are required to be linked through the BDA field.



buffer_desc_org_buff_ring.eps

Figure 3: **Buffer Descriptors Organized into a Buffer Ring**

SG Operation

When Scatter Gather is enabled (SWCR.SGE=1 and DMACR.SGS=0), the SG automation will process the list of buffers referenced by the channel's Buffer Descriptor Address register (BDA). This processing will continue until it is either synchronously stopped by reaching a Buffer Descriptor with SGS=1 or asynchronously stopped by a programmed I/O write of 0 to the SGE bit of the Software Control Register.

The *current Buffer Descriptor* is the one whose address is in the BDA register. Once a current Buffer Descriptor is established, it is processed by the following steps.

1. The DMA Control Register is set to the value of the corresponding field in the Buffer Descriptor.
2. The Source Address is set to the value of the corresponding field in the Buffer Descriptor.
3. The Destination Address is set to the value of the corresponding field in the Buffer Descriptor.
4. The LENGTH Register is set to the value of the corresponding field in the Buffer Descriptor, which also starts the DMA operation.
5. When the DMA operation completes, the status in the DMA Status Register is written to the corresponding field in the Buffer Descriptor. If the DMA operation is the last for a packet on a type 3 channel, the SR field will be written with the status under which the packet was received.
6. If a stop condition is established, SG operation will stop. Otherwise, the channel Buffer Descriptor Address register will be loaded from the BDA field in the Buffer Descriptor, establishing a new current Buffer Descriptor, and the process will continue at step 1.

Stopping and Starting SG Operation

Scatter Gather processing of Buffer Descriptors continues until reaching a descriptor that has loaded a 1 into the DMACR.SGS bit (*synchronous stop*). Stopping this way is analogous to the software notion of reaching a "NIL pointer" or a "sentinel".

There may also be times when software wishes to stop SG operation without reaching the sentinel in the Buffer Descriptor list *(asynchronous stop)*. Situations where this is the case, might include

- High priority Buffer Descriptors need to be inserted ahead of already scheduled Buffer Descriptors.
- New Buffer Descriptors need to be inserted at the end of the Buffer-Descriptor list.
- Scatter Gather activity needs to be paused or aborted.

To effect an asynchronous stop, 1 is written to the SWCR.SGD bit—which, in essence, asynchronously inserts a new sentinel into the list. Scatter Gather automation will stop after completing any DMA operation (or any packets, if the channel is a packet channel) that it might have started.

The SWCR.SGE bit will be automatically cleared as a result of stopping. If the stop is synchronous, the Scatter/Gather End (SGEND) interrupt will be generated and if asynchronous, the Scatter/Gather Disable Acknowledge (SGDA) interrupt. If an asynchronous-stop point coincides with a synchronous-stop point, both interrupts will be generated.

With SG stopped, software may make any needed insertions or deletions to the Buffer-Descriptor linked list. When software wishes SG operation to restart, it establishes the condition SWCR.SGE=1, SWCR.SGD=0 and DMACR.SGS=0.

When SG restarts, software may intend to simply resume from the point of discontinuance. If so, the SG automation hardware's first step will be to establish a new Buffer Descriptor Address by dereferencing the current Buffer Descriptor's BDA field. On the other hand, software may intend to begin with a new current Buffer Descriptor. In this case, software will load a new Buffer Descriptor Address and set the DMACR.BDAEL[1] bit before starting SG.

Packet SG on Subchannels of Physical Channels

When a DMASG channel is dedicated to move packet data to or from a communications controller—the most common anticipated usage, additional considerations apply. Firstly, each channel is associated with a direction (transmit or receive), a physical channel, a subchannel within the physical channel and buffer storage for the subchannel (such as a channelized FIFO[2]). In addition, a Status FIFO is shared among all subchannels and serves as a medium for the communications controller to communicate control and status information to the DMASG automation. Figure 4 illustrates a system in which a channelized communications core is serviced by channelized DMASG.

Figure 5 gives the format of a Status-FIFO entry. The communications controller puts entries into the Status FIFO to indicate progress that it has made toward completion of a packet. The <Physical Channel, Subchannel> pair of fields give the physical subchannel on which progress has been made. The Length field gives the number of bytes that have been added to a subchannel receive buffer or removed from a subchannel transmit buffer. The EOP field indicates whether the end-of-packet condition has been established. The Status field gives the receive status if EOP is true and the subchannel is a receive subchannel).

We emphasize that a DMASG packet channel forms an association of a DMASG packet channel with a <physical channel, subchannel> pair. A physical channel represents a physical medium for transmission and reception of data. The bandwidth of the physical channel may be shared, in which case it is divided into subchannels. Subchannels are considered to always come in pairs, one for transmission and one for reception.[3]

The mapping between DMASG channels and <physical channel, subchannel> pairs is deterministic and fixed. First, all of the subchannels of the first physical channel, physical channel zero, are assigned in numerical order to DMASG channels starting at channel zero. Then subsequent physical channels, in numerical order, have their subchannels mapped to DMASG channels starting where the previous physical channel left off. An all cases, the transmission subchannel immediately precedes its paired reception subchannel.

[1] BDAEL stands for Buffer Descriptor Address Explicitly Loaded.

[2] A channelized FIFO is a buffer that embodies multiple logical FIFOs addressable one at a time. Part of the address for a channelized FIFO is the subchannel address. Data operations and FIFO-condition indications are relative to the addressed subchannel. A channelized FIFO core is available from Xilinx for use in channelized communications controllers.

[3] The two half-duplex subchannels (one for transmission and one for reception) are considered, when taken together, to form a *full-duplex* subchannel on the physical channel. However, the mapping of DMASG packet channels to <physical channel, subchannel> pairs is at the half-duplex level and "subchannel" in this document refers to such a half-duplex channel.

Thus, if physical channel zero has 2n subchannels, these are assigned as DMASG channels 0 to 2n-1. Then, the subchannels of physical channel one, if present, are assigned to DMASG channels starting at channel 2n. The mapping of subchannels to DMASG channels continues according to this scheme until all subchannels of all physical channels are assigned. Table 4 shows an example of such mappings for a communications controller having three physical channels, these having respectively two, six and four subchannels.

Table 4: **Example of Physical Subchannels Mapped to DMASG Channels**

DMASG Channel	Physical Channel	Subchannel	Direction
0	0	0	Tx
1		1	Rx
2	1	0	Tx
3		1	Rx
4		2	Tx
5		3	Rx
6		4	Tx
7		5	Rx
8	2	0	Tx
9		1	Rx
10		2	Tx
11		3	Rx

If there are DMASG channels in the system that are not packet channels (simple DMA or simple SG channels), these are assigned after the packet channels.

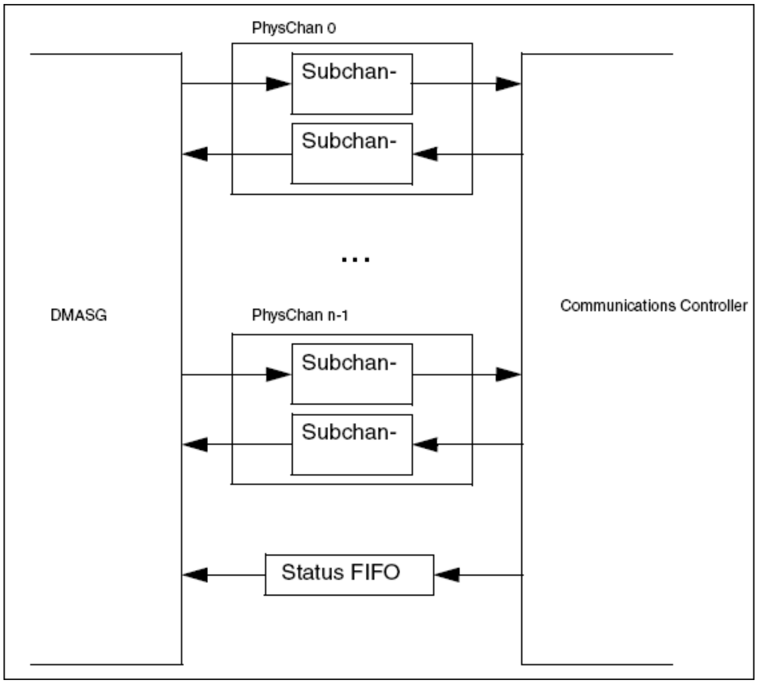


Figure 4: **Physical Channels, Subchannel FIFOs, and Status FIFO**

Physical Channel	Sub-channel	Length	EOP	Status
0 3	4 9	10 23	24	25 31

status_fifo_entry.eps

Figure 5: **Status-FIFO Entry**

Secondly, each receive packet-SG channel handles a status value. The Status Register (SR) field of the packet's last Buffer Descriptor is written with the receive packet status value when the last Buffer Descriptor of the packet is processed. The receive packet status value for the packet is made available via the Status FIFO and is placed in the same status-FIFO entry as the end-of-packet indication. There is no corresponding transmit packet status value for transmit channels.

Thirdly, packets are delimited in the SG list. A Last bit (L) is associated with the last Buffer Descriptor for a packet[1]. For packet-transmit channels, the L bit is in the DMA Control Register; for packet-receive channels it is in the DMA Status Register[2].

Fourthly, packet boundaries need to be communicated between the Channelized DMASG Controller and the communications controller. Here, transmit and receive employ different methods.

[1] Because whole packets can correspond to just one Buffer Descriptor, the first Buffer Descriptor and the last Buffer Descriptor may be the same.

[2] For packet-transmit, the L value is the DMA Control Register is copied to the DMA Status Register as part of the status for the Buffer Descriptor.

For transmit, the last byte of a packet is tagged when it is written to the transmit channel's buffer storage. The tag also indicates the amount of unused pad, if any. For receive, the receiver places an explicit end-of-packet indication in the final entry for the packet in the Status FIFO.

Finally, DMASG needs indications of data availability for receive channels or readiness to accept new data (space availability) for transmit channels. When a receive channel has made available a quantum of data, it writes a data-availability indication to the Status FIFO. This indication is tagged by the physical channel number and with end-of-packet and status indications if it is the last for a packet.

Similar indications apply to transmit channels. Whenever the transmitter for a channel has transmitted a quantum of data, it places an entry, tagged by the channel number, in the Status FIFO to indicate the amount of freed buffer storage.

After initializing the availability to zero for receive channels and to the full buffer space for transmit channels, SG automation uses the "change of availability" indications from the Status FIFO to keep track of the buffer-storage status and make decisions about when to move data. Further, because these indications are tagged by channel number, the processing for packets in progress can be demand driven.

The amount of data that should be indicated in entries in the Status FIFO can best be determined in the light of overall system properties. As the size of the subchannel buffers decreases, the quantum may need to decrease to prevent overruns and underruns. However, as the quantum decreases, the overhead involved in servicing the Status FIFO increases as does the required capacity of the Status FIFO. Further, the size of DMA bursts might also impact the choice of quanta.

The Channelized DMASG Controller does not depend on a fixed-size quantum; Status FIFO entries may report arbitrary-size quanta. However, the simple method of producing zero or more indications at a fixed quantum size, followed by a "remainer" indication at the end of a packet will certainly work and might benefit from uniformity.

Figure 6 shows the detailed, generalized operation of a SG channel from the viewpoint of the Channelized DMASG Controller. This operation also relies on the channel DMA operation as shown in the state called "DMA Operation" and as detailed in Figure 1 on page 5. The state machines of Figure 1 and Figure 6 are intended to illustrate operation but are not a normative implementation.

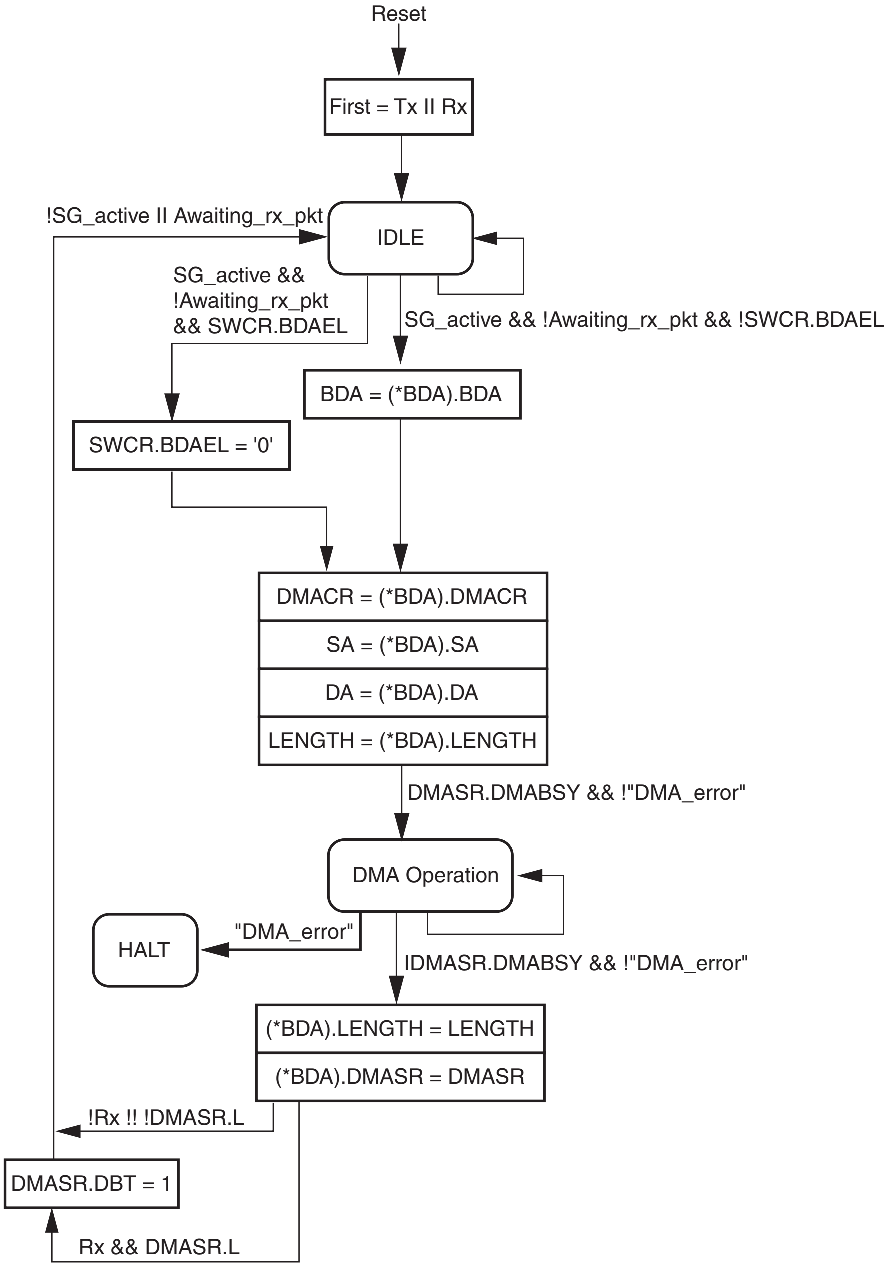


Figure 6: **Scatter-Gather Operation**

NOTES

1. C programming-language notation is used for operators such as assignment, comparison, pointer-dereference, etc.
2. **DMACR, SA, DA, LENGTH, DMASR, BDA, SWCR**: see Table 13 on page 26, Table 14 and Table 15; also, the same names are used for Buffer-Descriptor fields.
3. **Rx**: Channel is type 3.
4. **Tx**: Channel is type 2.
5. **SG_active:** A SG channel becomes active when software establishes (SWCR.SGE='1' and SWCR.SGD='0' andDMACR.SGS='0') and remains active until hardware establishes SWCR.SGE='0', which will happen in response to DMACR.SGS='1' or SWCR.SGD='1', but only after any started but not completed packet is finished.
6. **Awaiting_rx_pkt:** Rx and a packet is unavailable.

A Packet-Movement Example

In this section an example that shows packet movements and control indications from the standpoint of subchannel FIFOs and the Status FIFO is presented. The example assumes a communications controller with 12 subchannels distributed among three physical channels as outlined in Table 4 on page 10. The states of the FIFOs as time unfolds are shown in Table 6 as one of several possible orderings of the actions by the communications controller and DMASG to progress the packets.

We will assume the following workload: transmission of 99 bytes on P2S0 (physical channel 2, subchannel 0), reception of 32 bytes on P2S3, transmission of 8 bytes on P1S4 and reception of 33 bytes on P2S1. (The communications controller will use a quantum of 32 bytes, although, as stated previously, a fixed quantum and its size are optional.)

The table uses following abbreviations listed in Table 5.

Table 5: **States of the FIFO**

Abbreviation	Meaning
PC	Physical Channel
SC	Subchannel
L	Length (a change in occupancy of a receive FIFO or change in vacancy of a transmit FIFO)
E or EOP	End of Packet
S	Status
P<n>	Physical Channel <n>, where <n> can be replaced by any non-negative integer (for example, P2)
S<n>	Subchannel <n>
S<m>:<n>	Subchannel <m> has <n> bytes of data (for example, S1:32)
DMASG	The Channelized DMASG Controller, that is, the subject of this specification.
CC	The Communications Controller attached to DMASG

Table 6: **An Example of Two Transmit and Two Receive Packets Moving through Subchannel FIFOs and the Status FIFO**

Status FIFO					Subchannel FIFOs Tx			Subchannel FIFOs Rx			Comment
PC	SC	L	E	S	P0	P1	P2	P0	P1	P2	
							S0:32				DMASG channel 8 places 32 transmit data bytes into P2S0.
							S0:64				DMASG channel 8 places 32 transmit data bytes into P2S0.
							S0:32				CC removes 32 transmit bytes from P2S0.
2	0	32	0				S0:32				CC indicates 32 bytes removed from P2S0.

Table 6: **An Example of Two Transmit and Two Receive Packets Moving through Subchannel FIFOs and the Status FIFO *(Contd)***

Status FIFO					Subchannel FIFOs Tx			Subchannel FIFOs Rx			Comment
PC	SC	L	E	S	P0	P1	P2	P0	P1	P2	
2	0	32	0				S0:32			S3:32	CC places 32 receive data bytes into P2S3.
2 2	0 3	32 32	0 1	 ok			S0:32			S3:32	CC indicates 32 bytes added to P2S3, EOP and status ok.
2 2	0 3	32 32	0 1	 ok		S4:8.	S0:32			S3:32	DMASG channel 6 places 8 transmit data bytes into P1S4 and writes EOP tag.
2 2	0 3	32 32	0 1	 ok		S4:8.				S3:32	CC removes 32 transmit bytes from P2S0.
2 2 2	0 3 0	32 32 32	0 1 0	 ok		S4:8.				S3:32	CC indicates 32 removed bytes from P2S0.
2 2	3 0	32 32	1 0	ok		S4:8.				S3:32	DMASG 8 takes indication and adds 32 to its vacancy count.
2 2	3 0	32 32	1 0	ok		S4:8.	S0:32			S3:32	DMASG channel 8 places 32 transmit data bytes into P2S0.
2	0	32	0			S4:8.	S0:32			S3:32	DMASG 11 takes indication and adds 32 to its occupancy count.
2	0	32	0			S4:8.	S0:32			S3:32 S1:32	CC places 32 receive data bytes into P2S1.
2	0	32	0			S4:8.	S0:32			S1:32	DMASG 11 removes 32 data bytes from P2S3, completing a packet.
2 2	0 1	32 32	0 0			S4:8.	S0:32				CC indicates 32 added bytes to P2S1.
2 2	0 1	32 32	0 0			S4:8.	S0:32			S1:33	CC places 1 receive data byte into P2S1.
2	1	32	0			S4:8.	S0:32			S1:33	DMASG 8 takes indication and adds 32 to its vacancy count.
2 2	1 1	32 1	0 1	 ok		S4:8.	S0:32			S1:33	CC indicates 1 added bytes to P2S1, EOP and status ok.
2 2	1 1	32 1	0 1	 ok		S4:8.	S0:35 .			S1:33	DMASG channel 8 places 3 transmit data bytes into P2S0 and writes EOP and pad tags.
2 2	1 1	32 1	0 1	 ok		S4:8.	S0:3.			S1:33	CC removes 32 transmit bytes from P2S0.

Table 6: **An Example of Two Transmit and Two Receive Packets Moving through Subchannel FIFOs and the Status FIFO *(Contd)***

Status FIFO					Subchannel FIFOs Tx			Subchannel FIFOs Rx			Comment
PC	SC	L	E	S	P0	P1	P2	P0	P1	P2	
2 2 2	1 1 0	32 1 32	0 1 0	ok		S4:8.	S0:3.			S1:33	CC indicates 32 bytes removed from P2S0.
2 2	1 0	1 32	1 0	ok		S4:8.	S0:3.			S1:33	DMASG 9 takes indication and adds 32 to its occupancy count.
2 2	1 0	1 32	1 0	ok		S4:8.				S1:33	CC removes 3 transmit bytes from P2S0.
2 2	1 0	1 32	1 0	ok		S4:8.				S1:1	DMASG 9 removes 32 data bytes from P2S1.
2	0	32	0			S4:8.				S1:1	DMASG 9 takes indication and adds 1 to its occupancy count.
2	0	32	0			S4:8.					DMASG 9 removes 1 data byte from P2S1, completing a packet.
2 2	0 0	32 3	0 1			S4:8.					CC indicates 3 bytes removed from P2S0, completing a packet.
2	0	3	1			S4:8.					DMASG 8 takes indication and adds 32 to its vacancy count.
						S4:8.					DMASG 8 takes indication and adds 3 to its vacancy count.
											CC removes 8 transmit data bytes from P1S4.
1	4	8	1								CC indicates 8 bytes removed from P1S4, completing packet.
											DMASG 6 takes indication and adds 8 to its vacancy count.

Stopping Packet SG Channels

When packet SG channels are signalled to stop, either synchronously or asynchronously, they first complete any incomplete packets to reach a stop condition, then generate the SGEND and/or SGDA interrupts. A packet is complete once the LENGTH, DMASR (with L bit set) and SR (for Rx channels) fields have been written back to the last BD for the packet.[1]

[1] Tx packets are complete from the SG perspective when they have been written to the buffer, which, in general, will be before they have actually been transmitted.

To cleanly stop on packet boundaries, the DMACR.L and DMACR.SGS bits of Buffer Descriptors must be set according to these rules: For Tx channels, the SGS bit must be set on a Buffer Descriptor that also has the L bit set (a last Buffer Descriptor of a packet). For Rx channels, SGS must be set on one or more consecutive Buffer Descriptors such that the cumulative LENGTH fields of the Buffer Descriptors equals or exceeds the maximum sized packet that may be received.

Freeze Behavior

Whenever the Bus2IP_Freeze signal is asserted, the GCSR.GEN bit is cleared. (See page 24.)

Channelized DMASG Controller Parameters

To allow users to obtain a Channelized DMASG Controller that is tailored to their system, certain features are parameterizable. The features that are parameterizable are shown in Table 7.

Table 7: **IPIF Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value[2]	VHDL Type
G1	Bus address width	C_BUS_AWIDTH	32	none	natural
G2	Bus data width, the width of data moved by DMA	C_BUS_DWIDTH	32, 64	none	natural
G3	The base address of the global DMA register space, MIR, GCSR and EFIFO	C_DMA_GLOBAL_REGS_BASEADDR	A valid address on a 4-byte boundary, with unused leftmost bits set to zero	none	std_logic_vector(0 to 63)
G4	The base address of the channel-register space[1]	C_DMA_CHANNEL_BASEADDR	A valid address on a 32-byte boundary, with unused leftmost bits set to zero	none	std_logic_vector(0 to 63)
G5	The address of the Status FIFO. The status FIFO transfers per-channel status updates, including data FIFO vacancy/occupancy deltas, end-of-pack and packet status	C_DMA_STATUS_FIFO_ADDR	Value must correspond to the appropriate Status FIFO	none	Unconstrained array of std_logic_vector(0 to 63)
G6	Parameterization of the DMASG channels. See "Description of the C_IPIF_CHDMA_CHANNELS_PROPS Parameter" on page 18	C_IPIF_CHDMA_CHANNELS_PROPS	See Table 8 on page 19	none	DEPENDENT_PROPS_TYPE (Defined in ipif_pkg.vhd)

Table 7: **IPIF Design Parameters *(Contd)***

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value[2]	VHDL Type
G7	Externally assigned block ID. This value is reported in a field of the MIR	C_DEV_BLOCK_ID	0-255	none	Integer
G8	The number of bits needed to specify the burst size of a DMA transfer.	C_MA2SA_NUM_ WIDTH	Greater than zero	none	Integer
G9	The targeted FPGA family	C_FAMILY	spartan3, spartan3e, virtex4	none	String

Notes:

1. There is a register set for each channel, each register set comprising eight 32-bit registers. Channel register sets are allocated contiguously, channel zero at C_DMA_CHANNEL_BASEADDR, channel one at C_DMA_BASEADDR + 32, etc.
2. A default value of "none" indicates that an explicit value for the corresponding generic should always be specified.

Description of the C_IPIF_CHDMA_CHANNELS_PROPS Parameter

The Channelized DMASG Controller is designed to be a sub-block of the IP Interface (IPIF). A parameterization scheme for including sub-blocks and their block-dependent parameterization already exists at the IPIF level. This is conceptually known as *dependent property parameterization*, where the dependent properties—the parameters that are dependent upon the type of block—are passed in an array parameter called C_ARD_DEPENDENT_PROPS_ARRAY that is associated with an *address space*. The C_ARD_DEPENDENT_PROPS_ARRAY entry that is passed to the Channelized DMASG Controller as C_IPIF_CHDMA_CHANNELS is the entry associated with the IPIF_CHDMA_CHANNELS address space.

The definition and embodiment of the dependent properties parameterization concepts is found in VHDL package, ipif_pkg, which is in the ipif_pkg.vhd file for the ipif_common_v1_00_d library. An extract of the parameters relevant for the Channelized DMASG Controller is given next in Table 8. Each parameter has a value that is an integer. Boolean values are encoded by using only the values one and zero.

Table 8: **Parameters Passed in the C_IPIF_CHDMA_CHANNELS_PROPS**

Parameter	Description
NUM_SUBS_FOR_PHYS_0	The number of subchannels to include for the corresponding physical channel. Setting the value to zero for a given physical channel has the effect of not including the given physical channel. The number of subchannels must be even, because these channels come in half-duplex pairs, one for Tx at an even address followed by one for Rx at the succeeding odd address.
NUM_SUBS_FOR_PHYS_1	
NUM_SUBS_FOR_PHYS_2	
NUM_SUBS_FOR_PHYS_3	
NUM_SUBS_FOR_PHYS_4	
NUM_SUBS_FOR_PHYS_5	
NUM_SUBS_FOR_PHYS_6	
NUM_SUBS_FOR_PHYS_7	
NUM_SUBS_FOR_PHYS_8	
NUM_SUBS_FOR_PHYS_9	
NUM_SUBS_FOR_PHYS_10	
NUM_SUBS_FOR_PHYS_11	
NUM_SUBS_FOR_PHYS_12	
NUM_SUBS_FOR_PHYS_13	
NUM_SUBS_FOR_PHYS_14	
NUM_SUBS_FOR_PHYS_15	
NUM_SIMPLE_DMA_CHANS	The number of simple DMA channels
NUM_SIMPLE_SG_CHANS	The number of simple SG channels
INTR_COALESCE(1)	Whether or not to include interrupt coalescing support. 0 - Interrupt coalescing is disabled 1 - Interrupt coalescing is enabled
CLK_PERIOD_PS(2)	The period of the OPB Bus clock in ps. The default value of 0 is a special value that is synonymous with 10000 ps (10 ns). Knowledge of the period is only needed to divide the clock to form PACKET_WAIT_UNIT_NS.
PACKET_WAIT_UNIT_NS(2)	Gives the unit in which pack-wait bounds are measured. The default value of 0 is a special value that is synonymous with 1,000,000 ns (1 ms). A non-default value is typically used only for testing.

Table 8: **Parameters Passed in the C_IPIF_CHDMA_CHANNELS_PROPS**

Parameter	Description
BURST_SIZE	The maximum number of bus transfers that will comprise a burst. Allowed values are 1, 2, 4, 8 or 16. The default value of 0 is a special value that is synonymous with a burst size of 16. Setting the BURST_SIZE to 1 effectively disables bursts by making all bus transactions single-beat.
REMAINDER_AS_SINGLES	The type of bus transaction to use once the amount of data to move is less than BURST_SIZE. There are two options: 0 - Remainder handled as a short burst 1 - Remainder handled as a series of single-beats

Notes:

1. This parameter is relevant only if there is at least one physical channel with subchannels.
2. This parameter is relevant only if INTR_COALSCE = 1.

Allowable Parameter Combinations

See the notes of Table 8 for the few parameter interdependencies that exist.

Channelized DMASG Controller I/O Signals

The I/O signals for the Channelized DMASG Controller are listed in Table 9.

Table 9: **Channelized DMASG Controller I/O Signals**

Port	Signal Name	I/O	Description
P1	Bus2IP_Data(0:C_BUS_DWIDTH-1)	O	Data to IP
P2	DMA2Bus_Data(0:C_BUS_DWIDTH-1)	I	Data from IP
P3	Bus2IP_RNW	I	Read / not Write qualifier
P4	Bus2IP_Addr(0:C_BUS_AWIDTH-1)	O	Address to IP
P5	Bus2IP_RdReq	O	Read request
P6	Bus2IP_WrReq	O	Write request
P7	Bus2IP_BE(0 to C_IPIF_DBUS_WIDTH/8 - 1)	O	Byte enables
P8	Bus2IP_Burst	I	Burst-mode qualifier
P9	DMA2Bus_RdAck	I	Read acknowledgement
P10	DMA2Bus_WrAck	I	Write acknowledgement
P11	DMA2Bus_Retry	I	Retry response
P12	DMA2Bus_Error	I	Error response
P13	DMA2Bus_ToutSup	I	Asserted by the IP whenever its acknowledgement or retry response will take longer than 8 cycles
P14	Channels_CS	I	Select indicating that some channel register is selected
P15	Global_Regs_CS	I	Select indicating that some global register is selected
P16	DMA2Bus_Addr(0:C_BUS_AWIDTH-1)	O	The OPB address for the master transaction

Table 9: **Channelized DMASG Controller I/O Signals *(Contd)***

Port	Signal Name	I/O	Description
P17	DMA2IP_Addr(0:C_BUS_AWIDTH-1)	O	The local device address for the master transaction; this address will be the source for a master write transaction and the sink for a master read transaction.
P18	DMA2IP_Tag(0:C_BUS_DWIDTH/8-1)	O	Tag bits for Tx data moved by DMA. Only the rightmost 1+log$_2$(C_BUS_DWIDTH) bits are used; the rest are driven to zero. Of the used rightmost bits, the leftmost bit is the end-of-packet indicator. The rest of the bits give the number of bytes of the bus that are occupied by data, whereby the value zero is a special case that indicates that all bytes are occupied. When the tag indicates that one or more bytes are not occupied, this "pad" is at the right, that is, at the higher numbered bytes.
P19	DMA2Bus_MstWrReq	O	Master write request
P20	DMA2Bus_MstRdReq	O	Master read request
P21	DMA2Bus_MstBE(0:C_BUS_DWIDTH/8 - 1)	O	Byte-enables qualifiers
P22	DMA2Bus_MstNum(0 to C_MA2SA_NUM_WIDTH-1)	O	The number of bus data beats that are required to fulfill the currently active master read or write request.
P23	DMA2Bus_MstBurst	O	Burst qualifier
P24	DMA2Bus_MstBusLock	O	Bus-lock qualifier
P25	DMA2Bus_MstLoc2Loc	O	Always driven to ’0’ by this DMA controller
P26	DMA2Bus_MstSimple	O	The current master request is qualified as a simple operation. Only used for DMA master reads. DMA2IP_Addr is not used and the read data is taken from Bus2IP_MstData on the cycle when Bus2IP_MstWrStrb is asserted.
P27	Bus2IP_MstData(0 to 31)	I	Data for a simple master read.
P28	Bus2IP_MstWrStrb	I	One-cycle strobe indicating that simple-master read data is available.
P29	Bus2IP_MstRdAck	I	A one-cycle acknowledgement of a master read transfer; assertion is for one cycle.
P30	Bus2IP_MstWrAck	I	A one-cycle acknowledgement of a master write transfer
P31	Bus2IP_MstLastAck	I	A one-cycle acknowledgement of a master transaction; a transaction may consist of multiple transfers (burst transaction); LastAck will always accompany the last RdAck or WrAck for the transaction
P32	Bus2IP_MstError	I	Valid during the cycle that Bus2IP_MstLastAck is active; indicates whether the transfer has an error; note: a burst transaction reporting an error may have terminated prematurely.

Table 9: **Channelized DMASG Controller I/O Signals** ***(Contd)***

Port	Signal Name	I/O	Description
P33	Bus2IP_DMAMstTrans	I	Indication that the IPIF is currently performing a DMA master transaction.
	Bus2IP_MstRetry	I	A one-cycle alternative completion signal to Bus2IPMstLastAck; indicates that the requested transaction could not be performed but may succeed if retried; if IP2Bus_MstRdReq (or WrReq) remains asserted in the following cycle, the IPIF will retry the transaction and may reuse any state that it has built up in support of the transaction (the IP must leave addresses and transaction qualification signals unchanged); if otherwise the request signal is deasserted on the following cycle, the transaction is considered abandoned from the point of view of the IPIF.
	Bus2IP_MstTimeOut	I	A one-cycle alternative completion signal to Bus2IPMstLastAck; indicates that the requested transaction could not be performed within the time-out interval associated with the OPB.
	IP2DMA_StatusFIFO_Empty	I	Indication that the Status FIFO is empty.
	DMA2Intr_Intr(0:C_DMA_CHAN_TYPE'length-1)	O	Interrupt conditions, one for each of the channel Interrupt Source Controllers within the Channelized DMASG Controller.
	Bus2IP_Clk	I	Clock to which the Channelized DMASG Controller is synchronous
	Bus2IP_Reset	I	Hardware reset
	Bus2IP_Freeze	I	Tells the Channelized DMASG Controller to gracefully freeze
	Mstr_sel_ma	I	If equal to '1', DMASG is the active master on any on-going master operation.
P34	Bus2IP_DMA_Req	O	(Place-holder signal for possible future expansion of functionality.)
P35	IP2Bus_DMA_Ack	I	(Place-holder signal for possible future expansion of functionality.)

Channelized DMASG Controller Register Descriptions

Registers are 32-bits wide and come in two categories, *global* and *per-channel*. Each category is represented by a range of contiguous addresses. The base addresses for the address ranges are established by the environment that instantiates the Channelized DMASG Controller. Therefore, addresses given here are expressed as offsets from the corresponding base address for either the Global Registers or the Channel Registers.

Global Registers

There are three global registers. These are summarized in Table 10. A summary of register fields is given in Table 11 and details in Table 12.

Table 10: **Global Registers, Summary**

Register Name	Abbreviation	Address Offset from C_DMA_GLOBAL_REGS_ BASEADDR	Access
Module Identification Register	MIR	0x0	Read
Global CSR	GCSR	0x4	Read/Write
Event FIFO	EFIFO	0x8	Read

Table 11: **Global Registers, Summary of Fields**

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	MIR	Major Revision				Minor Revision							Revision Letter					Block ID								Type							
1	GCSR	GEN	reserved																						EEFO	reserved							EFO
2	EFIFO	SGEND	SGDA	PWBR	PCTR	PD	DE	DD	reserved	CHAN (Channel Number)												EP (Event Parameter)											

Table 12: **Global Registers, Details of Fields**

Register Name	Field Name	Bits	Description	Reset	Properties
MIR	Module Identification Register				
		0-3	Major Revision. Most significant component of the revision identifier for the implementation of theChannelized DMASG Controller. This would be the "2" in a version number like v2_01_d.		r
		4-10	Minor Revision. Second most significant component of the revision identifier for the implementation of theChannelized DMASG Controller. This would be the "01" in a version number like v2_01_d		r
		11-15	Revision Letter. Software-compatibility component of the revision identifier for the implementation of theChannelized DMASG Controller. This would be the "d" in a version number like v2_01_d.		r
		16-23	Block ID. This value varies from build to build and is a unique block identification for the current build.		r
		24-31	Block Type. This is a constant value, fixed to 0x4D for the Channelized DMASG Controller.		r
GCSR	Global Control Status Register				
	GEN	0	Global Enable. When false, all channels are held disabled. When true, the channels are allowed to make progress as programmed. GEN=0 also resets the Event FIFO. A Transition to GEN=0 takes affect immediately, interrupting any ongoing activity on any channel. A "graceful" stop can be achieved by allowing all DMA/SG operations to complete before clearing GEN. Channel registers do not have reset values. Registers must be programmed to desired initial states while the Global Enable bit is false. This initialization requirement extends also to bits reserved for system use. (See registers SYS, DMCR and DMASR.)	0	rw
		2-22	Reserved. Not written; read as zero.	0	
	EEFO	23	Enable Event FIFO Overflow.	0	rw
	EFO	31	Event FIFO Overflow. Interrupt status bit for the event-FIFO overflow condition. If this bit is set and is not masked by EEFO=0, it causes an interrupt. This bit is cleared whenever GEN=0.	0	r
		24-30	Reserved. Not written; read as zero.	0	

Table 12: **Global Registers, Details of Fields**

Register Name	Field Name	Bits	Description	Reset	Properties
EFIFO	Event FIFO				
	SGEND[1]	0	SG END event. SG operation has stopped as a result of finishing packet when DMACR.SGS=1 on the last BD.		r
	SGDA[1]	1	SG Disable Acknowledge event. SG operation has stopped as a result of finishing a packet when SWCR.SGE=0.		r
	PWBR[1]	2	Packet Wait Bound Reached event. The number of packets in the Event Parameter (EP) field is being reported due to a packet waiting to be reported for a time interval equal to the Packet Wait Bound.		r
	PCTR[1]	3	Packet Count Threshold Reached event. The number of unreported packets has reached the Packet Count Threshold. (The Event Parameter field is set to this value.)		r
	PD[1]	4	Packet Done event. The handling of a packet has been completed.		r
	DE[1]	5	DMA Error event. A DMA operation has finished with an error. If a DMA error occurs under SG operation, the effect is fatal. The channel and any associated FIFOs or devices must be re-initialized.		r
	DD[1]	6	DMA Done event. A DMA operation has finished either normally or with an error.		r
		7	Reserved		r
	CHAN	8-19	Channel number. Identifies the channel number on which the event occurred.		r
	EP	20-31	Event Parameter. The Event Parameter is the number of packets, possibly zero, whose completion is being reported in conjunction with this event.		r

Note:
1. Reporting is enabled via a corresponding enable bit in the Software Control Register (SWCR).

Channel Registers

The Channelized DMASG Controller contains eight register addresses for each channel. The eight registers are summarized in Table 13. A summary of register fields is given in Table 14 and details in Table 15.

Address offsets are a function of the channel number. Channel numbers are consecutive and start at zero.

Table 13: **Channel Registers, Summary**

Register Name	Abbreviation	Address Offset From C_DMA_CHANNEL_ BASEADDR	Access
System	SYS	dmasg_chan*0x20 + 0x0	none[1]
DMA Control Register	DMACR	dmasg_chan*0x20 + 0x4	Read/Write
Source Address	SA	dmasg_chan*0x20 + 0x8	Read/Write
Destination Address	DA	dmasg_chan*0x20 + 0xC	Read/Write
DMA Length.	LENGTH	dmasg_chan*0x20 + 0x10	Read/Write
DMA Status Register	DMASR	dmasg_chan*0x20 + 0x14	Read
Buffer Descriptor Address	BDA	dmasg_chan*0x20 + 0x18	Read/Write
Software Control Register	SWCR	dmasg_chan*0x20 + 0x1C	Read/Write

Notes:

1. Here, access is defined as access during steady-state operation. Some registers that are not writable during operation nevertheless must be initialized by software before channel operation is enabled.

Table 14: **Channel Registers, Summary of Fields**

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	SYS[1]	System										System (Ancy)														System							
1	DMACR[1]	SINC	DINC	SLOCAL	DLOCAL	SGS		L	System																								
2	SA																																
3	DA																																
4	LENGTH																																
5	DMASR[1]	DMABSY	DBE	DBT	L	SGBSY	System							System (Packet Wait Timer)										System (Unreported Packet Count)									
6	BDA																																
7	SWCR	ESGEND	ESGDA	EPWBR	EPCTR	EPD	EDE	EDD	BDAEL	SGE	SGD			PWB (Packet Wait Bound)										PCT (Packet Count Threshold)									

Notes:

1. This register has bits reserved for system use that must be initialized before the controller transitions from the globally disabled to the globally enabled state. (See "Global Control Status Register" on page 24.)

Table 15: **Channel Registers, Details of Fields**

Register Name	Field Name	Bits	Description	Initialization Required	Properties
SYS			System Register		
		0-9	For system use; must be initialized to zero during GEN=0.	yes	w
	ANCY	10-23	Ancy, occup*ancy* for receive channels and vac*ancy* for transmit channels. For system use; must be initialized during GEN=0, to zero for receive channels and to the available buffer storage, in bytes and a multiple of the bus width, for transmit channels.	yes	w
		24-31	For system use; must be initialized to zero during GEN=0.	yes	w
DMACR			DMA Control Register		
	SINC	0	Source Increment. During DMA operation, increment SA by one for each byte transferred.		rw
	DINC	1	Destination Increment. During DMA operation, increment DA by one for each byte transferred.		rw
	SLOCAL	2	Source address is Local. Indicates to the DMA engine that all source addresses for the DMA operation will be local to the device[1].		rw
	DLOCAL	3	Destination address is Local. Indicates to the DMA engine that all destination addresses for the DMA operation will be local to the device[1].		rw
	SGS	4	Scatter Gather Stop. SGS is used to tell the SG automation to stop as soon as a stop condition holds. For simple SG channels, the stop condition is established by completion of the buffer descriptor for which the SGS bit is set. For packet SG channels, the start condition is established by completion of all partially completed packets. (See also "Stopping and Starting SG Operation" on page 8, "Stopping Packet SG Channels" on page 16, and the SWCR.SGE and EFIFO.SGEND descriptions.)		rw
		5	Reserved. Not used for system purposes.		
	L	6	Last. The buffer associated with this descriptor represents the last of a packet. This bit is relevant when a packet is being sourced from the Buffer Descriptors.		rw
		7-31	For system use; must be initialized to zero during GEN=0.	yes	w

Notes:

1. Implementations may restrict the combinations of SLOCAL and DLOCAL that are permitted. A common restricted configuration is one that requires exactly one of the source or destination.

Table 15: **Channel Registers, Details of Fields *(Contd)***

Register Name	Field Name	Bits	Description	Initialization Required	Properties
SA	Source Address				
		0-31	The source address for DMA.		rw
DA	Destination Address				
		0-31	The destination address for DMA.		rw
LENGTH	DMA Length				
		0-31	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 a non-zero value to this register allows the DMA operation to start, so the register should be written last when setting up a DMA operation. For packet-receive channels, the loaded value must be a multiple of the bus width.		rw
DMASR	DMA Status Register				
	DMABSY	0	DMA Busy. This bit is set when hardware reaches a point that it recognizes and acts on a newly established LENGTH-not-equal-to-zero condition. There is, in general, some delay from the establishment of the condition to the assertion of DMABSY as DMA processing starts for the channel. Therefore, polling this bit as a test for DMA completion is not recommended. Instead, test for LENGTH=0.		r
	DBE	1	DMA Bus Error		r
	DBT	2	DMA Bus Timeout		r
	L	3	Last. This Buffer Descriptor is the last of a packet. Must be initialized to one ('1') during GEN=0.	yes	r
	SGBSY	4	SG Busy. This bit is set when hardware reaches a point that it recognizes and acts on a newly established SGE=1 and SGD=0 and SGS=0 condition. There is, in general, some delay from the establishment of the condition to the assertion of SGBSY as SG processing starts for the channel. Therefore, polling this bit as a test for SG completion is not recommended. Instead, test for SGE=0, because SGE is cleared by hardware as SG completes.		r
		5-31	For system use; must be initialized to zero during GEN=0.	yes	

Table 15: **Channel Registers, Details of Fields *(Contd)***

Register Name	Field Name	Bits	Description	Initialization Required	Properties
BDA	Buffer Descriptor Address				
		0-31	Address of the Buffer Descriptor that corresponds to the current DMA operation.		rw
SWCR	Software Control Register				
	ESGEND	0	Enable SG End event.		rw
	ESGDA	1	Enable SG Disable Acknowledge event.		rw
	EPWBR	2	Enable Packet Wait Bound Reached event.		rw
	EPCTR	3	Enable Packet Count Threshold Reached event.		rw
	EPD	4	Enable Packet Done event.		rw
	EDE	5	Enable DMA Error event.		rw
	EDD	6	Enable DMA Done event.		rw
	BDAEL	7	Buffer Descriptor Address Explicitly Loaded. If the application writes a new BDA while SG is stopped, then it must also set this bit. Otherwise, when SG is restarted, BDA will be loaded with the next BDA taken from the current Buffer Descriptor.		rw
	SGE	8	Scatter-Gather Enable and Scatter-Gather Disable. SG automation does not begin unless SGE=1 and DMACR.SGS = 0. SG automation can be stopped while active by setting SGD=1. Whenever SG automation stops—irrespective of whether caused by SGS=1 or SGD=1—SGE is cleared by the hardware. (See also "Stopping and Starting SG Operation" on page 8, "Stopping Packet SG Channels" on page 16, and the DMACR.SGS and EFIFO.SGDA descriptions.)		rw
	SGD	9			
		10-11	Reserved. Not used for system purposes.		
	PWB	12-21	Packet Wait Bound. The maximum amount of time that an unreported packet is required to wait until unreported packets are reported via a PWBR event. (Must remain unchanged for the duration of a SG operation.)		rw
	PCT	22-31	Packet Count Threshold. The number of unreported packets that must accumulate before they are reported via a PCTR event. (Must remain unchanged for the duration of a SG operation.)		rw

Channelized DMASG Controller Interrupt Descriptions

DMA-Channel Event Reporting

Each channel may generate one or more events that are of interest to the user. Events are reported via the *Event FIFO*, shown in Figure 7. The Event FIFO is shared in common by all channels. The FIFO generates a system interrupt when it is non-empty.

Event information is read by software through the FIFO interface to the bus through the EFIFO global register. The format of an Event FIFO entry is given in Table 12 on page 24. This entry allows events to be reported in terms of the channel, the event(s) that occurred, and, if needed, additional information in an even parameter.

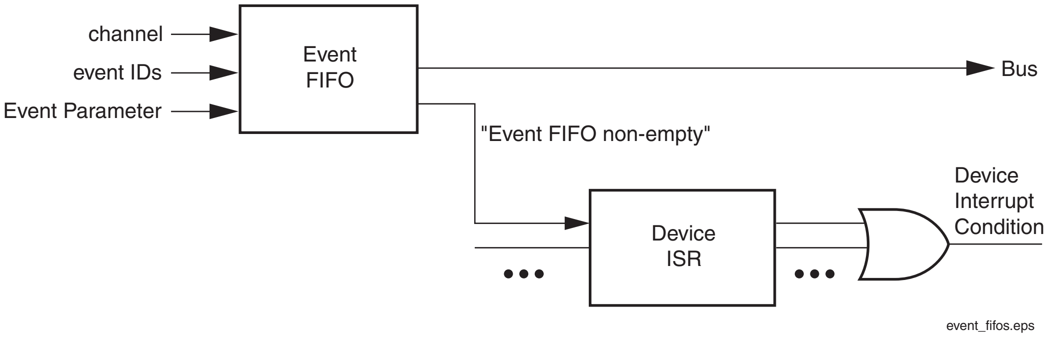


Figure 7: **Event FIFO**

Interrupt Coalescing

Under SG operation, many packet transfers and the underlying DMA operations of which they are composed, proceed automatically. Nevertheless, software needs to receive reports of packets that have been handled by DMASG so that it can perform its part of the process, for example, posting of completion notifications to client processes, management of buffer resources, etc. For low packet-rate devices, an interrupt per packet might be acceptable. For high packet-rate devices, however, this may result in an undesirably high interrupt rate. Reducing the interrupt rate to below one per packet is called *interrupt coalescing*.

The user specifies the number of packets to be reported in aggregate by setting the value of the Packet Count Threshold (PCT). SG keeps a count of the number packets it has handled but not yet reported. Whenever this unreported packet count grows equal to the Packet Count Threshold value, a Packet Count Threshold Reached (PCTR) event is added as an entry in the Event FIFO. In this entry, the Event Parameter (EP) field is set to the unreported packet count, which is then cleared.

Please be aware, however, that unreported packets may be reported in the EP field with any event, not just a PCTR event. Successful management of packet-completion reporting by software requires that the EP value always be added to the number of packets requiring software service for the channel corresponding to the event. Each packet completion will be reported exactly once.

Coalescing packets and reporting them in aggregate has an undesirable side effect, however. If the number of unreported packets does not grow large enough to reach the Packet Count Threshold—as could happen, for example, if the channel enters a period of inactivity—packets might wait indefinitely before being reported.

To solve this case, activity of a channel is monitored by a Packet Wait Timer. Whenever the unreported packet count is zero, the timer is inactive. Whenever the unreported packet count goes from zero to one, the timer starts toward a time-out at the Packet Wait Bound (PWB) value. If the time-out is reached, a Packet Wait Bound Reached (PWBR) event is added to the Event FIFO with the Event Parameter (EP) field set to the unreported packet count, which is then cleared.

Design Implementation

Design Tools

The Channelized DMASG Controller design is implemented in VHDL, and may use both RTL-style code and instantiation of Xilinx FPGA primitives. The design is processed by the Alliance tool suite.

Design Verification

The Channelized DMASG Controller design is verified using a dedicated test bench to exercise its functions. The test bench accepts the same parameterization as the Channelized DMA[SG] core, allowing to test different numbers of channels, different combinations of channel types, and other parameterizable features. The test bench presents a non-recurring workload that can run for an indefinite period of time

Specification Exceptions

Implementations compliant with this version of this document may have these restrictions

- For each DMA operation, either the source or destination is local and the other remote.
- Initial source and destination addresses for each DMA transfer are required to be aligned to a byte address divisible by four. The length of the transfer need not be a number of bytes divisible by the number of bytes on the bus and all length and packet-length calculations will be byte-correct but the actual number of bytes moved will be the smallest multiple of the number of bytes on the bus greater than or equal to the specified length, resulting in padding for non-aligned lengths[1].

Reference Documents

The following documents contain reference information:

1. Xilinx LogiCORE™ *DS-416 Direct Memory Access and Scatter Gather* Product Specification.

[1] This implies that packets defined over multiple Buffer Descriptors must restrict the LENGTH field of intermediate Buffer Descriptors to be a multiple of four bytes if intermediate padding within the packet is to be avoided.

Revision History

Date	Version	Revision
06/14/04	1.0	Initial released version.
8/11/04	1.1	Updated for release 6.3i; updated trademarks and supported device family listing, made minor content edits.
01/31/05	1.2	Converted to new DS template; updated figures to Xilinx graphic standards, reformatted tables.
4/22/08	1.3	Added Automotive Spartan-3E, Automotive Spartan-3A, Automotive Spartan-3, and Automotive Spartan-3A DSP support.
4/24/09	1.4	Updated tool support and legal information. Eliminated deprecated architectures.
10/23/09	1.5	Incorporated CR531889; removed *VHDL test bench* from Verification tool field; removed Spartan-3A/DSP from supported device families list.
2/25/10	1.5.1	Documentation change only. Changed document date to February 25, 2010. This new version will replace the 4/24/09 version posted on the web.

Notice of Disclaimer

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