LogiCORE IP AXI DMA v6.01.a

Product Guide

PG021 July 25, 2012

XILINX®

XILINX®

Table of Contents

SECTION I: SUMMARY

Introduction

The Advanced eXtensible Interface (AXI) Direct Memory Access (AXI DMA) core is a soft Xilinx Intellectual Property (IP) core for use with Xilinx® Embedded Development Kit (EDK), the CORE Generator™ tools, and Vivado™ Design Suite. The AXI DMA engine provides high-bandwidth direct memory access between memory and AXI4-Stream-type target peripherals. Its optional scatter gather capabilities also off-load data movement tasks from the Central Processing Unit (CPU). Initialization, status, and management registers are accessed through an AXI4-Lite slave interface, suitable for the Xilinx MicroBlaze™ microprocessor.

Features

- AXI4 compliant
- Optional Independent Scatter/Gather Direct Memory Access (DMA) support
- Optional Simple DMA Support
- Primary AXI4 Memory Map data width support of 32, 64, 128, 256, 512 and 1024 bits
- Primary AXI4-Stream data width support of 8, 16, 32, 64, 128, 256, 512 and 1024 bits
- Supports Multi Channelling up to 16 channels in Scatter/Gather mode
- Supports optional 2D transfers
- Optional Keyhole (setting up of source and destination addresses as fixed) support
- Optional Data Re-Alignment support
- Optional AXI Control and Status Streams
- Use with Xilinx Platform Studio (XPS), Xilinx Software Development Kit (SDK), Xilinx CORE Generator tool, Vivado Design Suite

LogiCORE™ IP Facts Table	
Core Specifics	
Supported Device Family(1)(2)	Zynq™-7000(3), Artix™-7, Virtex®-7, Kintex™-7, Virtex-6, Spartan®-6
Supported User Interfaces	AXI4, AXI4-Lite, AXI4-Stream
Resources	See Table 2-4, Table 2-5, and Table 2-6
Provided with Core	
Design Files	ISE®: VHDL Vivado™: VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	Not Provided
Supported S/W Drivers(4)	Standalone and Linux
Tested Design Flows(5)	
Design Entry	SDK 14.2 XPS 14.2 Integrated Software Environment (ISE) 14.2 Vivado Design Suite 2012.2(6)
Simulation	Mentor Graphics ModelSim,
Synthesis	Xilinx Synthesis Technology (XST) Vivado Synthesis
Support	
Provided by Xilinx, Inc@ www.xilinx.com/support.	

1. For a complete list of supported EDK derivative devices, see Embedded Edition Derivative Device Support.
2. For a complete listing of supported devices for IP cores, see the release notes for this core.
3. Supported in ISE Design Suite implementations only.
4. Standalone driver information can be found in the EDK or SDK installation directory. See xilinx_drivers.htm in <install_directory>/doc/usenglish. Linux OS and driver support information is available from wiki.xilinx.com.
5. For the supported versions of the tools, see the Xilinx Design Suite: Release Notes Guide.
6. Supports only 7 series devices.

Overview

The AXI Direct Memory Access (AXI DMA) IP provides high-bandwidth direct memory access between the AXI4 memory mapped and AXI4-Stream IP interfaces. Its optional scatter gather capabilities also off load data movement tasks from the Central Processing Unit (CPU) in processor-based systems. Initialization, status, and management registers are accessed through an AXI4-Lite slave interface. Figure 1-1 illustrates the functional composition of the core. The core's design has four AXI4 Memory Map interfaces:

- AXI4-Lite Slave
- AXI4 Memory Map Read Master
- AXI4 Memory Map Write Master
- Optional AXI4 Memory Map Scatter/Gather Read/Write Master

Associated with the memory map interfaces are four AXI4-Stream interfaces:

- AXI4 Memory Map to Stream (MM2S) Stream Master
- AXI4-Stream to Memory Map (S2MM), Stream Slave
- AXI Control Stream Master
- AXI Status Stream Slave

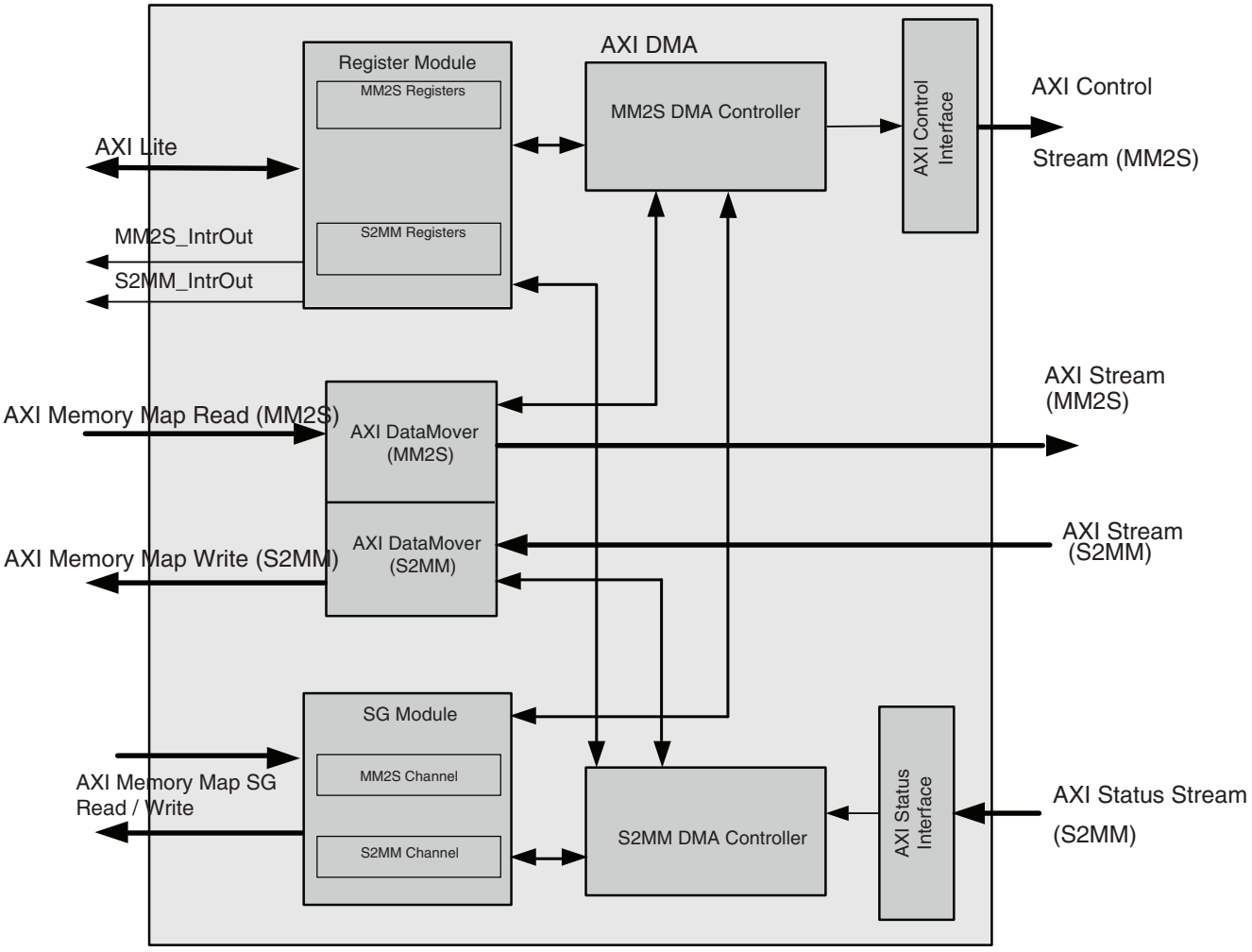

Figure 1-1: **AXI DMA Block Diagram**

Primary high-speed DMA data movement between system memory and stream target is through the AXI4 Memory Map Read Master to AXI MM2S Stream Master, and AXI S2MM Stream Slave to AXI4 Memory Map Write Master. AXI DMA also enables up to 16 multiple channels of data movement on both MM2S and S2MM paths in Scatter/Gather mode. The AXI DataMover is used for high throughput transfer of data from memory to stream and from stream to memory. The MM2S channel and S2MM channel operate independently and in a full-duplex-like method. The AXI DataMover provides the AXI DMA with 4 KB address boundary protection, automatic burst partitioning, as well as providing the ability to queue multiple transfer requests using nearly the full bandwidth capabilities of the AXI4-Stream buses. Furthermore, the AXI DataMover provides byte-level data realignment allowing memory reads and writes to any byte offset location.

Associated with each primary data channel is a stream channel for off-loading packet metadata from the primary datapath. The MM2S channel supports an AXI Control stream for sending user application data to the target IP. For the S2MM channel, an AXI Status stream is provided for receiving user application data from the target IP.

The optional Scatter/Gather Engine fetches and updates buffer descriptors from system memory through the AXI4 Memory Map Scatter Gather Read/Write Master interface. Optional descriptor queuing is provided to maximize primary data throughput.

Typical System Interconnect

The AXI DMA core is designed to be connected through the AXI Interconnect in the user's system. A typical MicroBlaze™ processor configuration is shown in Figure 1-2. The system's microprocessor has access to the AXI DMA through the AXI4-Lite interface. An integral Scatter/Gather Engine fetches buffer descriptors from DDRx which then coordinates primary data transfers between AXI IP and DDRx. Optional Control and status streams provide packet-associated information, such as checksum off-load control/status, to and from an Ethernet based IP. The dual interrupt output of the AXI DMA core is routed to the System Interrupt Controller.

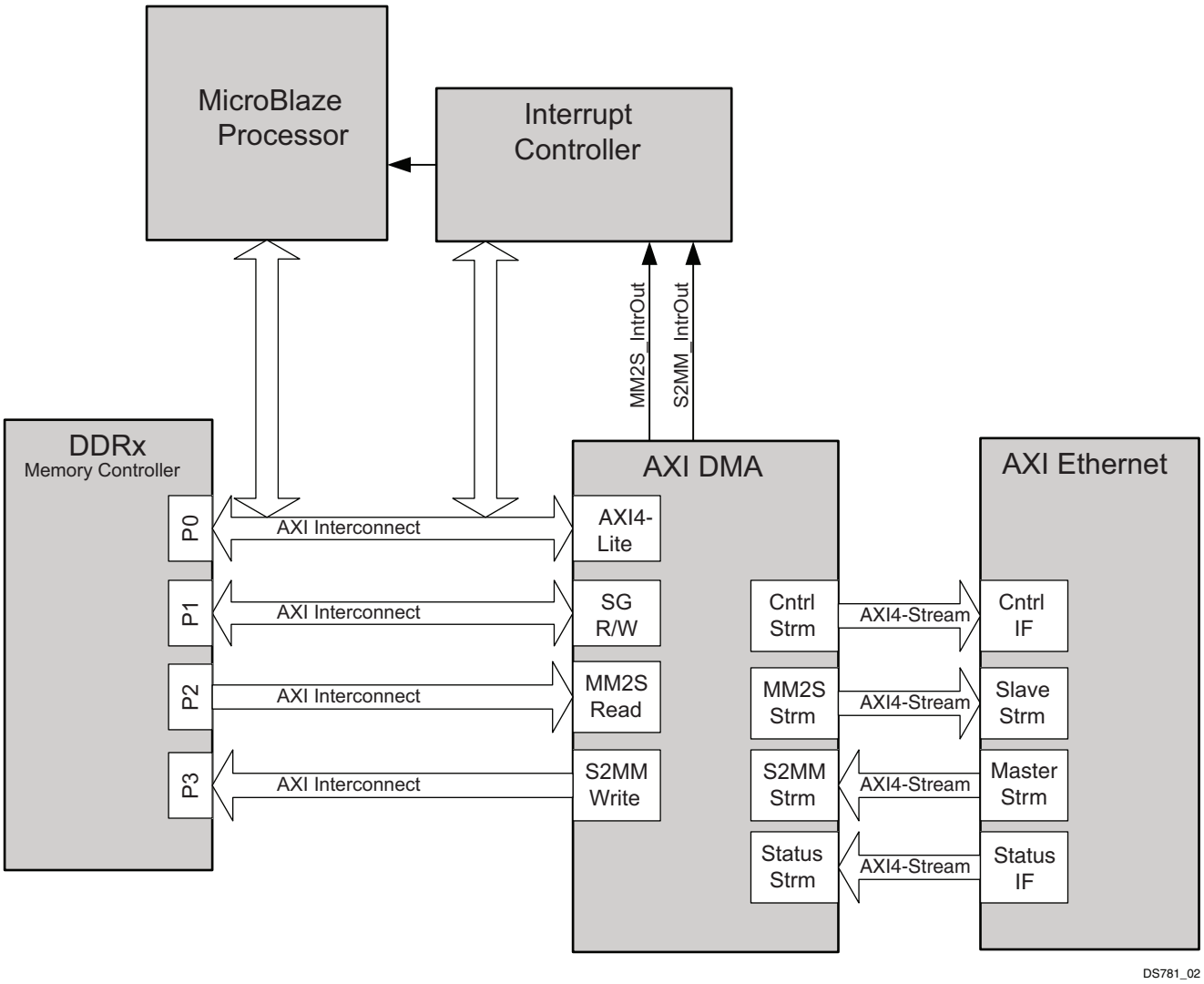

Figure 1-2: **Typical MicroBlaze Processor System Configuration (AXI Ethernet)**

The AXI DMA core can also be connected to a user's system other than with an Ethernet-based AXI IP. Control and Status streams are optional and can be used with Ethernet based IPs only.

Operating System Requirements

For a list of System Requirements, see Xilinx Design Suite: Release Notes Guide.

Feature Summary

- AXI4 compliant
- Optional Independent Scatter/Gather Direct Memory Access (DMA) support
  - Provides off-loading of DMA management work from the CPU
  - Provides fetch and update of transfer descriptors independent from primary data bus
  - Allows descriptor placement to be in any memory-mapped location separate from data buffers. For example, descriptors can be placed in block RAM.
  - Provides optional local transfer descriptor queuing
  - Provides optional User Application Fields
  - Provides Tail Descriptor Mode
  - Provides optional up to 16 multiple channels of data movement on both MM2S and S2MM paths in Scatter/Gather mode
  - Provides optional 2D transfers
  - Provides optional keyhole operation
- Optional Simple DMA Support

  A lower performance but less Field Programmable Gate Array (FPGA) resource intensive mode can be enabled by excluding the Scatter Gather engine (C_INCLUDE_SG = 0). This mode is referenced in this document as Simple DMA mode. In this mode transfers are commanded by setting a Source Address (for MM2S) or Destination Address (For S2MM) and then specifying a byte count in a length register.

  A non-zero write to the length register for the specific channel triggers the transfer. On MM2S, length bytes are read from the Source Address and transmitted out the MM2S AXI4-Stream Interface.

  On S2MM, the length value indicates to AXI DMA the size of the storage buffer in bytes; therefore, length values that are written must be equal to or larger than the largest packet that is received on S2MM. A length value written that is smaller than the received packet produces undefined results.

  On S2MM the number of bytes in the received packet are written to the Destination Address and the length register is updated with the number of bytes transferred.

- Primary AXI4 Memory Map data width support of 32, 64, 128, 256, 512 and 1024 bits
- Primary AXI4-Stream data width support of 8, 16, 32, 64, 128, 256, 512 and 1024 bits
- Optional Data Re-Alignment Engine

  Allows data realignment to the byte (8 bits) level on the primary memory map and stream datapaths

- Optional AXI Control and Status Streams

  Provides optional Control Stream for MM2S Channel and Status Stream for the S2MM channel to off-load low-bandwidth control and status from high-bandwidth datapath.

Applications

The AXI DMA provides high-speed data movement between system memory and an AXI4-Stream-based target IP such as AXI Ethernet.

Licensing and Ordering Information

This Xilinx LogiCORE™ IP module is provided at no additional cost with the Xilinx Vivado™ Design Suite and ISE® Design Suite Embedded Edition tools under the terms of the Xilinx End User License.

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

Standards Compliance

AXI4, AXI4-Lite and AXI4-Stream compliant

Performance

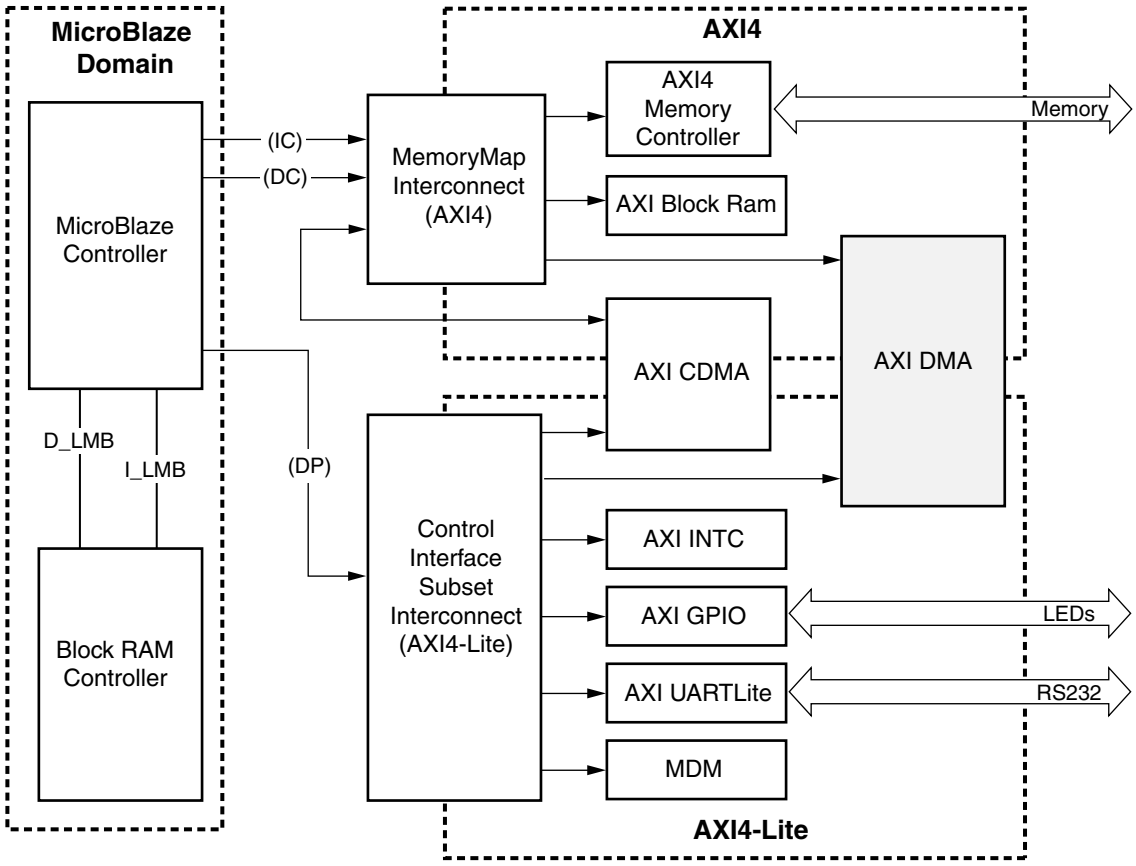

Figure 2-1: **FPGA System Configuration Diagram**

Maximum Frequencies

The target FPGA was filled with logic to drive the Lookup Table (LUT) and block RAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target F_{MAX} numbers are shown in Table 2-1.

Table 2-1: **System Performance**

Family	Device	Speed Grade	F_{MAX}		
			AXI4	AXI4-Lite	AXI4-Stream
Spartan-6	xc6slx45t (1)	-2	150 MHz	100 MHz	150 MHz
Virtex-6	xc6vlx240t (2)	-1	200 MHz	150 MHz	200 MHz
Virtex-7	xc7vx485t (2)	-1	200 MHz	150 MHz	200 MHz

Notes:

1. In Spartan-6 FPGAs, MicroBlaze™ processor frequency is 80 MHz.
2. In Virtex-6 and Virtex-7 FPGAs, processor MicroBlaze frequency is 150 MHz.

Latency and Throughput

Table 2-2 and Table 2-3 describe the throughput and latency for the AXI DMA. The tables provide performance information for a typical configuration. The throughput test consisted of transferring 10000 bytes on MM2S and S2MM side.

Throughput is measured from completion of descriptor fetches (DMACR.Idle = 1) to frame count interrupt assertion. Latency is measured on both the mm2s and s2mm path. Table 2-2 the AXI DMA core latency cycles only and does not include system dependent latency or throttling.

AXI DMA Configuration

- C_PRMRY_IS_ACLK_ASYNC = 0
- C_INCLUDE_SG = 1
- C_SG_INCLUDE_DESC_QUEUE = 1
- C_INCLUDE_MM2S = 1
- C_MM2S_BURST_SIZE = 16
- C_M_AXI_MM2S_DATA_WIDTH = 32
- C_M_AXIS_MM2S_TDATA_WIDTH = 32
- C_INCLUDE_S2MM = 1
- C_S2MM_BURST_SIZE = 16
- C_M_AXI_S2MM_DATA_WIDTH = 32
- C_S_AXIS_S2MM_TDATA_WIDTH = 32

Table 2-2: **AXI DMA Latency Numbers**

Description	Clocks
MM2S Channel	
Tail Descriptor write to m_axi_sg_arvalid	10
m_axi_sg_arvalid to m_axi_mm2s_arvalid	28
m_axi_mm2s_arvalid to m_axis_mm2s_tvalid	6
S2MM Channel	
Tail Descriptor write to m_axi_sg_arvalid	10
s_axis_s2mm_tvalid to m_axi_s2mm_awvalid	39

Table 2-3: **AXI DMA Throughput Numbers**

Channel	Clock Frequency (MHz)	Bytes Transferred	Total Throughput (MB/s)	Percent of Theoretical
MM2S[a]	100	10000	399.04	99.76
S2MM[b]	100	10000	298.59	74.64

a. The MM2S throughput is measured between first ARVALID on Memory Map side to the TLAST on streaming side.
b. The S2MM throughput is measured between first TVALID on streaming side to last WLAST on the Memory Map side.

Resource Utilization

Resources required for the AXI DMA core have been estimated for 7 series and Zynq™-7000 devices (Table 2-4), Virtex®-6 (Table 2-5) and Spartan®-6 FPGAs (Table 2-6). These values were generated using the Xilinx EDK tools v14.2. They are derived from post-MAP reports and can change during PAR.

Table 2-4: **7 Series and Zynq-7000 Devices Resource Estimates**

C_M_AXI_MM2S_DATA_WIDTH	C_M_AXI_S2MM_DATA_WIDTH	C_M_AXIS_MM2S_TDATA_WIDTH	C_S_AXIS_S2MM_TDATA_WIDTH	C_S2MM_BURST_SIZE	C_MM2S_BURST_SIZE	C_SG_INCLUDE_DESC_QUEUE	C_INCLUDE_MM2S_DRE	C_INCLUDE_S2MM_DRE	C_DLYTMR_RESOLUTION	C_PRMRY_IS_ACLK_ASYNC	C_SG_LENGTH_WIDTH	C_INCLUDE_MM2S	C_INCLUDE_S2MM	C_ENABLE_MULTI_CHANNEL	C_NUM_S2MM_CHANNELS	C_NUM_MM2S_CHANNELS	C_SG_INCLUDE_STSCNTRL_STRM	C_SG_USE_STSAPP_LENGTH	Slices	Slice LUTS	Slice s	Block RAM
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	0	1	1	0	0	4553	3541	1731	2
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	0	1	1	1	1	4775	3881	1784	5
64	64	64	64	32	32	1	1	1	1250	0	23	1	1	0	1	1	0	0	4727	3706	1206	6
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	1	2	2	0	0	5152	3879	1855	2
32	32	32	32	16	16	0	1	1	1250	0	23	1	1	1	8	8	0	0	5518	4119	2126	2
64	64	64	64	16	16	0	1	1	1250	0	23	1	1	1	8	8	0	0	6156	4807	2339	4

Table 2-5: **Virtex-6 FPGA Resource Estimates**

C_M_AXI_MM2S_DATA_WIDTH	C_M_AXI_S2MM_DATA_WIDTH	C_M_AXIS_MM2S_TDATA_WIDTH	C_S_AXIS_S2MM_TDATA_WIDTH	C_S2MM_BURST_SIZE	C_MM2S_BURST_SIZE	C_SG_INCLUDE_DESC_QUEUE	C_INCLUDE_MM2S_DRE	C_INCLUDE_S2MM_DRE	C_DLYTMR_RESOLUTION	C_PRMRY_IS_ACLK_ASYNC	C_SG_LENGTH_WIDTH	C_INCLUDE_MM2S	C_INCLUDE_S2MM	C_ENABLE_MULTI_CHANNEL	C_NUM_S2MM_CHANNELS	C_NUM_MM2S_CHANNELS	C_SG_INCLUDE_STSCNTRL_STRM	C_SG_USE_STSAPP_LENGTH	Slices	Slice LUTS	Slice s	Block RAM
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	0	1	1	0	0	4555	3433	1829	4
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	0	1	1	1	1	4776	3764	1949	5
64	64	64	64	32	32	1	1	1	1250	0	23	1	1	0	1	1	0	0	4726	3303	1801	6
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	1	2	2	0	0	5152	3608	1953	2
32	32	32	32	16	16	0	1	1	1250	0	23	1	1	1	8	8	0	0	5518	4139	2122	2
64	64	64	64	16	16	0	1	1	1250	0	23	1	1	1	8	8	0	0	6156	4546	2398	4

Table 2-6: **Spartan-6 FPGA Resource Estimates**

C_M_AXI_MM2S_DATA_WIDTH	C_M_AXI_S2MM_DATA_WIDTH	C_M_AXIS_MM2S_TDATA_WIDTH	C_S_AXIS_S2MM_TDATA_WIDTH	C_S2MM_BURST_SIZE	C_MM2S_BURST_SIZE	C_SG_INCLUDE_DESC_QUEUE	C_INCLUDE_MM2S_DRE	C_INCLUDE_S2MM_DRE	C_DLYTMR_RESOLUTION	C_PRMRY_IS_ACLK_ASYNC	C_SG_LENGTH_WIDTH	C_INCLUDE_MM2S	C_INCLUDE_S2MM	C_ENABLE_MULTI_CHANNEL	C_NUM_S2MM_CHANNELS	C_NUM_MM2S_CHANNELS	C_SG_INCLUDE_STSCNTRL_STRM	C_SG_USE_STSAPP_LENGTH	Slices	Slice LUTS	Slice s	Block RAM
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	0	1	1	0	0	4568	3442	1550	6
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	0	1	1	1	1	4802	3659	1757	6
64	64	64	64	32	32	1	1	1	1250	0	23	1	1	0	1	1	0	0	4728	3345	1604	8
32	32	32	32	16	16	1	1	1	1250	0	23	1	1	1	2	2	0	0	5160	3700	1863	4
32	32	32	32	16	16	0	1	1	1250	0	23	1	1	1	8	8	0	0	5526	4311	1870	4
64	64	64	64	16	16	0	1	1	1250	0	23	1	1	1	8	8	0	0	6164	4841	2108	6

Port Descriptions

The AXI DMA I/O signals are described in Table 2-7.

Table 2-7: **I/O Signal Description**

Signal Name	Interface	Signal Type	Init Status	Description
s_axi_lite_aclk	Clock	I		AXI DMA AXI4-Lite Clock. Must be less than or equal to axi_sg_aclk for asynchronous mode. (C_PRMRY_IS_ACLK_ASYNC=1).
m_axi_sg_aclk	Clock	I		AXI DMA Scatter Gather Clock. Scatter Gather clock must be less than or equal to the slowest of m_axi_mm2s_aclk or m_axi_s2mm_aclk for asynchronous mode. (C_PRMRY_IS_ACLK_ASYNC=1).
m_axi_mm2s_aclk	Clock	I		AXI DMA MM2S Primary Clock
m_axi_s2mm_aclk	Clock	I		AXI DMA S2MM Primary Clock
axi_resetn	Reset	I		AXI DMA Reset. Active-Low reset. When asserted low, resets entire AXI DMA core. Must be synchronous to s_axi_lite_aclk.
mm2s_introut	Interrupt	O	0	Interrupt Out for Memory Map to Stream Channel.
s2mm_introut	Interrupt	O	0	Interrupt Out for Stream to Memory Map Channel.
AXI4-Lite Interface Signals				
s_axi_lite_awvalid	S_AXI_LITE	I		AXI4-Lite Write Address Channel Write Address Valid. • 1 = Write address is valid. • 0 = Write address is not valid.
s_axi_lite_awready	S_AXI_LITE	O	0	AXI4-Lite Write Address Channel Write Address Ready. Indicates DMA ready to accept the write address. • 1 = Ready to accept address. • 0 = Not ready to accept address.
s_axi_lite_awaddr(9:0)	S_AXI_LITE	I		AXI4-Lite Write Address Bus
s_axi_lite_wvalid	S_AXI_LITE	I		AXI4-Lite Write Data Channel Write Data Valid. • 1 = Write data is valid. • 0 = Write data is not valid.
s_axi_lite_wready	S_AXI_LITE	O	0	AXI4-Lite Write Data Channel Write Data Ready. Indicates DMA ready to accept the write data. • 1 = Ready to accept data. • 0 = Not ready to accept data.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
s_axi_lite_wdata(31:0)	S_AXI_LITE	I		AXI4-Lite Write Data Bus
s_axi_lite_bresp(1:0)	S_AXI_LITE	O	zeros	AXI4-Lite Write Response Channel. Indicates results of the write transfer. The AXI DMA Lite interface always responds with OKAY. • 00b = OKAY - Normal access has been successful. • 01b = EXOKAY - Not supported. • 10b = SLVERR - Not supported. • 11b = DECERR - Not supported.
s_axi_lite_bvalid	S_AXI_LITE	O	0	AXI4-Lite Write Response Channel Response Valid. Indicates response is valid. • 1 = Response is valid. • 0 = Response is not valid.
s_axi_lite_bready	S_AXI_LITE	I		AXI4-Lite Write Response Channel Ready. Indicates target is ready to receive response. • 1 = Ready to receive response. • 0 = Not ready to receive response.
s_axi_lite_arvalid	S_AXI_LITE	I		AXI4-Lite Read Address Channel Read Address Valid. • 1 = Read address is valid. • 0 = Read address is not valid.
s_axi_lite_arready	S_AXI_LITE	O	0	AXI4-Lite Read Address Channel Read Address Ready. Indicates DMA ready to accept the read address. • 1 = Ready to accept address. • 0 = Not ready to accept address.
s_axi_lite_araddr(9:0)	S_AXI_LITE	I		AXI4-Lite Read Address Bus.
s_axi_lite_rvalid	S_AXI_LITE	O	0	AXI4-Lite Read Data Channel Read Data Valid • 1 = Read data is valid • 0 = Read data is not valid
s_axi_lite_rready	S_AXI_LITE	I		AXI4-Lite Read Data Channel Read Data Ready. Indicates target ready to accept the read data. • 1 = Ready to accept data. • 0 = Not ready to accept data.
s_axi_lite_rdata(31:0)	S_AXI_LITE	O	zeros	AXI4-Lite Read Data Bus
s_axi_lite_rresp(1:0)	S_AXI_LITE	O	zeros	AXI4-Lite Read Response Channel Response. Indicates results of the read transfer. The AXI DMA Lite interface always responds with OKAY. • 00b = OKAY - Normal access has been successful. • 01b = EXOKAY - Not supported • 10b = SLVERR - Not supported • 11b = DECERR - Not supported

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
MM2S Memory Map Read Interface Signals				
m_axi_mm2s_araddr (C_M_AXI_MM2S_ADDR_ WIDTH-1: 0)	M_AXI_MM2S	O	zeros	Read Address Channel Address Bus
m_axi_mm2s_arlen(7:0)	M_AXI_MM2S	O	zeros	Read Address Channel Burst Length. In data beats - 1.
m_axi_mm2s_arsize(2:0)	M_AXI_MM2S	O	zeros	Read Address Channel Burst Size. Indicates width of burst transfer. • 000b = 1 byte (8-bit wide burst) • 001b = 2 bytes (16-bit wide burst) • 010b = 4 bytes (32-bit wide burst) • 011b = 8 bytes (64-bit wide burst) • 100b = 16 bytes (128-bit wide burst) • 101b = 32 bytes (256-bit wide burst) • 110b = 64 bytes (512-bit wide burst) • 111b = 128 bytes (1024 bit wide burst)
m_axi_mm2s_arburst(1:0)	M_AXI_MM2S	O	zeros	Read Address Channel Burst Type. Indicates type burst. • 00b = FIXED - Fixed Address • 01b = INCR - Incrementing address • 10b = WRAP - Not supported • 11b = Reserved
m_axi_mm2s_arprot(2:0)	M_AXI_MM2S	O	010b	Read Address Channel Protection. Always driven with a constant output of 010b.
m_axi_mm2s_arcache(3:0)	M_AXI_MM2S	O	0011b	Read Address Channel Cache. In multichannel mode, these bits are driven from fields of MM2S (Tx) descriptor. See Multichannel DMA Support for details.
m_axi_mm2s_aruser(3:0)	M_AXI_MM2S	O	0	Read Address Channel User. A side band signal used for user-defined information. These bits are applicable only in multichannel mode. In multichannel mode, these bits are driven from fields of MM2S (Tx) descriptor. See Multichannel DMA Support for details.
m_axi_mm2s_arvalid	M_AXI_MM2S	O	0	Read Address Channel Read Address Valid. Indicates m_axi_mm2s_araddr is valid. • 1 = Read address is valid. • 0 = Read address is not valid.
m_axi_mm2s_arready	M_AXI_MM2S	I		Read Address Channel Read Address Ready. Indicates target is ready to accept the read address. • 1 = Target ready to accept address. • 0 = Target not ready to accept address.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
m_axi_mm2s_rdata (C_M_AXI_MM2S_DATA_ WIDTH-1: 0)	M_AXI_MM2S	I		Read Data Channel Read Data
m_axi_mm2s_rresp(1:0)	M_AXI_MM2S	I		Read Data Channel Response. Indicates results of the read transfer. • 00b = OKAY - Normal access has been successful. • 01b = EXOKAY - Not supported. • 10b = SLVERR - Slave returned error on transfer. • 11b = DECERR - Decode error, transfer targeted unmapped address.
m_axi_mm2s_rlast	M_AXI_MM2S	I		Read Data Channel Last. Indicates the last data beat of a burst transfer. • 1 = Last data beat • 0 = Not last data beat
m_axi_mm2s_rvalid	M_AXI_MM2S	I		Read Data Channel Data Valid. Indicates m_axi_mm2s_rdata is valid. • 1 = Valid read data • 0 = Not valid read data
m_axi_mm2s_rready	M_AXI_MM2S	O	0	Read Data Channel Ready. Indicates the read channel is ready to accept read data. • 1 = Ready • 0 = Not ready
MM2S Master Stream Interface Signals				
mm2s_prmry_reset_out_n	M_AXIS_MM2S	O	1	Primary MM2S Reset Out. Active-Low reset.
m_axis_mm2s_tdata (C_M_AXIS_MM2S_TDATA_ WIDTH-1: 0)	M_AXIS_MM2S	O	zeros	AXI4-Stream Stream Data Out
m_axis_mm2s_tkeep (C_M_AXIS_MM2S_TDATA_ WIDTH/8-1: 0)	M_AXIS_MM2S	O	zeros	AXI4-Stream Write Keep Out. Indicates valid bytes on stream data.
m_axis_mm2s_tuser(3:0)	M_AXIS_MMS	O	0	AXI4-Stream Tuser. A side band signal used for user-defined information. These bits are not defined in legacy mode. In multichannel mode, these bits are driven from fields of MM2S (Tx) descriptor. See Multichannel DMA Support for details.
m_axis_mm2s_tid(4:0)	M_AXIS_MM2S	O	0	AXI4-Stream TID. Used as stream identifier. These bits are not defined in legacy mode. In multichannel mode, these bits are driven from fields of MM2S (Tx) descriptor. See Multichannel DMA Support for details.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
m_axis_mm2s_tdest(4:0)	M_AXIS_MM2S	O	0	AXI4-Stream TDEST. Indicates the Slave number on streaming side to which packets are destined. These bits are not defined in legacy mode. In multichannel mode, these bits are driven from fields of MM2S (Tx) descriptor. See Multichannel DMA Support for details.
m_axis_mm2s_tvalid	M_AXIS_MM2S	O	0	AXI4-Stream Stream Valid Out. Indicates stream data bus, m_axis_mm2s_tdata, is valid • 1 = Write data is valid. • 0 = Write data is not valid.
m_axis_mm2s_tready	M_AXIS_MM2S	I		AXI4-Stream Ready. Indicates to MM2S channel target is ready to receive stream data. • 1 = Ready to receive data. • 0 = Not ready to receive data.
m_axis_mm2s_tlast	M_AXIS_MM2S	O	0	AXI4-Stream Last. Indicates last data beat of stream data. • 1 = Last data beat • 0 = Not last data beat
MM2S Master Control Stream Interface Signals				
mm2s_cntrl_reset_out_n	M_AXIS_CNTRL	O	1	Control Reset Out. Active-Low reset.
m_axis_mm2s_cntrl_tdata (C_M_AXIS_MM2S_CNTRL_TDATA_WIDTH-1: 0)	M_AXIS_CNTRL	O	zeros	AXI Control Stream Stream Data Out
m_axis_mm2s_cntrl_tkeep (C_M_AXIS_MM2S_CNTRL_TDATA_WIDTH/8-1: 0)	M_AXIS_CNTRL	O	zeros	AXI Control Stream Write Keep Out. Indicates valid bytes on stream data.
m_axis_mm2s_cntrl_tvalid	M_AXIS_CNTRL	O	0	AXI Control Stream Stream Valid Out. Indicates stream data bus, m_axis_mm2s_cntrl_tdata, is valid. • 1 = Write data is valid. • 0 = Write data is not valid.
m_axis_mm2s_cntrl_tready	M_AXIS_CNTRL	I		AXI Control Stream Ready. Indicates to MM2S channel target is ready to receive stream data. • 1 = Ready to receive data • 0 = Not ready to receive data
m_axis_mm2s_cntrl_tlast	M_AXIS_CNTRL	O	0	AXI Control Stream Last. Indicates last data beat of stream data. • 1 = Last data beat • 0 = Not last data beat

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
S2MM Memory Map Write Interface Signals				
m_axi_s2mm_awaddr (C_M_AXI_S2MM_ADDR_ WIDTH-1: 0)	M_AXI_S2MM	O	zeros	Write Address Channel Address Bus
m_axi_s2mm_awlen(7: 0)	M_AXI_S2MM	O	zeros	Write Address Channel Burst Length. In data beats - 1.
m_axi_s2mm_awsize(2: 0)	M_AXI_S2MM	O	zeros	Write Address Channel Burst Size. Indicates with of burst transfer. • 000b = 1 byte (8-bit wide burst) • 001b = 2 bytes (16-bit wide burst) • 010b = 4 bytes (32-bit wide burst) • 011b = 8 bytes (64-bit wide burst). • 100b = 16 bytes (128-bit wide burst) • 101b = 32 bytes (256-bit wide burst) • 110b = 64 bytes (512-bit wide burst) • 111b = 128 bytes (1024-bit wide burst)
m_axi_s2mm_awburst(1:0)	M_AXI_S2MM	O	zeros	Write Address Channel Burst Type. Indicates type burst. • 00b = FIXED - Fixed Address • 01b = INCR - Incrementing address • 10b = WRAP - Not supported • 11b = Reserved
m_axi_s2mm_awprot(2:0)	M_AXI_S2MM	O	010b	Write Address Channel Protection. This is always driven with a constant output of 0010b.
m_axi_s2mm_awcache(3:0)	M_AXI_S2MM	O	0011b	Write Address Channel Cache. In multichannel mode, these bits are driven from fields of MM2S (Tx) descriptor. See Multichannel DMA Support for details.
m_axi_s2mm_awuser(3:0)	M_AXI_S2MM	O	0	Write Address Channel User. A side band signal used for user-defined information. These bits are not defined in legacy mode. In multichannel mode, these bits are driven from fields of S2MM (Rx) descriptor. See Multichannel DMA Support for details.
m_axi_s2mm_awaddr	M_AXI_S2MM	O	0	Write Address Channel Write Address Valid. Indicates if mm2s_axim_awaddr is valid. • 1 = Write Address is valid. • 0 = Write Address is not valid.
m_axi_s2mm_awready	M_AXI_S2MM	I		Write Address Channel Write Address Ready. Indicates target is ready to accept the write address. • 1 = Target ready to accept address. • 0 = Target not ready to accept address.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
m_axi_s2mm_wdata (C_M_AXI_S2MM_DATA_WIDTH-1: 0)	M_AXI_S2MM	O	zeros	Write Data Channel Write Data Bus
m_axi_s2mm_wstrb (C_M_AXI_S2MM_DATA_WIDTH/8 - 1: 0)	M_AXI_S2MM	O	zeros	Write Data Channel Write Strobe Bus. Indicates which bytes are valid in the write data bus. This value is passed from the stream side strobe bus.
m_axi_s2mm_wlast	M_AXI_S2MM	O	0	Write Data Channel Last. Indicates the last data beat of a burst transfer. • 1 = Last data beat • 0 = Not last data beat
m_axi_s2mm_wvalid	M_AXI_S2MM	O	0	Write Data Channel Data Valid. Indicates m_axi_s2mm_wdata is valid. • 1 = Valid write data • 0 = Not valid write data
m_axi_s2mm_wready	M_AXI_S2MM	I		Write Data Channel Ready. Indicates the write channel target is ready to accept write data. • 1 = Target is ready. • 0 = Target is not ready.
m_axi_s2mm_bresp(1:0)	M_AXI_S2MM	I		Write Response Channel Response. Indicates results of the write transfer. • 00b = OKAY - Normal access has been successful. • 01b = EXOKAY - Not supported • 10b = SLVERR - Slave returned error on transfer. • 11b = DECERR - Decode error, transfer targeted unmapped address.
m_axi_s2mm_bvalid	M_AXI_S2MM	I		Write Response Channel Response Valid. Indicates response, m_axi_s2mm_bresp, is valid. • 1 = Response is valid. • 0 = Response is not valid.
m_axi_s2mm_bready	M_AXI_S2MM	O	0	Write Response Channel Ready. Indicates S2MM write channel is ready to receive response. • 1 = Ready to receive response. • 0 = Not ready to receive response.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
S2MM Slave Stream Interface Signals				
s2mm_prmry_reset_out_n	S_AXIS_S2MM	O	1	Primary S2MM Reset Out. Active-Low reset.
s_axis_s2mm_tdata (C_S_AXIS_S2MM_TDATA_WIDTH-1: 0)	S_AXIS_S2MM	I		AXI4-Stream Stream Data In
s_axis_s2mm_tkeep (C_S_AXIS_S2MM_TDATA_WIDTH/8-1: 0)	S_AXIS_S2MM	I		AXI4-Stream Write Keep In. Indicates valid bytes on stream data.
s_axis_s2mm_tvalid	S_AXIS_S2MM	I		AXI4-Stream Stream Valid In. Indicates stream data bus, s_axis_s2mm_tdata, is valid. 1 = Write data is valid. 0 = Write data is not valid.
s_axis_s2mm_tready	S_AXIS_S2MM	O	0	AXI4-Stream Ready. Indicates S2MM channel stream interface ready to receive stream data. 1 = Ready to receive data 0 = Not ready to receive data
s_axis_s2mm_tlast	S_AXIS_S2MM	I		AXI4-Stream Last. Indicates last data beat of stream data. 1 = Last data beat 0 = Not last data beat
m_axis_s2mm_tuser(3:0)	M_AXIS_S2MM	O	0	AXI4-Stream Tuser. A side band signal used for user-defined information. These bits are not defined in legacy mode. In multichannel mode, these bits are driven with the value that is there in the fields of S2MM (Rx) descriptor. See Multichannel DMA Support for details.
m_axis_s2mm_tid(4:0)	M_AXIS_S2MM	O	0	AXI4-Stream TID. Used as stream identifier. These bits are not defined in legacy mode. In multichannel mode, these bits are driven with the value that is there in the fields of S2MM (Rx) descriptor. See Multichannel DMA Support for details.
m_axis_s2mm_tdest(4:0) M_AXIS_S2MM	M_AXIS_S2MM	O	0	AXI4-Stream TDEST. Indicates the Slave number on streaming side to which packets are destined. These bits are not defined in legacy mode. In multichannel mode, these bits are driven with the value that is there in the fields of S2MM (Rx) descriptor. See Multichannel DMA Support for details.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
S2MM Slave Status Stream Interface Signals				
s2mm_sts_reset_out_n	S_AXIS_STS	O	1	AXI Status Stream (STS) Reset Output. Active-Low reset.
s_axis_s2mm_sts_tdata (C_S_AXIS_S2MM_STS_TDATA_WIDTH-1: 0)	S_AXIS_STS	I		AXI Status Stream Stream Data In
s_axis_s2mm_sts_tkeep (C_S_AXIS_S2MM_STS_TDATA_WIDTH/8-1: 0)	S_AXIS_STS	I		AXI Status Stream Write Keep In. Indicates valid bytes on stream data.
s_axis_s2mm_sts_tvalid	S_AXIS_STS	I		AXI Status Stream Stream Valid In. Indicates stream data bus, s_axis_s2mm_sts_tdata, is valid. • 1 = Write data is valid. • 0 = Write data is not valid.
s_axis_s2mm_sts_tready	S_AXIS_STS	O	0	AXI Status Stream Ready. Indicates S2MM channel stream interface ready to receive stream data. • 1 = Ready to receive data. • 0 = Not ready to receive data.
s_axis_s2mm_sts_tlast	S_AXIS_STS	I		AXI Status Stream Last. Indicates last data beat of stream data. • 1 = Last data beat • 0 = Not last data beat
Scatter Gather Memory Map Read Interface Signals				
m_axi_sg_araddr (C_M_AXI_SG_ADDR_WIDTH-1: 0)	M_AXI_SG	O	zeros	Scatter Gather (SG) Read Address Channel Address Bus
m_axi_sg_arlen(7: 0)	M_AXI_SG	O	zeros	Scatter Gather Read Address Channel Burst Length. Length in data beats - 1.
m_axi_sg_arsize(2: 0)	M_AXI_SG	O	zeros	Scatter Gather Read Address Channel Burst Size. Indicates with of burst transfer. • 000b = Not Supported • 001b = Not Supported • 010b = 4 bytes (32-bit wide burst). • 011b = Not Supported • 100b = Not Supported • 101b = Not Supported • 110b = Not Supported • 111b = Not Supported

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
m_axi_sg_arburst(1:0)	M_AXI_SG	O	zeros	Scatter Gather Read Address Channel Burst Type. Indicates type burst. • 00b = FIXED - Not supported • 01b = INCR - Incrementing address • 10b = WRAP - Not supported • 11b = Reserved
m_axi_sg_arprot(2:0)	M_AXI_SG	O	010b	Scatter Gather Read Address Channel Protection. This is always driven with a constant output of 010b.
m_axi_sg_arcache(3:0)	M_AXI_SG	O	0011b	Scatter Gather Read Address Channel Cache. These bits are driven from register bits SG_CACHE(3:0) that is defined in a register space 0x2C. This register is only available when DMA is configured in multichannel mode.
m_axi_sg_aruser(3:0)	M_AXI_SG	O	0	Scatter Gather Read Address Channel User. These bits are not defined in legacy mode. These bits are driven from register bits SG_USER(11:8) defined in register space offset 0x2C. This register is only available when DMA is configured in multichannel mode.
m_axi_sg_arvalid	M_AXI_SG	O	0	Scatter Gather Read Address Channel Read Address Valid. Indicates if m_axi_sg_araddr is valid. • 1 = Read Address is valid. • 0 = Read Address is not valid.
m_axi_sg_arready	M_AXI_SG	I		Scatter Gather Read Address Channel Read Address Ready. Indicates target is ready to accept the read address. • 1 = Target ready to accept address • 0 = Target not ready to accept address
m_axi_sg_rdata (C_M_AXI_SG_DATA_ WIDTH-1: 0)	M_AXI_SG	I		Scatter Gather Read Data Channel Read Data
m_axi_sg_rresp(1:0)	M_AXI_SG	I		Scatter Gather Read Data Channel Response. Indicates results of the read transfer. • 00b = OKAY - Normal access has been successful. • 01b = EXOKAY - Not supported • 10b = SLVERR - Slave returned error on transfer. • 11b = DECERR - Decode error, transfer targeted unmapped address.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
m_axi_sg_rlast	M_AXI_SG	I		Scatter Gather Read Data Channel Last. Indicates the last data beat of a burst transfer. • 1 = Last data beat • 0 = Not last data beat
m_axi_sg_rdata	M_AXI_SG	I		Scatter Gather Read Data Channel Data Valid. Indicates m_sg_aximry_rdata is valid. • 1 = Valid read data • 0 = Not valid read data
m_axi_sg_rready	M_AXI_SG	O	0	Scatter Gather Read Data Channel Ready. Indicates the read channel is ready to accept read data. • 1 = Is ready • 0 = Is not ready
Scatter Gather Memory Map Write Interface Signals				
m_axi_sg_awaddr (C_M_AXI_SG_ADDR_ WIDTH-1: 0)	M_AXI_SG	O	zeros	Scatter Gather Write Address Channel Address Bus
m_axi_sg_awlen(7: 0)	M_AXI_SG	O	zeros	Scatter Gather Write Address Channel Burst Length. Length in data beats - 1.
m_axi_sg_awsize(2: 0)	M_AXI_SG	O	zeros	Scatter Gather Write Address Channel Burst Size. Indicates with of burst transfer,000b = Not Supported by AXI DMA SG Engine • 001b = Not Supported by AXI DMA SG Engine • 010b = 4 bytes (32-bit wide burst), • 011b = Not Supported • 100b = Not Supported • 101b = Not Supported • 110b = Not Supported • 111b = Not Supported
m_axi_sg_awburst(1:0)	M_AXI_SG	O	zeros	Scatter Gather Write Address Channel Burst Type. Indicates type burst. • 00b = FIXED - Not supported • 01b = INCR - Incrementing address • 10b = WRAP - Not supported • 11b = Reserved
m_axi_sg_awprot(2:0)	M_AXI_SG	O	010b	Scatter Gather Write Address Channel Protection. This is always driven with a constant output of 010b.
m_axi_sg_awcache(3:0)	M_AXI_SG	O	0011b	Scatter Gather Write Address Channel Cache. This is always driven with a constant output of 0011b.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
m_axi_sg_awuser(3:0)	M_AXI_SG	O	zeros	Scatter Gather Write Address Channel User. These bits are not defined in legacy mode. These bits are driven from register bits SG_USER(11:8) defined in register space offset 0x2C. This register is only available when DMA is configured in multichannel mode.
m_axi_sg_awvalid	M_AXI_SG	O	0	Scatter Gather Write Address Channel Write Address Valid. Indicates if m_axi_sg_awaddr is valid. • 1 = Write Address is valid. • 0 = Write Address is not valid.
m_axi_sg_awready	M_AXI_SG	I		Scatter Gather Write Address Channel Write Address Ready. Indicates target is ready to accept the write address. • 1 = Target ready to accept address. • 0 = Target not ready to accept address.
m_axi_sg_wdata (C_M_AXI_SG_DATA_ WIDTH-1 : 0)	M_AXI_SG	O	zeros	Scatter Gather Write Data Channel Write Data Bus
m_axi_sg_wstrb (C_M_AXI_SG_DATA_ WIDTH/8 - 1: 0)	M_AXI_SG	O	1111b	Scatter Gather Write Data Channel Write Strobe Bus. All bytes always valid.
m_axi_sg_wlast	M_AXI_SG	O	0	Scatter Gather Write Data Channel Last. Indicates the last data beat of a burst transfer. • 1 = Last data beat • 0 = Not last data beat
m_axi_sg_wvalid	M_AXI_SG	O	0	Scatter Gather Write Data Channel Data Valid. Indicates M_SG_AXIMry_WDATA is valid. • 1 = Valid write data • 0 = Not valid write data
m_axi_sg_wready	M_AXI_SG	I		Scatter Gather Write Data Channel Target Ready. Indicates the write channel target is ready to accept write data. • 1 = Target is ready. • 0 = Target is not ready.

Table 2-7: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
m_axi_sg_bresp(1:0)	M_AXI_SG	I		Scatter Gather Write Response Channel Response. Indicates results of the write transfer. • 00b = OKAY - Normal access has been successful. • 01b = EXOKAY - Not supported • 10b = SLVERR - Slave returned error on transfer. • 11b = DECERR - Decode error, transfer targeted unmapped address.
m_axi_sg_bvalid	M_AXI_SG	I		Scatter Gather Write Response Channel Response Valid. Indicates response, m_axi_sg_bresp, is valid. • 1 = Response is valid. • 0 = Response is not valid.
m_axi_sg_bready	M_AXI_SG	O	0	Scatter Gather Write Response Channel Ready. Indicates source is ready to receive response. • 1 = Ready to receive response • 0 = Not ready to receive response

Design Parameters

The AXI DMA Design Parameters are listed and described in Table 2-8.

Table 2-8: **Design Parameter Description**

Parameter Name	Allowable Values	Default Values	VHDL Type	Feature/Description
AXI DMA General Parameters				
C_S_AXI_LITE_DATA_WIDTH	32	32	integer	Data width in bits of AXI4-Lite Interface
C_S_AXI_LITE_ADDR_WIDTH	10	10	integer	Address width in bits of AXI4-Lite Interface
C_DLYTMR_RESOLUTION	1 - 1000000	125	integer	Resolution of the interrupt delay timer in m_axi_sg_aclk cycles
C_PRMRY_IS_ACLK_ASYNC	0, 1	0	integer	Primary clock is asynchronous.
C_ENABLE_MULTI_CHANNEL	0,1	0	integer	Enables multichannel mode. Also enables m_axi_mm2s_arcache, m_axi_s2mm_awcache to be controlled by Descriptor field.
C_FAMILY	virtex6, spartan6, virtex7, kintex7, Artix7, Zynq	virtex6	String	Specifies the target Field Programmable Gate Array (FPGA) family

Table 2-8: **Design Parameter Description *(Cont'd)***

Parameter Name	Allowable Values	Default Values	VHDL Type	Feature/Description
Scatter Gather Engine Parameters				
C_INCLUDE_SG	0,1	1	Integer	Include or Exclude Scatter Gather Engine • 0 = Enables Simple DMA mode. • 1 = Enables Scatter/Gather Mode.
C_M_AXI_SG_DATA_WIDTH	32	32	integer	Data width of AXI Scatter Gather Engine
C_M_AXI_SG_ADDR_WIDTH	32	32	integer	Address width of AXI Scatter Gather Engine
C_SG_INCLUDE_DESC_QUEUE	0,1	0	integer	Include or Exclude Descriptor Queuing • 0 = Exclude Descriptor Queue • 1 = Include Descriptor Queue
C_SG_INCLUDE_STSCNTRL_STRM	0,1	1	integer	Include or Exclude Control and Status Streams • 0 = Exclude Status and Control Streams • 1 = Include Status and Control Streams
C_SG_USE_STSAPP_LENGTH	0,1	1	integer	Enable use of receive length in Status Stream Application (APP) Field
C_SG_LENGTH_WIDTH	8 to 23	14	integer	Width of the Buffer Length and Transferred Bytes fields as well as receive length value in the status stream application word
C_M_AXIS_MM2S_CNTRL_TDATA_WIDTH	32	32	integer	AXI Control Stream Data Width
C_S_AXIS_S2MM_STS_TDATA_WIDTH	32	32	integer	AXI Status Stream Data Width

Table 2-8: **Design Parameter Description *(Cont'd)***

Parameter Name	Allowable Values	Default Values	VHDL Type	Feature/Description
Memory Map to Stream Parameters				
C_INCLUDE_MM2S	0,1	1	integer	Include or exclude the Memory Map to Stream channel. When excluded, all unused ports are tied off or driven to zero. This also excludes MM2S AXI Control Stream. • 0 = Exclude MM2S • 1 = Include MM2S
C_INCLUDE_MM2S_DRE	0,1	0	integer	Include or exclude the Memory Map to Stream channel Data Realignment Engine (DRE). • 0 = Exclude DRE • 1 = Include DRE ***Note:*** DRE support is not available for AXI4-Stream data widths of 128 bits and more.
C_M_AXI_MM2S_ADDR_WIDTH	32	32	integer	Address width of AXI4 Memory Map on the Memory Map to Stream interface
C_M_AXI_MM2S_DATA_WIDTH	32,64,128, 256,512, 1024	32	integer	Data width of AXI4 Memory Map on the Memory Map to Stream Interface
C_M_AXIS_MM2S_TDATA_WIDTH	8,16, 32,64,128,25 6,512,1024	32	integer	Data width of AXI4-Stream on the Stream to Memory Map Interface. Width must be equal or less than C_M_AXI_MM2S_DATA_WIDTH.
C_MM2S_BURST_SIZE	16,32,64,128, 256,512,1024	16	integer	Maximum burst size per burst request on Memory Map Read interface
C_NUM_MM2S_CHANNELS	1-16	1	integer	Number of channels supported on MM2S side. This parameter is valid only when C_ENABLE_MULTI_CHANNEL is set to 1; otherwise its ignored. ***Note:*** If channel selected is 4, it supports channels 0,1,2,3. It cannot support random set of four channels.

Table 2-8: **Design Parameter Description *(Cont'd)***

Parameter Name	Allowable Values	Default Values	VHDL Type	Feature/Description
Stream to Memory Map Parameters				
C_INCLUDE_S2MM	0,1	1	integer	Include or exclude the Stream to Memory Map. When excluded, all unused ports are tied off or driven to zero. This also excludes S2MM AXI Status Stream. • 0 = Exclude S2MM • 1 = Include S2MM
C_INCLUDE_S2MM_DRE	0,1	0	integer	Include or exclude the Stream to Memory Map channel Data Realignment Engine. • 0 = Exclude DRE • 1 = Include DRE ***Note:*** DRE support is not available for AXI4-Stream data widths of 128 bits and more.
C_M_AXI_S2MM_ADDR_WIDTH	32	32	integer	Address width of AXI4 Memory Map on the Stream to Memory Map interface
C_M_AXI_S2MM_DATA_WIDTH	32,64,128,256,512,1024	32	integer	Data width of AXI4 Memory Map on the Stream to Memory Map Interface
C_S_AXIS_S2MM_TDATA_WIDTH	32,64,128,256,512,1024	32	integer	Data width of AXI4-Stream on the Stream to Memory Map Interface. Width must be equal or less than C_M_AXI_S2MM_DATA_WIDTH.
C_S2MM_BURST_SIZE	16,32,64, 128,256	16	integer	Maximum burst size per burst request on Memory Map Write interface
C_NUM_S2MM_CHANNELS	1-16	1	integer	Number of channels supported on S2MM side. This parameter is valid only when C_ENABLE_MULTI_CHANNEL is set to 1; otherwise its ignored.

In addition to the parameters listed in this table, there are also parameters that are inferred for each AXI interface in the EDK tools. Through the design, these inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to the AXI interface, see *AXI Interconnect IP Data Sheet* (DS768).

Register Space

The AXI DMA core register space for Scatter / Gather Mode (C_INCLUDE_SG = 1) is shown in Table 2-9. The AXI DMA core register space for Simple DMA Mode (C_INCLUDE_SG = 0) is shown in Table 2-10. The AXI DMA Registers are memory-mapped into non-cacheable memory space. This memory space must be aligned on a AXI word (32-bit) boundary.

AXI DMA Register Address Mapping

Table 2-9: **AXI DMA Scatter / Gather Mode Register Address Mapping (C_INCLUDE_SG = 1)**

Address Space Offset[1]	Name	Description
00h	MM2S_DMACR	MM2S DMA Control Register
04h	MM2S_DMASR	MM2S DMA Status Register
08h	MM2S_CURDESC	MM2S Current Descriptor Pointer
0Ch	Reserved	N/A
10h	MM2S_TAILDESC	MM2S Tail Descriptor Pointer
14h to 2Bh	Reserved	N/A
2Ch [2]	SG_CTL	Scatter/Gather User and Cache
30h	S2MM_DMACR	S2MM DMA Control Register
34h	S2MM_DMASR	S2MM DMA Status Register
38h	S2MM_CURDESC	S2MM Current Descriptor Pointer
3Ch	Reserved	N/A
40h	S2MM_TAILDESC	S2MM Tail Descriptor Pointer

Notes:

1. Address Space Offset is relative to C_BASEADDR assignment. C_BASEADDR is defined in the AXI DMA mpd file and set by Xilinx Platform Studio (XPS).
2. Register 2Ch is available only when DMA is configured in multichannel Mode.

Table 2-10: **AXI DMA Simple DMA Mode Register Address Mapping (C_INCLUDE_SG = 0)**

Address Space Offset[1]	Name	Description
00h	MM2S_DMACR	MM2S DMA Control Register
04h	MM2S_DMASR	MM2S DMA Status Register
08h - 14h	Reserved	N/A
18h	MM2S_SA	MM2S Source Address
1Ch - 24h	Reserved	N/A
28h	MM2S_LENGTH	MM2S Transfer Length (Bytes)
30h	S2MM_DMACR	S2MM DMA Control Register
34h	S2MM_DMASR	S2MM DMA Status Register

Table 2-10: **AXI DMA Simple DMA Mode Register Address Mapping (C_INCLUDE_SG = 0)**

Address Space Offset[1]	Name	Description
38h - 44h	Reserved	N/A
48h	S2MM_DA	S2MM Destination Address
4Ch - 54h	Reserved	N/A
58h	S2MM_LENGTH	S2MM Buffer Length (Bytes)

Notes:

1. Address Space Offset is relative to C_BASEADDR assignment. C_BASEADDR is defined in the AXI DMA mpd file and set by XPS.

Endianess

All registers are in Little Endian format, as shown in Figure 2-2.

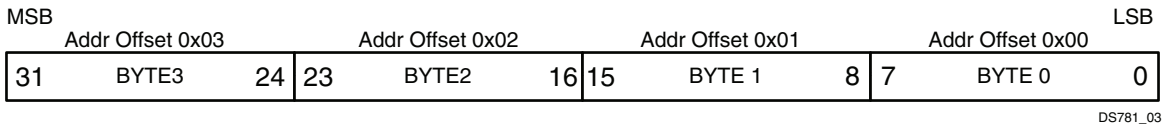

Figure 2-2: **32-bit Little Endian Example**

Memory Map to Stream Register Detail

MM2S_DMACR (MM2S DMA Control Register - Offset 00h) (C_INCLUDE_SG = 1/0)

This register provides control for the Memory Map to Stream DMA Channel.

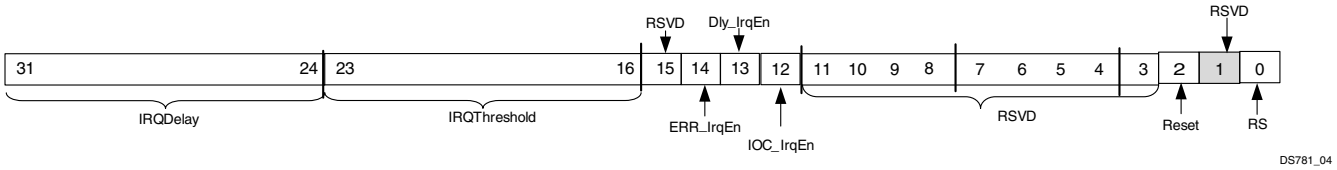

Figure 2-3: **MM2S DMACR Register**

Table 2-11: **MM2S_DMACR Register Details**

Bits	Field Name	Default Value	Access Type	Description
0	RS	0	R/W	Run / Stop control for controlling running and stopping of the DMA channel. • 0 = Stop - DMA stops when current (if any) DMA operations are complete. For Scatter / Gather Mode (C_INCLUDE_SG = 1) pending commands/transfers are flushed or completed. AXI4-Stream outs are potentially terminated early. Descriptors in the update queue are allowed to finish updating to remote memory before engine halt. For Simple DMA Mode (C_INCLUDE_SG = 0) pending commands/transfers are flushed or completed. AXI4-Stream outs are potentially terminated early. The halted bit in the DMA Status Register asserts to 1 when the DMA engine is halted. This bit is cleared by AXI DMA hardware when an error occurs. The CPU can also choose to clear this bit to stop DMA operations. • 1 = Run - Start DMA operations. The halted bit in the DMA Status Register deasserts to 0 when the DMA engine begins operations.
1	Reserved	1	RO	Writing to this bit has no effect, and is always read as 1.
2	Reset	0	RW	Soft reset for resetting the AXI DMA core. Setting this bit to a 1 causes the AXI DMA to be reset. Reset is accomplished gracefully. Pending commands/transfers are flushed or completed. AXI4-Stream outs are potentially terminated early. Setting either MM2S_DMACR.Reset = 1 or S2MM_DMACR.Reset = 1 resets the entire AXI DMA engine. After completion of a soft reset, all registers and bits are in the Reset State. • 0 = Reset NOT in progress - Normal operation. • 1 = Reset in progress.
3	Keyhole	0	RW	Keyhole Read. Setting this bit to 1 causes AXI DMA to initiate MM2S reads in non-incrementing address mode. This bit can be modified when AXI DMA is in idle. When using Key Hole operation the Max Burst Length should not exceed 16.
11 to 4	Reserved	0	RO	Writing to these bits has no effect, and they are always read as zeros.
12	IOC_IrqEn	0	R/W	Interrupt on Complete (IOC) Interrupt Enable. When set to 1, allows DMASR.IOC_Irq to generate an interrupt out for descriptors with the IOC bit set. • 0 = IOC Interrupt disabled • 1 = IOC Interrupt enabled
13	Dly_IrqEn	0	R/W	Interrupt on Delay Timer Interrupt Enable. When set to 1, allows DMASR.Dly_Irq to generate an interrupt out. • 0 = Delay Interrupt disabled • 1 = Delay Interrupt enabled ***Note:*** This bit is ignored when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0)

Table 2-11: **MM2S_DMACR Register Details *(Cont'd)***

Bits	Field Name	Default Value	Access Type	Description
14	Err_IrqEn	0	R/W	Interrupt on Error Interrupt Enable. When set to 1, allows DMASR.Err_Irq to generate an interrupt out. • 0 = Error Interrupt disabled • 1 = Error Interrupt enabled
15	Reserved	0	RO	Writing to this bit has no effect and it is always read as zeros.
23 to 16	IRQThreshold	01h	R/W	Interrupt Threshold. This value is used for setting the interrupt threshold. When IOC interrupt events occur, an internal counter counts down from the Interrupt Threshold setting. When the count reaches zero, an interrupt out is generated by the DMA engine. ***Note:*** The minimum setting for the threshold is 0x01. A write of 0x00 to this register has no effect. ***Note:*** This field is ignored when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0)
31 to 24	IRQDelay	00h	R/W	Interrupt Delay Time Out. This value is used for setting the interrupt timeout value. The interrupt timeout is a mechanism for causing the DMA engine to generate an interrupt after the delay time period has expired. Timer begins counting at the end of a packet and resets with receipt of a new packet or a timeout event occurs. ***Note:*** Setting this value to zero disables the delay timer interrupt. ***Note:*** This field is ignored when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0)

Notes:

1. RO = Read Only. Writing has no effect.
2. R/W = Read and Write Accessible

MM2S_DMASR (MM2S DMA Status Register- Offset 04h) (C_INCLUDE_SG = 1/0)

This register provides the status for the Memory Map to Stream DMA Channel.

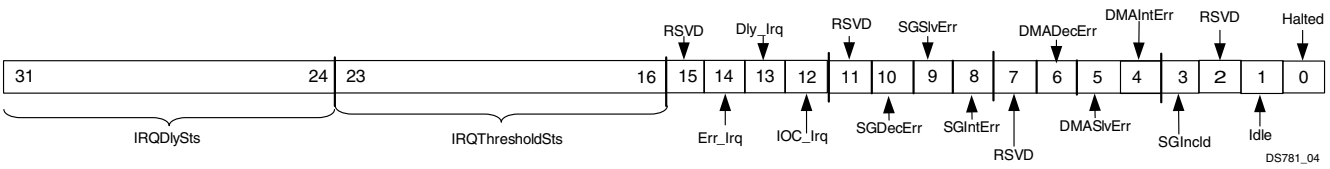

Figure 2-4: **MM2S DMASR Register**

Table 2-12: **MM2S_DMASR Register Details**

Bits	Field Name	Default Value	Access Type	Description
0	Halted	1	RO	DMA Channel Halted. Indicates the run/stop state of the DMA channel. • 0 = DMA channel running. • 1 = DMA channel halted. For Scatter / Gather Mode (C_INCLUDE_SG = 1) this bit gets set when DMACR.RS = 0 and DMA and SG operations have halted. For Simple DMA Mode (C_INCLUDE_SG = 0) this bit gets set when DMACR.RS = 0 and DMA operations have halted. There can be a lag of time between when DMACR.RS = 0 and when DMASR.Halted = 1. ***Note:*** When halted (RS= 0 and Halted = 1), writing to CURDESC_PTR or TAILDESC_PTR pointer registers has no effect on DMA operations when in Scatter Gather Mode (C_INCLUDE_SG = 1). For Simple DMA Mode (C_INCLUDE_SG = 0), writing to the LENGTH register has no effect on DMA operations.
1	Idle	0	RO	DMA Channel Idle. Indicates the state of AXI DMA operations. For Scatter / Gather Mode (C_INCLUDE_SG = 1) when IDLE indicates the SG Engine has reached the tail pointer for the associated channel and all queued descriptors have been processed. Writing to the tail pointer register automatically restarts DMA operations. For Simple DMA Mode (C_INCLUDE_SG = 0) when IDLE indicates the current transfer has completed. • 0 = Not Idle. For Scatter / Gather Mode, SG has not reached tail descriptor pointer and/or DMA operations in progress. For Simple DMA Mode, transfer is not complete. • 1 = Idle. For Scatter / Gather Mode, SG has reached tail descriptor pointer and DMA operation paused. for Simple DMA Mode, DMA transfer has completed and controller is paused. ***Note:*** This bit is 0 when channel is halted (DMASR.Halted=1). This bit is also 0 prior to initial transfer when AXI DMA configured for Simple DMA mode (C_INCLUDE_SG = 0).
2	Reserved	0	RO	Writing to this bit has no effect and it is always read as zero.
3	SGIncld	C_ INCLUDE_ SG	RO	Scatter Gather Engine Included. DMASR.SGIncld = 1 indicates the Scatter Gather engine is included and the AXI DMA is configured for Scatter Gather mode. DMASR.SGIncld = 0 indicates the Scatter Gather engine is excluded and the AXI DMA is configured for Simple DMA mode.

Table 2-12: **MM2S_DMASR Register Details *(Cont'd)***

Bits	Field Name	Default Value	Access Type	Description
4	DMAIntErr	0	RO	DMA Internal Error. Internal error detected by primary AXI DataMover. This situation only happens if the buffer length specified in the fetched descriptor is set to 0. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Internal Errors • 1 = DMA Internal Error detected. DMA Engine halts. ***Note:*** This bit is not used and is fixed at 0 when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
5	DMASlvErr	0	RO	DMA Slave Error. Slave error detected by primary AXI DataMover. This error occurs if the slave read from the Memory Map interface issues a Slave Error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Slave Errors. • 1 = DMA Slave Error detected. DMA Engine halts.
6	DMADecErr	0	RO	DMA Decode Error. Decode error detected by primary AXI DataMover. This error occurs if the address request is to an invalid address (that is, the Descriptor Buffer Address points to an invalid address). This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Decode Errors. • 1 = DMA Decode Error detected. DMA Engine halts.
7	Reserved	0	RO	Writing to this bit has no effect, and it is always read as zeros.
8	SGIntErr	0	RO	Scatter Gather Internal Error. Internal error detected by Scatter Gather AXI DataMover. This error occurs if a descriptor with the Complete bit already set is fetched. This indicates to the SG Engine that the descriptor is a stale descriptor. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No SG Internal Errors. • 1 = SG Internal Error detected. DMA Engine halts. ***Note:*** This bit is not used and is fixed at 0 when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).

Table 2-12: **MM2S_DMASR Register Details *(Cont'd)***

Bits	Field Name	Default Value	Access Type	Description
9	SGSlvErr	0	RO	Scatter Gather Slave Error. Slave error detected by Scatter Gather AXI DataMover. This error occurs if the slave read from on the Memory Map interface issues a Slave error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No SG Slave Errors. • 1 = SG Slave Error detected. DMA Engine halts. ***Note:*** This bit is not used and is fixed at 0 when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
10	SGDecErr	0	RO	Scatter Gather Decode Error. Decode Error detected by the Scatter Gather AXI DataMover. This error occurs if the address request is to an invalid address (that is, CURDESC_PTR and/or NXTDESC_PTR points to an invalid address). This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No SG Decode Errors. • 1 = SG Decode Error detected. DMA Engine halts. ***Note:*** This bit is not used and is fixed at 0 when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
11	Reserved	0	RO	Writing to this bit has no effect, and it is always read as zeros.
12	IOC_Irq	0	R/WC	Interrupt on Complete. When set to 1 for Scatter / Gather Mode (C_INCLUDE_SG = 1), indicates an interrupt event was generated on completion of a descriptor. This occurs for descriptors with the End Of Frame (EOF) bit set. When set to 1 for Simple DMA Mode (C_INCLUDE_SG = 0), indicates an interrupt event was generated on completion of a transfer. If enabled (IOC_IrqEn = 1) and if the interrupt threshold has been met, causes an interrupt out to be generated from the AXI DMA. • 0 = No IOC Interrupt. • 1 = IOC Interrupt detected.
13	Dly_Irq	0	R/WC	Interrupt on Delay. When set to 1, indicates an interrupt event was generated on delay timer timeout. If enabled (Dly_IrqEn = 1), an interrupt out is generated from the AXI DMA. • 0 = No Delay Interrupt. • 1 = Delay Interrupt detected. ***Note:*** This bit is not used and is fixed at 0 when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
14	Err_Irq	0	R/WC	Interrupt on Error. When set to 1, indicates an interrupt event was generated on error. If enabled (Err_IrqEn = 1), an interrupt out is generated from the AXI DMA. • 0 = No error Interrupt. • 1 = Error interrupt detected.

Table 2-12: **MM2S_DMASR Register Details *(Cont'd)***

Bits	Field Name	Default Value	Access Type	Description
15	Reserved	0	RO	Always read as zero.
23 to 16	IRQThresholdSts	01h	RO	Interrupt Threshold Status. Indicates current interrupt threshold value. ***Note:*** This field is not used and is fixed to zeros when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
31 to 24	IRQDelaySts	00h	RO	Interrupt Delay Time Status. Indicates current interrupt delay time value. ***Note:*** This field is not used and is fixed to zeros when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).

Notes:

1. RO = Read Only. Writing has no effect.
2. R/WC = Read / Write to Clear. A CPU write of 1 clears the associated bit to 0.

MM2S_CURDESC (MM2S DMA Current Descriptor Pointer Register- Offset 08h) (C_INCLUDE_SG = 1)

This register provides the Current Descriptor Pointer for the Memory Map to Stream DMA Scatter Gather Descriptor Management.

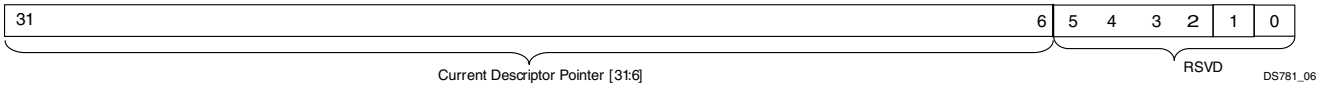

Figure 2-5: **MM2S CURDESC Register**

Table 2-13: **MM2S_CURDESC Register Details**

Bits	Field Name	Default Value	Access Type	Description
5 to 0 (Offset 0x38)	Reserved	0	RO	Writing to these bits has no effect and they are always read as zeros.
31 to 6	Current Descriptor Pointer	zeros	R/W (RO)(1)	Indicates the pointer of the current descriptor being worked on. This register must contain a pointer to a valid descriptor prior to writing the TAILDESC_PTR register. Otherwise, undefined results occur. When DMACR.RS is 1, CURDESC_PTR becomes Read Only (RO) and is used to fetch the first descriptor. When the DMA Engine is running (DMACR.RS=1), CURDESC_PTR registers are updated by AXI DMA to indicate the current descriptor being worked on. On error detection, CURDESC_PTR is updated to reflect the descriptor associated with the detected error. ***Note:*** The register can only be written to by the CPU when the DMA Engine is Halted (DMACR.RS=0 and DMASR.Halted =1). At all other times, this register is Read Only (RO). Descriptors must be 16 word aligned, that is, 0x00, 0x40, 0x80 and others. Any other alignment has undefined results.

Notes:

1. RO = Read Only. Writing has no effect when channel is running.
2. R/W = Read and Write accessible.

MM2S_TAILDESC (MM2S DMA Tail Descriptor Pointer Register- Offset 10h) (C_INCLUDE_SG = 1)

This register provides the Tail Descriptor Pointer for the Memory Map to Stream DMA Scatter Gather Descriptor Management.

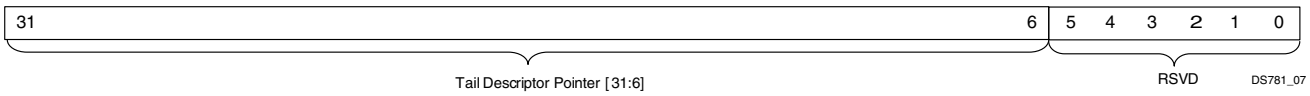

Figure 2-6: **MM2S_TAILDESC Register**

Table 2-14: **MM2S_TAILDESC Register Details**

Bits	Field Name	Default Value	Access Type	Description
5 to 0	Reserved	0	RO	Writing to these bits has no effect, and they are always read as zeros.
31 to 6	Tail Descriptor Pointer	zeros	R/W	Indicates the pause pointer in a descriptor chain. The AXI DMA SG Engine pauses descriptor fetching after completing operations on the descriptor whose current descriptor pointer matches the tail descriptor pointer. When AXI DMA Channel is not halted (DMASR.Halted = 0), a write by the CPU to the TAILDESC_PTR register causes the AXI DMA SG Engine to start fetching descriptors or restart if it was idle (DMASR.Idle = 1). If it was not idle, writing TAILDESC_PTR has no effect except to reposition the pause point. If the AXI DMA Channel is halted (DMASR.Halted = 1 and DMACR.RS = 0), a write by the CPU to the TAILDESC_PTR register has no effect except to reposition the pause point. ***Note:*** The software must not move the tail pointer to a location that has not been updated. The software processes and reallocates all completed descriptors (Cmplted = 1), clears the completed bits and then moves the tail pointer. The software must move the pointer to the last descriptor it updated. Descriptors must be 16-word aligned, that is, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results.

Notes:

1. RO = Read Only. Writing has no effect.
2. R/W = Read and Write accessible.

MM2S_SA (MM2S DMA Source Address Register- Offset 18h) (C_INCLUDE_SG = 0)

This register provides the Source Address for reading system memory for the Memory Map to Stream DMA transfer.

Figure 2-7: **MM2S_SA Register**

Table 2-15: **MM2S_SA Register Details**

Bits	Field Name	Default Value	Access Type	Description
31 to 0	Source Address	zeros	R/W	Indicates the source address AXI DMA reads from to transfer data to AXI4-Stream on the MM2S Channel. ***Note:*** If Data Realignment Engine is included (C_INCLUDE_MM2S_DRE = 1), the Source Address can be at any byte offset. If Data Realignment Engine is not included (C_INCLUDE_MM2S_DRE = 0), the Source Address must be S2MM stream data width aligned.

MM2S_LENGTH (MM2S DMA Transfer Length Register- Offset 28h) (C_INCLUDE_SG = 0)

This register provides the number bytes to read from system memory and transfer to MM2S AXI4-Stream.

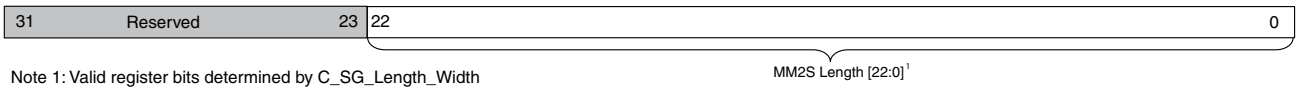

Figure 2-8: **MM2S_LENGTH Register**

Table 2-16: **MM2S_LENGTH Register Details**

Bits	Field Name	Default Value	Access Type	Description
22[a] to 0	Length	zeros	R/W	Indicates the number of bytes to transfer for the MM2S channel. Writing a non-zero value to this register starts the MM2S transfer.
31 to 23	Reserved	0	RO	Writing to these bits has no effect and they are always read as zeros.

a. Width of Length field determined by C_SG_LENGTH_WIDTH parameter. Minimum width is 8 bits (7 to 0) and maximum width is 23 bits (22 to 0).

SG_CTL (Scatter/Gather User and Cache Control Register- Offset 2Ch)(C_INCLUDE_SG=1)

This register is available only when DMA is configured in multichannel Mode.

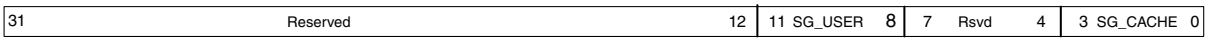

Figure 2-9: **SG_CTL Register**

Table 2-17: **SG_CTL Register Details**

Bits	Field Name	Default Value	Access Type	Description
3 to 0	SG_CACHE	0011b	R/W	Scatter/Gather Cache Control. Values written in this register will reflect on m_axi_sg_arcache.
7 to 4	Reserved	0	RO	Writing to these bits has no effect and they are always read as zeros.
11 to 8	SG_USER	0	R/W	Scatter/Gather User Control. Values written in this register will reflect on m_axi_sg_aruser.

Stream to Memory Map Register Detail

S2MM_DMACR (S2MM DMA Control Register - Offset 30h) (C_INCLUDE_SG = 1/0)

This register provides control for the Stream to Memory Map DMA Channel.

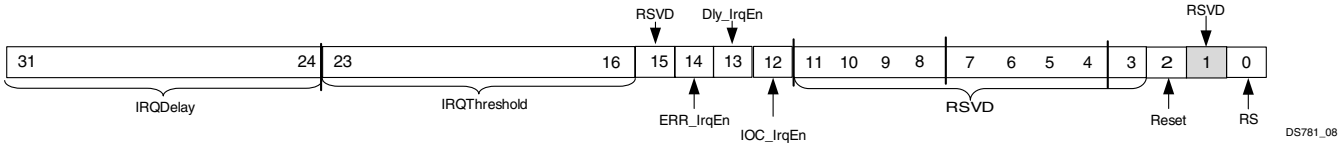

Figure 2-10: **S2MM DMACR Register**

Table 2-18: **S2MM_DMACR Register Details**

Bits	Field Name	Default Value	Access Type	Description
0	RS	0	R/W	Run / Stop control for controlling running and stopping of the DMA channel. • 0 = Stop - DMA stops when current (if any) DMA operations are complete. For Scatter / Gather Mode (C_INCLUDE_SG = 1) pending commands/transfers are flushed or completed. AXI4-Streams are potentially terminated early. Descriptors in the update queue are allowed to finish updating to remote memory before engine halt. For Simple DMA Mode (C_INCLUDE_SG = 0) pending commands/transfers are flushed or completed. AXI4-Streams are potentially terminated early. Data integrity on S2MM AXI4 cannot be guaranteed. The halted bit in the DMA Status Register asserts to 1 when the DMA engine is halted. This bit is cleared by AXI DMA hardware when an error occurs. The CPU can also choose to clear this bit to stop DMA operations. • 1 = Run - Start DMA operations. The halted bit in the DMA Status Register deasserts to 0 when the DMA engine begins operations.
1	Reserved	1	RO	Writing to this bit has no effect, and is always read as 1.
2	Reset	0	R/W	Soft reset for resetting the AXI DMA core. Setting this bit to a 1 causes the AXI DMA to be reset. Reset is accomplished gracefully. Pending commands/transfers are flushed or completed. AXI4-Stream outs are terminated early, if necessary with associated TLAST. Setting either MM2S_DMACR.Reset = 1 or S2MM_DMACR.Reset = 1 resets the entire AXI DMA engine. After completion of a soft reset, all registers and bits are in the Reset State. • 0 = Reset not in progress. Normal operation. • 1 = Reset in progress.
3	Keyhole	0	R/W	Keyhole Write. Setting this bit to 1 causes AXI DMA to initiate S2MM writes in non-incrementing address mode. This bit can be modified when AXI DMA is in idle. When enabling Key hole operation the max burst length cannot be more than 16.
11 to 4	Reserved	0	RO	Writing to these bits has no effect and they are always read as zeros.
12	IOC_IRqEn	0	R/W	Interrupt on Complete Interrupt Enable. When set to 1, allows Interrupt On Complete events to generate an interrupt out for descriptors with the IOC bit set. • 0 = IOC Interrupt disabled. • 1 = IOC Interrupt enabled.

Table 2-18: **S2MM_DMACR Register Details *(Cont'd)***

Bits	Field Name	Default Value	Access Type	Description
13	Dly_IrqEn	0	R/W	Interrupt on Delay Timer Interrupt Enable. When set to 1, allows error events to generate an interrupt out. • 0 = Delay Interrupt disabled. • 1 = Delay Interrupt enabled. ***Note:*** This bit is ignored when the AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
14	Err_IrqEn	0	R/W	Interrupt on Error Interrupt Enable. When set to 1, allows error events to generate an interrupt out. • 0 = Error Interrupt disabled. • 1 = Error Interrupt enabled.
15	Reserved	0	RO	Writing to this bit has no effect, and it is always read as zeros.
23 to 16	IRQThreshold	01h	R/W	Interrupt Threshold. This value is used for setting the interrupt threshold. When IOC interrupt events occur, an internal counter counts down from the Interrupt Threshold setting. When the count reaches zero, an interrupt out is generated by the DMA engine. ***Note:*** The minimum setting for the threshold is 0x01. A write of 0x00 to this register has no effect. ***Note:*** This field is ignored when the AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0)
31 to 24	IRQDelay	00h	R/W	Interrupt Delay Time Out. This value is used for setting the interrupt timeout value. The interrupt timeout is a mechanism for causing the DMA engine to generate an interrupt after the delay time period has expired. The timer begins counting at the end of a packet and resets with the receipt of a new packet or a timeout event occurs. ***Note:*** Setting this value to zero disables the delay timer interrupt. ***Note:*** This field is ignored when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0)

Notes:

1. RO = Read Only. Writing has no effect.
2. R/W = Read and Write Accessible.

S2MM_DMASR (S2MM DMA Status Register- Offset 34h) (C_INCLUDE_SG = 1/0)

This register provides the status for the Stream to Memory Map DMA Channel

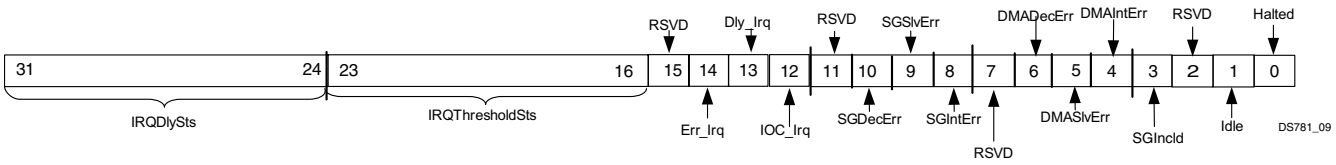

Figure 2-11: **S2MM DMASR Register**

Table 2-19: **S2MM_DMASR Register Details**

Bits	Field Name	Default Value	Access Type	Description
0	Halted	1	RO	DMA Channel Halted. Indicates the run/stop state of the DMA channel. • 0 = DMA channel running. • 1 = DMA channel halted. For Scatter / Gather Mode (C_INCLUDE_SG = 1) this bit gets set when DMACR.RS = 0 and DMA and SG operations have halted. For Simple DMA Mode (C_INCLUDE_SG = 0) this bit gets set when DMACR.RS = 0 and DMA operations have halted. There can be a lag of time between when DMACR.RS = 0 and when DMASR.Halted = 1. ***Note:*** When halted (RS= 0 and Halted = 1), writing to CURDESC_PTR or TAILDESC_PTR pointer registers has no effect on DMA operations when in Scatter Gather Mode (C_INCLUDE_SG = 1). For Simple DMA Mode (C_INCLUDE_SG = 0), writing to the LENGTH register has no effect on DMA operations.
1	Idle	0	RO	DMA Channel Idle. Indicates the state of AXI DMA operations. For Scatter / Gather Mode (C_INCLUDE_SG = 1) when IDLE indicates the SG Engine has reached the tail pointer for the associated channel and all queued descriptors have been processed. Writing to the tail pointer register automatically restarts DMA operations. For Simple DMA Mode (C_INCLUDE_SG = 0) when IDLE indicates the current transfer has completed. • 0 = Not Idle. For Scatter / Gather Mode, SG has not reached tail descriptor pointer and/or DMA operations in progress. For Simple DMA Mode, transfer is not complete. • 1 = Idle. For Scatter / Gather Mode, SG has reached tail descriptor pointer and DMA operation paused. For Simple DMA Mode, DMA transfer has completed and controller is paused. ***Note:*** This bit is 0 when channel is halted (DMASR.Halted=1). This bit is also 0 prior to initial transfer when AXI DMA is configured for Simple DMA mode (C_INCLUDE_SG = 0).
2	Reserved	0	RO	Writing to this bit has no effect and it is always read as zero.
3	SGIncld	C_ INCLUDE_ SG	RO	Scatter Gather Engine Included. DMASR.SGIncld = 1 indicates the Scatter Gather engine is included and the AXI DMA is configured for Scatter Gather mode. DMASR.SGIncld = 0 indicates the Scatter Gather engine is excluded and the AXI DMA is configured for Simple DMA mode.

Table 2-19: **S2MM_DMASR Register Details *(Cont'd)***

Bits	Field Name	Default Value	Access Type	Description
4	DMAIntErr	0	RO	DMA Internal Error. Internal Error detected by primary AXI DataMover. This error can occur if a 0 length bytes to transfer is fed to the AXI DataMover. This happens if the buffer length specified in the fetched descriptor is set to 0. Also, when in Scatter Gather Mode, C_INCLUDE_SG = 1 and using the status app length field, C_SG_INCLUDE_STSCNTRL_STRM = 1 and C_SG_USE_STSAPP_LEGNTH = 1, this error occurs when the Status AXI4-Stream packet's RxLength field does not match the S2MM packet being received by the S_AXIS_S2MM interface. When, C_INCLUDE_SG=0, this error is flagged if there is any error during Memory write or if the incoming packet is bigger than what is specified in the DMA length register. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Internal Errors. • 1 = DMA Internal Error detected. DMA Engine halts.
5	DMASlvErr	0	RO	DMA Slave Error. Slave Error detected by primary AXI DataMover. This error occurs if the slave read from the Memory Map interface issues a Slave Error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0 and when the engine has completely shut down the DMASR.Halted bit is set to 1. • 0 = No DMA Slave Errors. • 1 = DMA Slave Error detected. DMA Engine halts.
6	DMADecErr	0	RO	DMA Decode Error. Decode error detected by primary AXI DataMover. This error occurs if the address request is to an invalid address (that is, the Descriptor Buffer Address points to an invalid address). This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Decode Errors. • 1 = DMA Decode Error detected. DMA Engine halts.
7	Reserved	0	RO	Writing to this bit has no effect and it is always read as zeros.

Table 2-19: **S2MM_DMASR Register Details *(Cont'd)***

Bits	Field Name	Default Value	Access Type	Description
8	SGIntErr	0	RO	Scatter Gather Internal Error. Internal Error detected by Scatter Gather AXI DataMover. This error occurs if a descriptor with the Complete bit already set is fetched. This indicates to the SG Engine that the descriptor is a tail descriptor. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No SG Internal Errors. • 1 = SG Internal Error detected. DMA Engine halts. This error cannot be logged into the descriptor. ***Note:*** This bit is not used and is fixed at 0 when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
9	SGSlvErr	0	RO	Scatter Gather Slave Error. Slave error detected by Scatter Gather AXI DataMover. This error occurs if the slave read from on the Memory Map interface issues a Slave Error. This error condition causes the AXI DMA to gracefully halt. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No SG Slave Errors. • 1 = SG Slave Error detected. DMA Engine halts. This error cannot be logged into the descriptor. ***Note:*** This bit is not used and is fixed at 0 when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
10	SGDecErr	0	RO	Scatter Gather Decode Error. Decode Error detected by the Scatter Gather AXI DataMover. This error occurs if the address request is to an invalid address (that is, CURDESC_PTR and/or NXTDESC_PTR points to an invalid address). This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0 and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No SG Decode Errors. • 1 = SG Decode Error detected. DMA Engine halts. This error cannot be logged into the descriptor. ***Note:*** This bit is not used and is fixed at 0 when AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
11	Reserved	0	RO	Writing to this bit has no effect and it is always read as zeros.

Table 2-19: **S2MM_DMASR Register Details *(Cont'd)***

Bits	Field Name	Default Value	Access Type	Description
12	IOC_Irq	0	R/WC	Interrupt on Complete. When set to 1 for Scatter / Gather Mode (C_INCLUDE_SG = 1) indicates an interrupt event was generated on completion of a descriptor. This occurs for descriptors with the EOF bit set. When set to 1 for Simple DMA Mode (C_INCLUDE_SG = 0) indicates an interrupt event was generate on completion of a transfer. If enabled (IOC_IrqEn = 1) and if the interrupt threshold has been met, causes an interrupt out to be generated from the AXI DMA. • 0 = No IOC Interrupt. • 1 = IOC Interrupt detected.
13	Dly_Irq	0	R/WC	Interrupt on Delay. When set to 1, indicates an interrupt event was generated on delay timer timeout. If enabled (Dly_IrqEn = 1), an interrupt out is generated from the AXI DMA. • 0 = No Delay Interrupt. • 1 = Delay Interrupt detected. ***Note:*** This bit is not used and is fixed at 0 when the AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
14	Err_Irq	0	R/WC	Interrupt on Error. When set to 1, indicates an interrupt event was generated on error. If enabled (Err_IrqEn = 1), an interrupt out is generated from the AXI DMA. • 0 = No Error Interrupt. • 1 = Error Interrupt detected.
15	Reserved	0	RO	Writing to this bit has no effect and it is always read as zeros.
23 to 16	IRQThresholdSts	01h	RO	Interrupt Threshold Status. Indicates current interrupt threshold value. ***Note:*** This field is not used and is fixed to zeros when the AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).
31 to 24	IRQDelaySts	00h	RO	Interrupt delay time Status. Indicates current interrupt delay time value. ***Note:*** This field is not used and is fixed to zeros when the AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0).

Notes:

1. RO = Read Only. Writing has no effect.
2. R/WC = Read / Write to Clear. A CPU write of 1 clears the associated bit to 0.

S2MM_CURDESC (S2MM DMA Current Descriptor Pointer Register- Offset 38h) (C_INCLUDE_SG = 1)

This register provides the Current Descriptor Pointer for the Stream to Memory Map DMA Scatter Gather Descriptor Management.

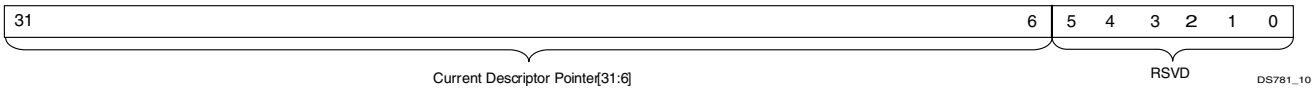

Figure 2-12: **S2MM CURDESC Register**

Table 2-20: **S2MM_CURDESC Register Details**

Bits	Field Name	Default Value	Access Type	Description
5 to 0 (Offset 0x38)	Reserved	0	RO	Writing to these bits has no effect and they are always read as zeros.
31 to 6	Current Descriptor Pointer	zeros	R/W (RO)(2)	Indicates the pointer of the current Buffer Descriptor being worked on. This register must contain a pointer to a valid descriptor prior to writing the TAILDESC_PTR register. Otherwise, undefined results occur. When DMACR.RS is 1, CURDESC_PTR becomes Read Only (RO) and is used to fetch the first descriptor. When the DMA Engine is running (DMACR.RS=1), CURDESC_PTR registers are updated by AXI DMA to indicate the current descriptor being worked on. On error detection, CURDESC_PTR is updated to reflect the descriptor associated with the detected error. ***Note:*** The register can only be written to by the CPU when the DMA Engine is halted (DMACR.RS=0 and DMASR.Halted =1). At all other times, this register is Read Only (RO). Buffer Descriptors must be 16-word aligned, that is, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results.

Notes:

1. RO = Read Only. Writing has no effect when engine is running.
2. R/W = Read and Write Accessible.

S2MM_TAILDESC (S2MM DMA Tail Descriptor Pointer Register- Offset 40h) (C_INCLUDE_SG = 1)

This register provides the Tail Descriptor Pointer for the Stream to Memory Map DMA Scatter Gather Descriptor Management.

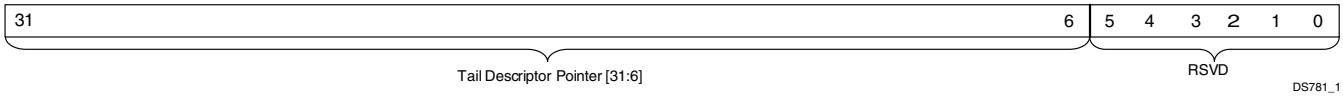

Figure 2-13: **S2MM TAILDESC Register**

Table 2-21: **S2MM_TAILDESC Register Details**

Bits	Field Name	Default Value	Access Type	Description
5 to 0	Reserved	0	RO	Writing to these bits has no effect and they are always read as zeros.
31 to 6	Tail Descriptor Pointer	zeros	R/W	Indicates the pause pointer in a descriptor chain. The AXI DMA SG Engine pauses descriptor fetching after completing operations on the descriptor whose current descriptor pointer matches the tail descriptor pointer. When AXI DMA Channel is not halted (DMASR.Halted = 0), a write by the CPU to the TAILDESC_PTR register causes the AXI DMA SG Engine to start fetching descriptors or restart if it was idle (DMASR.Idle = 1). If it was not idle, then writing TAILDESC_PTR has no effect except to reposition the pause point. If the AXI DMA Channel DMACR.RS bit is set to 0 (DMASR.Halted = 1 and DMACR.RS = 0), a write by the CPU to the TAILDESC_PTR register has no effect except to reposition the pause point. ***Note:*** The software must not move the Tail Pointer to a location that has not been updated. The software processes and reallocates all completed descriptors (Cmplted = 1), clears the completed bits and then moves the tail pointer. The software must move the pointer to the last descriptor it updated. Descriptors must be 16-word aligned, that is, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results.

Notes:
1. RO = Read Only. Writing has no effect.
2. R/W = Read and Write Accessible

S2MM_DA (S2MM DMA Destination Address Register- Offset 48h) (C_INCLUDE_SG = 0)

This register provides the Destination Address for writing to system memory for the Stream to Memory Map to DMA transfer.

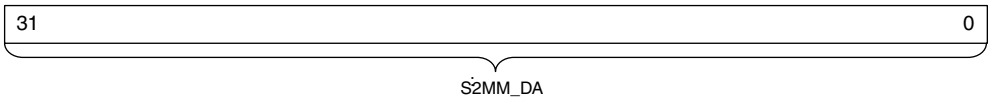

Figure 2-14: **S2MM_DA Register**

Table 2-22: **S2MM_DA Register Details**

Bits	Field Name	Default Value	Access Type	Description
31 to 0	Destination Address	zeros	R/W	Indicates the source address the AXI DMA reads from to transfer data to AXI4-Stream on S2MM Channel. ***Note:*** If Data Realignment Engine is included (C_INCLUDE_S2MM_DRE = 1), the Destination Address can be at any byte offset. If Data Realignment Engine is not included (C_INCLUDE_S2MM_DRE = 0), the Destination Address must be S2MM stream data width aligned.

Notes:

1. R/W = Read and Write accessible.

S2MM_LENGTH (S2MM DMA Buffer Length Register- Offset 58h) (C_INCLUDE_SG = 0)

This register provides the length in bytes of the buffer to write data from the Stream to Memory map DMA transfer.

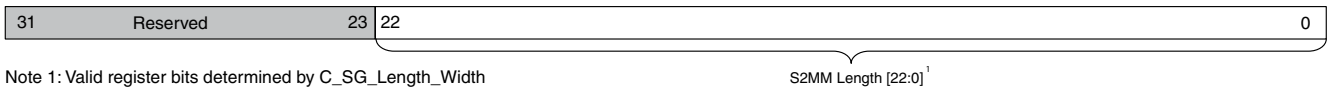

Figure 2-15: **S2MM_LENGTH Register**

Table 2-23: **S2MM_LENGTH Register Details**

Bits	Field Name	Default Value	Access Type	Description
22[2] to 0	Length	zeros	R/W	Indicates the length in bytes of the S2MM buffer available to write receive data from the S2MM channel. Writing a non-zero value to this register enables S2MM channel to receive packet data. At the completion of the S2MM transfer, the number of actual bytes written on S2MM AXI4 interface is updated to the S2MM_LENGTH register. ***Note:*** This value must be greater than or equal to the largest expected packet to be received on S2MM AXI4-Stream. Values smaller than the received packet result in undefined behavior.
31 to 23	Reserved	0	RO	Writing to these bits has no effect and they are always read as zeros.

Notes:

1. R/W = Read and Write accessible.
2. Width of Length field determined by C_SG_LENGTH_WIDTH parameter. Minimum width is 8 bits (7 to 0) and maximum width is 23 bits (22 to 0).

Designing with the Core

This chapter includes guidelines and additional information to make designing with the core easier.

Clocking

There are four clock inputs:

- `m_axi_mm2s_aclk` for MM2S interface
- `m_axi_s2mm_aclk` for S2MM interface
- `s_axi_lite_aclk` for AXI4-Lite control interface
- `m_axi_sg_clk` for Scatter Gather Interface

AXI DMA provides two clocking modes of operation: asynchronous and synchronous. Setting C_PRMRY_IS_ACLK_ASYNC = 1 enables this mode and creates four clock domains. This allows high performance users to run the primary datapaths at a higher clock rate than the DMA control (for example, AXI4-Lite interface, SG Engine, DMA Controller) helping in FPGA placement and timing. This parameter is set automatically by the EDK tool suite based on the clock sources feeding AXI DMA.

In synchronous mode (C_PRMRY_IS_ACLK_ASYNC = 0), all logic runs in a single clock domain. `s_axi_lite_aclk, m_axi_sg_aclk, m_axi_mm2s_aclk and m_axi_s2mm_aclk` must be tied to the same source. In asynchronous mode (C_PRMRY_IS_ACLK_ASYNC = 1) clocks can be run asynchronously, however `s_axi_lite_aclk` must be less than or equal to `m_axi_sg_aclk` and `m_axi_sg_aclk` must be less than or equal to the slower of `m_axi_mm2s_aclk` or `m_axi_s2mm_aclk`.

If AXI DMA is configured for Simple DMA Mode (C_INCLUDE_SG = 0) then the Scatter Gather clock input, `m_axi_sg_aclk`, is ignored and `s_axi_lite_aclk` is used to clock the DMA controllers, register block, and control interface.

The relationship of which signals and signal sets are clocked by what clock in asynchronous mode is shown in Table 3-1.

Table 3-1: **Asynchronous Mode Clock Distribution (C_PRMRY_IS_ACLK_ASYNC=1)**

Clock Source	I/O Ports (C_INCLUDE_SG = 1)	I/O Ports (C_INCLUDE_SG = 0)
s_axi_lite_aclk	All s_axi_lite_* Signals mm2s_introut s2mm_introut axi_resetn	All s_axi_lite_* Signals mm2s_introut s2mm_introut axi_resetn
m_axi_sg_aclk	All m_axi_sg_* Signals	N/A
m_axi_mm2s_aclk	All m_axi_mm2s_* Signals All m_axis_mm2s_* Signals mm2s_prmry_reset_out_n mm2s_cntrl_reset_out_n	All m_axi_mm2s_* Signals All m_axis_mm2s_* Signals mm2s_prmry_reset_out_n
m_axi_s2mm_aclk	All m_axi_s2mm_* Signals All s_axis_s2mm_* Signals s2mm_prmry_reset_out_n s2mm_sts_reset_out_n	All m_axi_s2mm_* Signals All s_axis_s2mm_* Signals s2mm_prmry_reset_out_n

Resets

The `axi_resetn` signal needs to be asserted a minimum of eight of the slowest clock's clock cycles and needs to be synchronized to `s_axi_lite_aclk`.

AXI DMA Simple DMA Operation

Simple DMA mode (C_INCLUDE_SG = 0) provides a configuration for doing simple DMA transfers on MM2S and S2MM channels that requires less FPGA resource utilization. Transfers are initiated by accessing the DMACR, the Source or Destination Address and the Length registers. When the transfer is completed, a DMASR.IOC_Irq assert for the associated channel and if enabled generates an interrupt out.

A DMA operation for the MM2S channel is set up and started by the following sequence:

1. Start the MM2S channel running by setting the run/stop bit to 1 (MM2S_DMACR.RS = 1). The halted bit (DMASR.Halted) should deassert indicating the MM2S channel is running.

2. If desired, enable interrupts by writing a 1 to MM2S_DMACR.IOC_IrqEn and MM2S_DMACR.Err_IrqEn. The delay interrupt, delay count, and threshold count are not used when the AXI DMA is configured for Simple DMA mode.
3. Write a valid source address to the MM2S_SA register. If the AXI DMA is not configured for Data Re-Alignment (C_INCLUDE_MM2S_DRE = 0 or C_M_AXIS_MM2S_TDATA_WIDTH > 64), then a valid address must be aligned or undefined results occur. What is considered aligned or unaligned is based on the stream data width C_M_AXIS_MM2S_TDATA_WIDTH.

   For example, if C_M_AXIS_MM2S_TDATA_WIDTH = 32, data is aligned if it is located at word offsets (32-bit offset), that is 0x0, 0x4, 0x8, 0xC, and so forth. If C_INCLUDE_MM2S_DRE = 1 and C_M_AXIS_MM2S_TDATA_WIDTH < 128, then Source Addresses can be of any byte offset.
4. Write the number of bytes to transfer in the MM2S_LENGTH register. A value of zero written has no effect. A non-zero value causes the MM2S_LENGTH number of bytes to be read on the MM2S AXI4 interface and transmitted out of the MM2S AXI4-Stream interface. The MM2S_LENGTH register must be written last. All other MM2S registers can be written in any order.

A DMA operation for the S2MM channel is set up and started by the following sequence:

1. Start the S2MM channel running by setting the run/stop bit to 1 (S2MM_DMACR.RS = 1). The halted bit (DMASR.Halted) should deassert indicating the S2MM channel is running.
2. If desired, enable interrupts by writing a 1 to S2MM_DMACR.IOC_IrqEn and S2MM_DMACR.Err_IrqEn. The delay interrupt, delay count, and threshold count are not used when the AXI DMA is configured for Simple DMA mode.
3. Write a valid destination address to the S2MM_DA register. If the AXI DMA is not configured for Data Re-Alignment (C_INCLUDE_S2MM_DRE = 0 or C_S_AXIS_S2MM_TDATA_WIDTH > 64) then a valid address must be aligned or undefined results occur. What is considered aligned or unaligned is based on the stream data width C_S_AXIS_S2MM_TDATA_WIDTH.

   For example, if C_S_AXIS_S2MM_TDATA_WIDTH = 32, data is aligned if it is located at word offsets (32-bit offset), that is, 0x0, 0x4, 0x8, 0xC, and so forth. If C_INCLUDE_S2MM_DRE = 1 and C_S_AXIS_S2MM_TDATA_WIDTH < 128 then Destination Addresses can be of any byte offset.
4. Write the length in bytes of the receive buffer in the S2MM_LENGTH register. A value of zero written has no effect. A non-zero value causes a write on the S2MM AXI4 interface of the number of bytes received on the S2MM AXI4-Stream interface. A value greater than or equal to the largest received packet must be written to S2MM_LENGTH. A receive buffer length value written that is less than the number of bytes received produces undefined results. The S2MM_LENGTH register must be written last. All other S2MM register can be written in any order.

Scatter Gather Descriptor (C_INCLUDE_SG = 1)

This section defines the fields of the S2MM (Receive) and MM2S (Transmit) Scatter Gather Descriptors for when the AXI DMA is configured for Scatter / Gather Mode. The descriptor is made up of eight 32-bit base words and 0 or 5 User Application words. The descriptor has future support for 64-bit addresses and support for User Application data. Multiple descriptors per packet are supported through the Start of Frame and End of Frame flags. Completed status and Interrupt on Complete are also included. The Buffer Length can describe up to 8 MB of data buffer per descriptor. Two descriptor chains are required for the two data transfer direction, MM2S and S2MM.

Table 3-2: **Descriptor Fields (Non-Multi Channel Mode)**

Address Space Offset(1)	Name	Description
00h	NXTDESC	Next Descriptor Pointer
04h	RESERVED	N/A
08h	BUFFER_ADDRESS	Buffer Address
0Ch	RESERVED	N/A
10h	RESERVED	N/A
14h	RESERVED	N/A
18h	CONTROL	Control
1Ch	STATUS	Status
20h	APP0	User Application Field 0 (2)
24h	APP1	User Application Field 1
28h	APP2	User Application Field 2
2Ch	APP3	User Application Field 3
30h	APP4	User Application Field 4

Notes:

1. Address Space Offset is relative to 16 - 32-bit word alignment in system memory, that is, 0x00, 0x40, 0x80 and so forth.
2. User Application fields (APP0, APP1, APP2, APP3, and APP4) are only used when the Control / Status Streams are included, C_SG_INCLUDE_STSCNTRL_STRM = 1. When the Control/ Status Streams are not included (C_SG_INCLUDED_STSCNTRL_STRM = 0), the User Application fields are not fetched or updated by the Scatter Gather Engine.

MM2S_NXTDESC (MM2S Next Descriptor Pointer)

This value provides the pointer to the next descriptor in the descriptor chain.

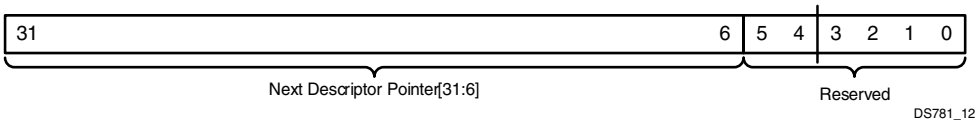

Figure 3-1: **MM2S_NXTDESC**

Table 3-3: **MM2S_NXTDESC Details**

Bits	Field Name	Description
5 to 0	Reserved	These bits are reserved and should be set to zero.
31 to 6	Next Descriptor Pointer	Indicates the lower order pointer pointing to the first word of the next descriptor. ***Note:*** Descriptors must be 16-word aligned, that is, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results.

MM2S_BUFFER_ADDRESS (MM2S Buffer Address)

This value provides the pointer to the buffer of data to transfer from system memory to stream.

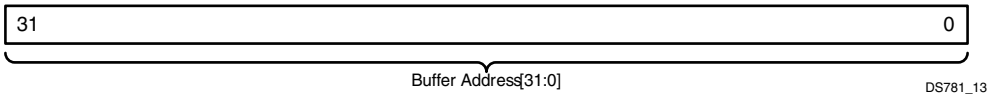

Figure 3-2: **MM2S Buffer Address**

Table 3-4: **MM2S_BUFFER_ADDRESS Details**

Bits	Field Name	Description
31 to 0	Buffer Address	Provides the location of the data to transfer from Memory Map to Stream. ***Note:*** If Data Realignment Engine is included (C_INCLUDE_MM2S_DRE = 1), the Buffer Address can be at any byte offset, but data within a buffer must be contiguous. If the Data Realignment Engine is not included (C_INCLUDE_MM2S_DRE = 0), the Buffer Address must be MM2S stream data-width aligned.

MM2S_CONTROL (MM2S Control)

This value provides control for MM2S transfers from memory map to stream.

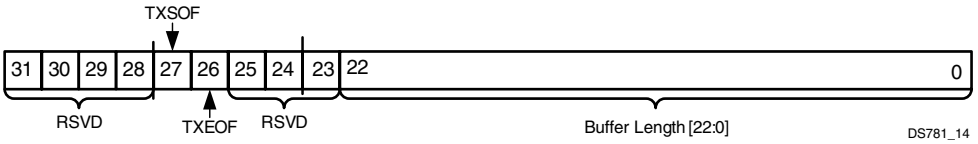

Figure 3-3: **MM2S_CONTROL**

Table 3-5: **MM2S_CONTROL Details**

Bits	Field Name	Description
22 to 0	Buffer Length	Indicates the size in bytes of the transfer buffer. This value indicates the amount of bytes to transmit out on the MM2S stream. The usable width of buffer length is specified by C_SG_LENGTH_WIDTH. A maximum of 8 MB of transfer can be described by this field. ***Note:*** Setting C_SG_LENGTH_WIDTH smaller than 23 reduces FPGA resource utilization.
5 to 23	Reserved	These bits are reserved and should be set to zero.
26	Transmit End Of Frame (TXEOF)	End of Frame. Flag indicating the last buffer to be processed. This flag is set by the CPU to indicate to AXI DMA that this descriptor describes the end of the packet. The buffer associated with this descriptor is transmitted last. • 0 = Not end of frame. • 1 = End of frame. ***Note:*** For proper operation, there must be an SOF descriptor (TXSOF=1) and an EOF descriptor (TXEOF=1) per packet. It is valid to have a single descriptor describe an entire packet that is a descriptor with both TXSOF=1 and TXEOF=1.
27	TXSOF	Start of Frame. Flag indicating the first buffer to be processed. This flag is set by the CPU to indicate to AXI DMA that this descriptor describes the start of the packet. The buffer associated with this descriptor is transmitted first. • 0 = Not start of frame. • 1 = Start of frame. ***Note:*** For C_SG_INCLUDE_STSCNTRL_STRM = 1, user application data from APP0 to APP4 of the SOF descriptor (TXSOF=1) is transmitted on the control stream output.
31 to 28	Reserved	This bit is reserved and should be written as zero.

MM2S_STATUS (MM2S Status)

This value provides status for MM2S transfers from memory map to stream.

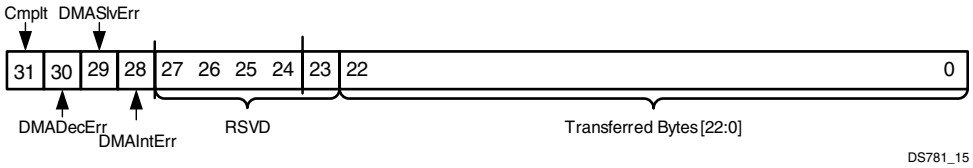

Figure 3-4: **MM2S_STATUS**

Table 3-6: **MM2S_STATUS Details**

Bits	Field Name	Description
22 to 0	Transferred Bytes	Indicates the size in bytes of the actual data transferred for this descriptor. This value indicates the amount of bytes to transmit out on MM2S stream. This value should match the Control Buffer Length field. The usable width of Transferred Bytes is specified by C_SG_LENGTH_WIDTH. A maximum of 8 MB of transfer can be described by this field. ***Note:*** Setting C_SG_LENGTH_WIDTH smaller than 23 reduces FPGA resource utilization.
27 to 23	Reserved	These bits are reserved and should be set to zero.
28	DMAIntErr	DMA Internal Error. Internal Error detected by primary AXI DataMover. This error can occur if a 0 length bytes to transfer is fed to the AXI DataMover. This only happens if the buffer length specified in the fetched descriptor is set to 0. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Internal Errors. • 1 = DMA Internal Error detected. DMA Engine halts.
29	DMASlvErr	DMA Slave Error. Slave Error detected by primary AXI DataMover. This error occurs if the slave read from the Memory Map interface issues a Slave Error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Slave Errors. • 1 = DMA Slave Error detected. DMA Engine halts.

Table 3-6: **MM2S_STATUS Details *(Cont'd)***

Bits	Field Name	Description
30	DMADecErr	DMA Decode Error. Decode Error detected by primary AXI DataMover. This error occurs if the address request is to an invalid address (that is, Descriptor Buffer Address points to an invalid address). This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Decode Errors. • 1 = DMA Decode Error detected. DMA Engine halts.
31	Cmplt	Completed. This indicates to the software that the DMA Engine has completed the transfer as described by the associated descriptor. The DMA Engine sets this bit to 1 when the transfer is completed. The software might manipulate any descriptor with the Completed bit set to 1 when in Tail Pointer Mode (currently the only supported mode). • 0 = Descriptor not completed. • 1 = Descriptor completed. ***Note:*** If a descriptor is fetched with this bit set to 1, the descriptor is considered a stale descriptor. An SGIntErr is flagged, and the AXI DMA engine halts.

MM2S_APP0 to MM2S_APP4 (MM2S User Application Fields 0 to 4)

This value provides User Application fields for MM2S control stream.

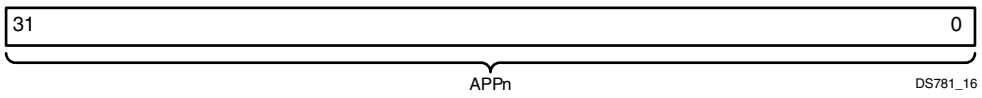

Figure 3-5: **MM2S_STATUS**

Table 3-7: **User Application Details**

Bits	Field Name	Description
31 to 0	APP0 to APP4	User application fields 0 to 4. Specifies user-specific application data. For C_SG_INCLUDE_STSCNTRL_STRM = 1, the Application (APP) fields of the SOF Descriptor are transmitted out the AXI Control Stream. For other MM2S descriptors with SOF = 0, the APP fields are fetched but ignored. ***Note:*** For C_SG_INCLUDE_STSCNTRL_STRM = 0, these fields are not fetched.

S2MM_NXTDESC (S2MM Next Descriptor Pointer)

This value provides the pointer to the next descriptor in the descriptor chain.

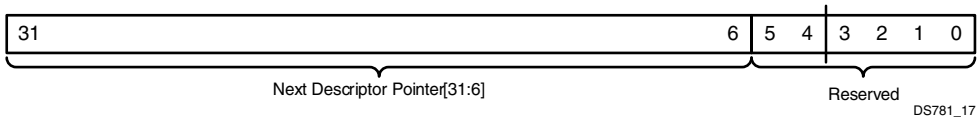

Figure 3-6: **S2MM_NXTDESC**

Table 3-8: **S2MM_NXTDESC Details**

Bits	Field Name	Description
5 to 0	Reserved	These bits are reserved and should be set to zero.
31 to 6	Next Descriptor Pointer	Indicates the lower order pointer pointing to the first word of the next descriptor. ***Note:*** Descriptors must be 16-word aligned, that is, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results.

S2MM_BUFFER_ADDRESS (S2MM Buffer Address)

This value provides the pointer to the buffer space available to transfer data from stream to system memory.

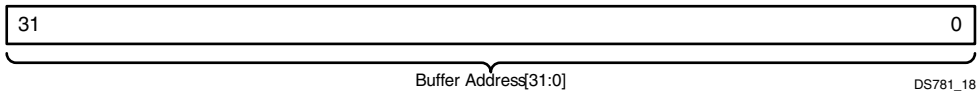

Figure 3-7: **S2MM Buffer Address**

Table 3-9: **S2MM_BUFFER_ADDRESS Details**

Bits	Field Name	Description
31 to 0	Buffer Address	Provides the location of the buffer space available to store data transferred from Stream to Memory Map. ***Note:*** If Data Realignment Engine is included (C_INCLUDE_S2MM_DRE = 1), the Buffer Address can be at any byte offset. If Data Realignment Engine is not included (C_INCLUDE_S2MM_DRE = 0), the Buffer Address must be S2MM stream data width aligned.

S2MM_CONTROL (S2MM Control)

This value provides control for S2MM transfers from stream to memory map.

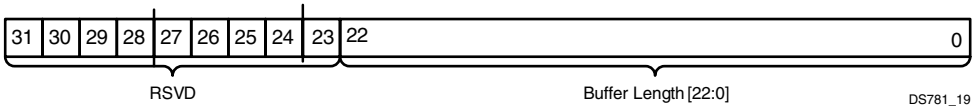

Figure 3-8: **S2MM_CONTROL**

Table 3-10: **S2MM_CONTROL Details**

Bits	Field Name	Description
22 to 0	Buffer Length	This value indicates the amount of space in bytes available for receiving data in an S2MM stream. The usable width of buffer length is specified by C_SG_LENGTH_WIDTH. A maximum of 8 MB of transfer can be described by this field. ***Note:*** The sum total of buffer space in the S2MM descriptor chain (that is, the sum of buffer length values for each descriptor in a chain) must be, at a minimum, capable of holding the maximum receive packet size. Undefined results occur if a packet larger than the defined buffer space is received. ***Note:*** Setting C_SG_LENGTH_WIDTH smaller than 23 reduces FPGA resource utilization.
31 to 23	Reserved	These bits are reserved and should be set to zero.

S2MM_STATUS (S2MM Status)

This value provides status for S2MM transfers from stream to memory map.

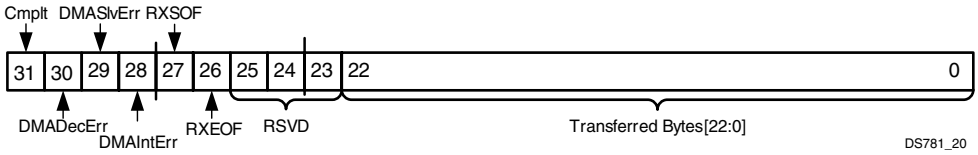

Figure 3-9: **S2MM_STATUS**

Table 3-11: **S2MM_STATUS Details**

Bits	Field Name	Description
22 to 0	Transferred Bytes	This value indicates the amount of data received and stored in the buffer described by this descriptor. This might or might not match the buffer length. For example, if this descriptor indicates a buffer length of 1024 bytes but only 50 bytes were received and stored in the buffer, then the Transferred Bytes field indicates 32h. The entire receive packet length can be determined by adding the Transferred Byte values from each descriptor from the RXSOF descriptor to the Receive Start Of Frame (RXEOF) descriptor. ***Note:*** The usable width of Transferred Bytes is specified by C_SG_LENGTH_WIDTH. A maximum of 8 Mbytes of transfer can be described by this field. ***Note:*** Setting C_SG_LENGTH_WIDTH smaller than 23 reduces FPGA resource utilization.
25 to 23	Reserved	These bits are reserved and should be set to zero.
26	RXEOF	End of Frame. Flag indicating buffer holds the last part of packet. This bit is set by AXI DMA to indicate to the CPU that the buffer associated with this descriptor contains the end of the packet. • 0 = Not end of frame. • 1 = End of frame. ***Note:*** For C_SG_INCLUDE_STSCNTRL_STRM = 1, User Application data sent through the status stream input is stored in APP0 to APP4 of the RXEOF descriptor.
27	RXSOF	Start of Frame. Flag indicating buffer holds first part of packet. This bit is set by AXI DMA to indicate to the CPU that the buffer associated with this descriptor contains the start of the packet. • 0 = Not start of frame. • 1 = Start of frame.
28	DMAIntErr	DMA Internal Error. Internal Error detected by primary AXI DataMover. This error can occur if a 0 length bytes to transfer is fed to the AXI DataMover. This only happens if the Buffer Length specified in the fetched descriptor is set to 0. This error can also be caused if an under-run or over-run condition occurs, and C_SG_USE_STSAPP_LENGTH = 1, indicating less bytes or more bytes than what was actually commanded are received. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Internal Errors. • 1 = DMA Internal Error detected. DMA Engine halts.
29	DMASlvErr	DMA Slave Error. Slave Error detected by primary AXI DataMover. This error occurs if the slave read from the Memory Map interface issues a Slave Error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Slave Errors. • 1 = DMA Slave Error detected. DMA Engine halts.

Table 3-11: **S2MM_STATUS Details *(Cont'd)***

Bits	Field Name	Description
30	DMADecErr	DMA Decode Error. Decode Error detected by primary AXI DataMover. This error occurs if the address request is to an invalid address (that is, Descriptor Buffer Address points to an invalid address). This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1. • 0 = No DMA Decode Errors. • 1 = DMA Decode Error detected. DMA Engine halts.
31	Cmplt	Completed. This indicates to the software that the DMA Engine has completed the transfer as described by the associated descriptor. The DMA Engine sets this bit to 1 when the transfer is completed. The software can manipulate any descriptor with the Completed bit set to 1. • 0 = Descriptor not completed. • 1 = Descriptor completed. ***Note:*** If a descriptor is fetched with this bit set to 1, the descriptor is considered a stale descriptor. An SGIntErr is flagged and the AXI DMA engine halts.

S2MM_APP0 to S2MM_APP3 (S2MM User Application Fields 0 to 3)

This value provides User Application field space for the S2MM received status on the Status Stream.

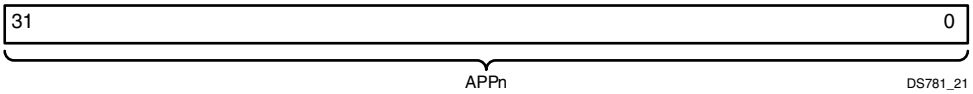

Figure 3-10: **S2MM_APP0 to S2MM_APP3**

Table 3-12: **User Application 0 to 3 Details**

Bits	Field Name	Description
31 to 0	APP0 to APP3	For C_SG_INCLUDE_STSCNTRL_STRM = 1, the status data received on the AXI Status Stream is stored into the APP fields of the EOF Descriptor. For other S2MM descriptors with EOF = 0, the APP fields are set to zero by the Scatter Gather Engine. ***Note:*** For C_SG_INCLUDE_STSCNTRL_STRM = 0, these fields are updated by the Scatter Gather Engine.

S2MM_APP4 (S2MM User Application Field 4)

This value provides User Application 4 field space for S2MM received status on the Status Stream.

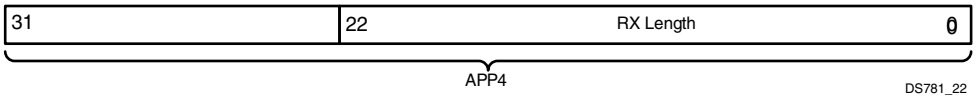

Figure 3-11: **S2MM_APP4**

Table 3-13: **User Application 4 Details**

Bits	Field Name	Description
31 to 0	APP4 / RxLength	User Application field 4 and Receive Byte Length. For C_SG_USE_STSAPP_LENGTH = 0, this field functions identically to APP0 to APP3 in that the status data received on the AXI Status Stream is stored into the APP4 field of the EOF Descriptor. For C_SG_USE_STSAPP_LENGTH = 1, this field has a dual purpose. First, the least significant C_SG_LENGTH_WIDTH bits specify the total number of receive bytes for a packet that were received on the S2MM primary data stream. Second, the remaining most significant bits are User Application data. ***Note:*** If using status application length (C_SG_USE_STSAPP_LENGTH=1), RxLength value must be stored in the C_SG_LENGTH_WIDTH least significant bits of the UserApp4 field in the AXI Status Stream Packet.

AXI DMA Scatter / Gather Operation (C_INCLUDE_SG = 1)

AXI DMA operation requires a memory-resident data structure that holds the list of DMA operations to be performed. This list of instructions is organized into what is referred to as a descriptor chain. Each descriptor has a pointer to the next descriptor to be processed. The last descriptor in the chain then points back to the first descriptor in the chain.

Scatter Gather operation allows a packet to be described by more than one descriptor. Typical use for this feature is to allow storing or fetching of headers from a location in memory and payload data from another location. Software applications that take advantage of this can improve throughput. To delineate packets in a buffer descriptor chain, the Start of Frame bit (TXSOF) and End of Frame bit (TXEOF) are utilized. When the DMA fetches a descriptor with the TXSOF bit set, the start of a packet is triggered. The packet continues with fetching the subsequent descriptors until it fetches a descriptor with the TXEOF bit set.

On the receive (S2MM) channel when a packet starts to be received, the AXI DMA marks the descriptor with an RXSOF indicating to the software that the data buffer associated with this descriptor contains the beginning of a packet. If the packet being received is longer in byte count than what was specified in the descriptor, the next descriptor's buffer is used to store the remainder of the receive packet. This fetching and storing process continues until the entire receive packet has been transferred. The descriptor being processed when the end of the packet is received is marked by AXI DMA with an RXEOF=1. This indicates to the software that the buffer associated with this descriptor contains the end of the packet.

Each descriptor contains, in the status field, the number of bytes actually transferred for that particular descriptor. The total number of bytes transferred for the receive packet can be determined by the software by walking from the RXSOF descriptor through the descriptor chain to the RXEOF descriptor. When the parameter C_INCLUDE_SG_DESC_QUEUE is set to 1, the Scatter Gather continues to fetch descriptors and store in a FIFO. This improves the DMA performance to a great extent. Setting this parameter to 0, will degrade the performance by inserting huge amounts of delay between descriptor fetching and descriptor update.

A DMA operation for the MM2S channel is set up and started by the following sequence:

1. Write the Current Descriptor register with the address of the starting descriptor.
2. Start the MM2S channel running by setting the run/stop bit to 1 (MM2S_DMACR.RS =1). The halted bit (DMASR.Halted) should deassert indicating the MM2S channel is running.
3. If desired, enable interrupts by writing a 1 to MM2S_DMACR.IOC_IrqEn and MM2S_DMACR.Err_IrqEn.
4. Write a valid address in the Tail Descriptor register.
5. Writing to Tail Descriptor register triggers the DMA to start fetching the descriptors from the memory.
6. The fetched descriptors are processed, data read from the memory and then output on the MM2S streaming channel.

A DMA operation for the S2MM channel is set up and started by the following sequence:

1. Write the Current Descriptor register with the address of the starting descriptor.
2. Start the S2MM channel running by setting the run/stop bit to 1 (S2MM_DMACR.RS =1). The halted bit (DMASR.Halted) should deassert indicating the S2MM channel is running.
3. If desired, enable interrupts by writing a 1 to S2MM_DMACR.IOC_IrqEn and S2MM_DMACR.Err_IrqEn.
4. Write a valid address in the Tail Descriptor register.
5. Writing to Tail Descriptor register triggers the DMA to start fetching the descriptors from the memory.
6. The fetched descriptors are processed and any data coming on the S2MM streaming channel is written to the Memory.

Descriptor Management

Prior to starting DMA operations, the software application must set up a descriptor chain. When the AXI DMA begins processing the descriptors, it fetches, processes, and then updates the descriptors. By analyzing the descriptors, the software application can read the status on the associated DMA transfer, fetch user information on receive (S2MM) channels, and determine completion of the transfer. With this information, the software application can manage the descriptors and data buffers.

Software applications process each buffer associated with completed descriptors and reallocate the descriptor for AXI DMA use. To prevent software and hardware from stepping on each other, a Tail Pointer Mode is created. The tail pointer is initialized by software to point to the end of the descriptor chain. This becomes the pause point for hardware. When hardware begins running, it fetches and processes each descriptor in the chain until it reaches the tail pointer. The AXI DMA then pauses descriptor processing. The software is allowed to process and re-allocate any descriptor with the Complete bit set to 1.

While the software is processing descriptors, AXI DMA hardware is prevented from stepping on the descriptors being processed by software by the tail pointer. When the software finishes re-allocating a set of descriptors, it then moves the tail pointer location to the end of the re-allocated descriptors by writing to the associated channel's TAILDESC register. The act of writing to the TAILDESC register causes the AXI DMA hardware, if it is paused at the tail pointer, to begin processing descriptors again. If the AXI DMA hardware is not paused at the TAILDESC pointer, writing to the TAILDESC register has no effect on the hardware. In this situation, the AXI DMA continues to process descriptors until reaching the new tail descriptor pointer location. Descriptor Management has to done by the software. AXI DMA does not manage the descriptors.

MM2S Descriptor Settings/Relationship

The relationship between descriptor SOF/EOF settings and the AXI Control Stream are illustrated in Figure 3-12. The descriptor with SOF=1 is the beginning of the packet and resets DRE for the MM2S direction. The User Application fields for this descriptor are also presented on the AXI Control Stream if the AXI Control Stream is included (C_SG_INCLUDE_STSCNTRL_STRM = 1). User Application fields following descriptor with SOF=1, up to and including descriptor with EOF =1, are ignored by the AXI DMA engine. If C_SG_INCLUDE_STSCNTRL_STRM = 0, the User Application fields are not fetched by the SG Fetch Engine.

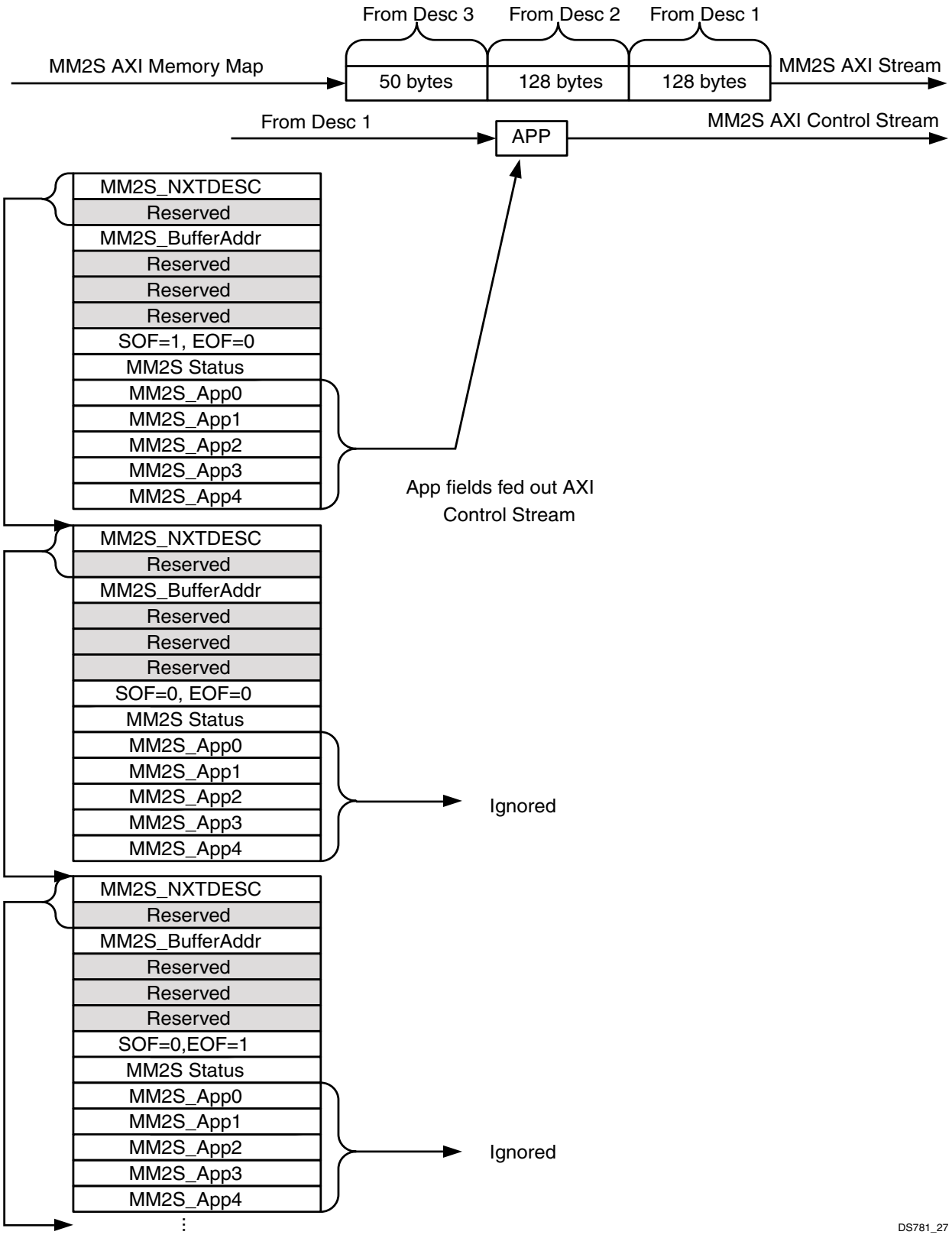

Figure 3-12: **Detail of Descriptor Relationship to MM2S Stream and Control Stream**

Control Stream Detail

The AXI control stream is provided from the Scatter Gather Descriptor to a target device for User Application data. The control data is associated with the MM2S primary data stream and can be sent out of AXI DMA prior to, during, or after the primary data packet. Throttling by the target device is allowed, and throttling by AXI DMA can occur. Figure 3-13 shows an example of how descriptor User Application fields are presented on the AXI control stream. AXI DMA inserts a flag indicating the data type to the target device. This is sent as the first word. For Ethernet, the control tag is 0xA in the four Most Significant Bits (MSBs) of the first word.

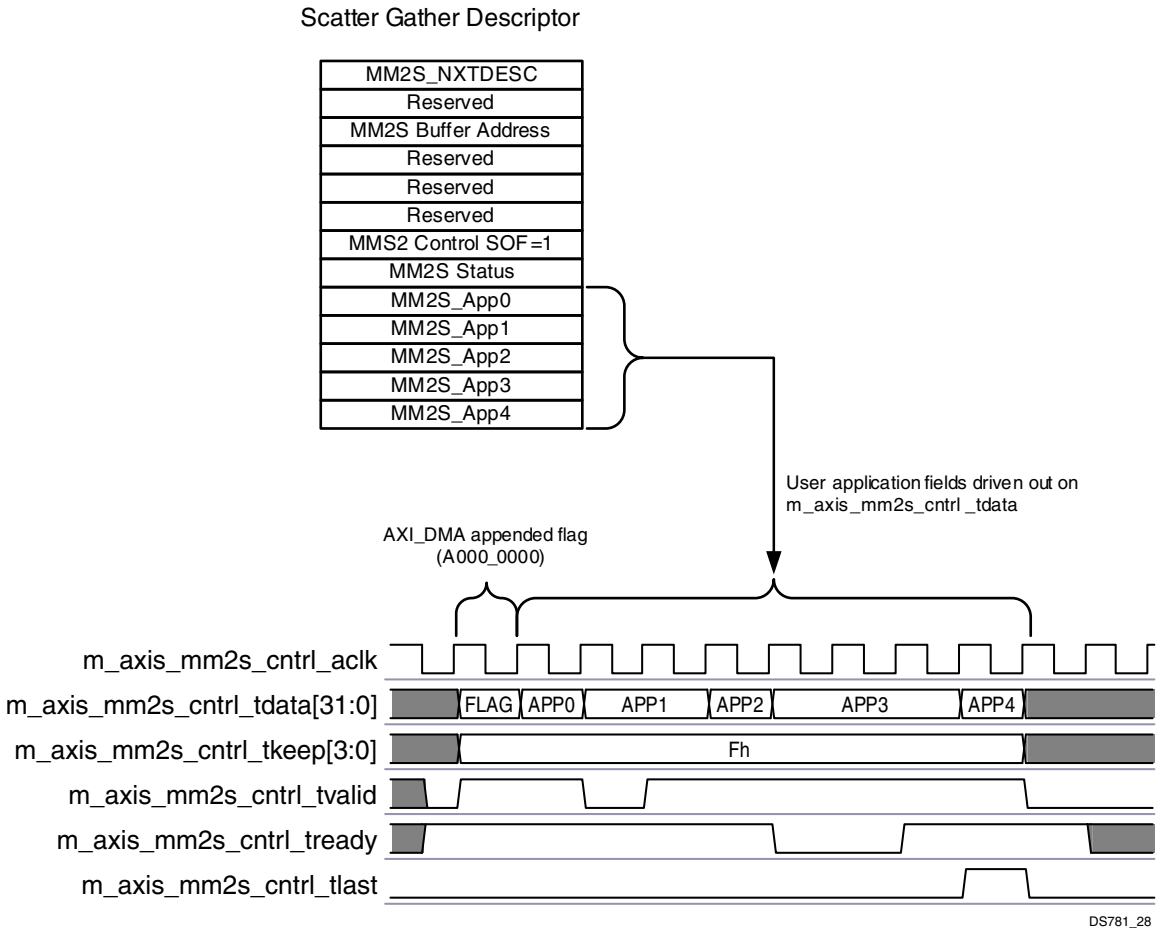

Figure 3-13: **Example User Application Field / Timing for MM2S Control Stream**

S2MM Descriptor Settings/Relationship

The relationship between descriptor RXSOF/RXEOF settings and the AXI Status Stream are illustrated in Figure 3-14. The descriptor with RXSOF=1 describes the buffer containing the first part of the receive packet. The Descriptor with RXEOF=1 describes the buffer containing the last part of the receive packet.

For proper operation, the software must specify enough buffer space (that is, the sum total buffer lengths in each descriptor of the descriptor chain) to be greater than or equal to the maximum sized packet that is received.

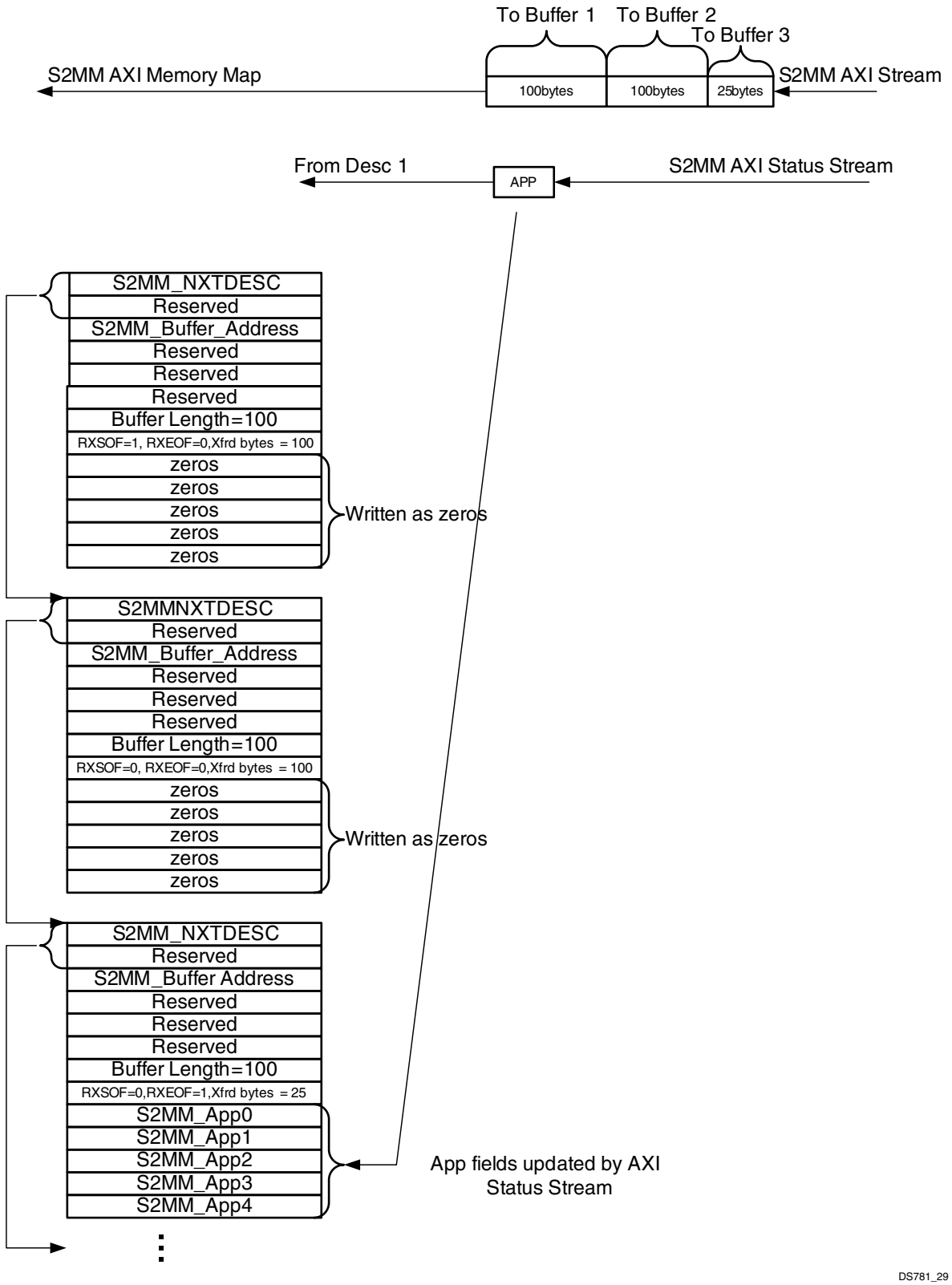

Figure 3-14: **Detail of Descriptor Relationship to S2MM Stream and Status Stream**

If the AXI Status Stream is included (C_SG_INCLUDE_STSCNTRL_STRM = 1), the status received is stored in the User Application fields (APP0 to APP4) of the descriptor with RXEOF set.

The actual byte count of received and stored data for a particular buffer is updated to the Transferred Bytes field in the associated descriptor. The software can determine how many bytes were received by walking the descriptors from RXSOF to RXEOF and adding the Bytes Transferred fields to get a total byte count. For applications where you provide the total length in the status stream, this value is stored in the user-defined application location in the descriptor with RXEOF=1.

Status Stream Detail

The AXI status stream is provided for transfer of target device status to User Application data fields in the Scatter Gather descriptor. The status data is associated with the S2MM primary data stream. As shown in Figure 3-15, the status packet updates to the app fields of the detected last descriptor (RXEOF = 1) describing the packet. Normally, the status stream should come at the start of the S2MM data stream. If the STS APP length is not used, then the status stream can come at any time during the course of S2MM frame. The EOF BD update would happen only when the entire status stream is received.

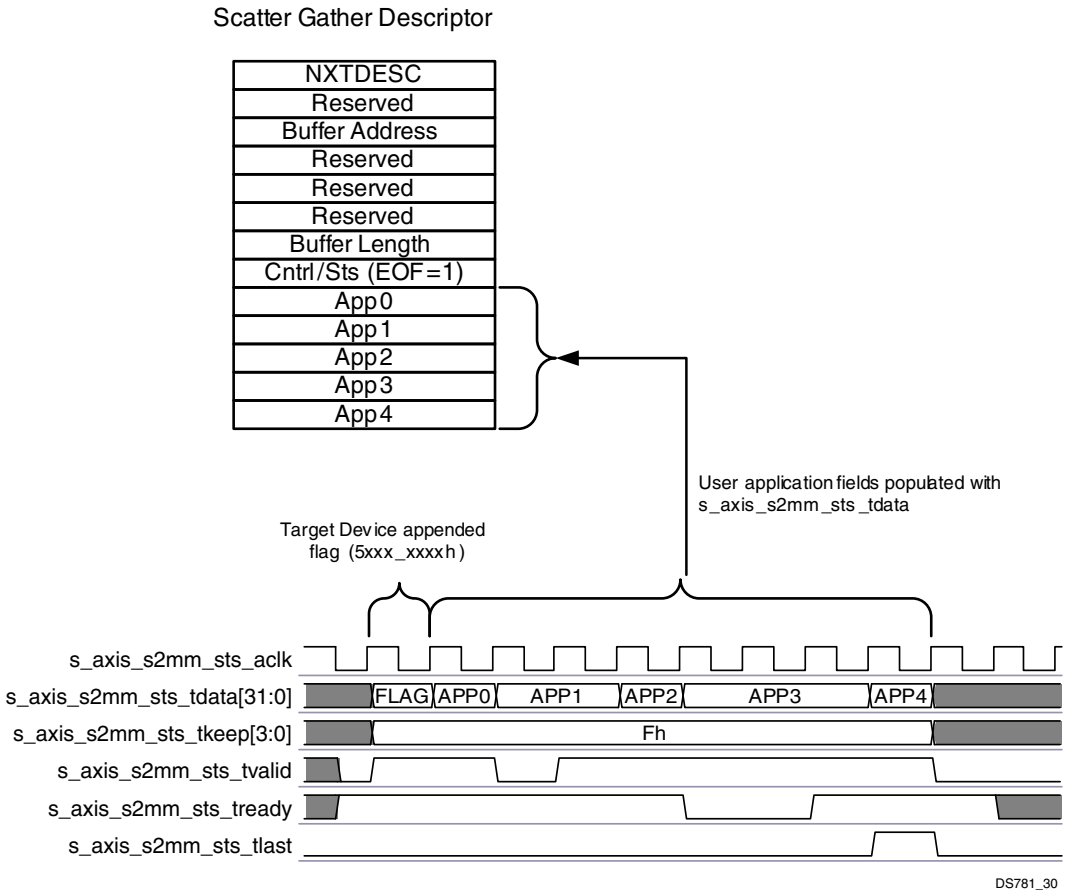

Figure 3-15: **Example User Application Field / Timing for S2MM Status Stream**

Multichannel DMA Support

Multichannel mode enables DMA to connect to multiple masters and slaves on the streaming side. A new set of signals associated with source and destination signaling are added. They are:

- TID - 5-bit signal. Provides a stream identifier and can be used to differentiate between multiple streams of data that are being transferred across the same interface.
- TDEST - 5-bit signal. Provides coarse routing information for the data stream.
- TUSER – 4-bit signal. User defined sideband signaling.

Scatter Gather Mode (C_INCLUDE_SG = 1)

New descriptor fields are added to support multichannel and 2D transfers. As described in AXI DMA Multichannel Operation, AXI DMA v6.01.a supports efficient two-dimensional memory access patterns, transferring 2-D blocks across the AXI4-Stream channel. Memory access patterns are controlled with three parameters: HSIZE, VSIZE, and STRIDE. Multiple descriptors per packet are supported through the Start of Packet and End of Packet flags.

In MC mode, the DMA does not provide separate status and control streams. Instead, sideband control information is contained within the AXI4-Stream in TID, DEST, and TUSER fields. Accordingly, the IP customization GUI will set parameter C_SG_INCLUDE_STSCNTRL_STRM = 0, and the TX/RX descriptors do not provide separate AppWord fields.

AXI DMA can be set in multichannel mode by setting the parameter C_ENABLE_MULTI_CHANNEL to 1 and selecting the required number of channels on MM2S and S2MM paths through C_NUM_MM2S_CHANNELS and C_NUM_S2MM_CHANNELS respectively.

MM2S (Tx) Descriptor

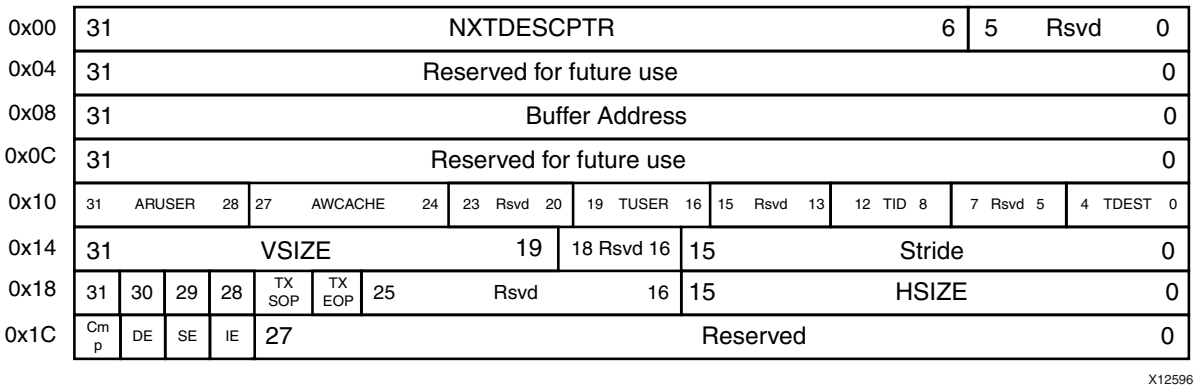

Figure 3-16: **Tx Descriptor**

Table 3-14: **Tx Descriptor Fields**

Address Space Offset	Name	Description
00h	NXTDESC	Bits 5:0 - Reserved Bits 31:6 - Next Descriptor Pointer
04h	RESERVED	Bits 31:0 - Reserved
08h	BUFFER_ADDRESS	Bits 31:0 - Buffer Address Provides the location of the data to transfer from Memory Map to Stream
0Ch	RESERVED	Bits 31:0 - Reserved
10h	MC_CTL	Multichannel Control bits. Bits 4:0 – TDEST provides routing information for the data stream. TDEST values are static for entire packet. TDEST values provided in the Tx descriptor field are presented on TDEST signals of streaming side.
		• Bits 7:5 – Reserved
		• Bits 12:8 - TID provides a stream identifier. TID values are static for entire packet. TID values provided in the Tx descriptor field are presented on TID signals of the streaming side.
		• Bits 15:13 – Reserved
		• Bits 19:16 – TUSER – sideband signals used for user-defined information. TUSER values are static for entire packet. TUSER values provided in the Tx descriptor field are presented on TUSER signals of streaming side.
		• Bits 23:20 – Reserved
		• Bits 27:24 – ARCACHE – Cache type. This signal provides additional information about the cacheable characteristics of the transfer. See the AMBA® *AXI Protocol Specification* for a different decoding mechanism. ARCACHE values from Tx descriptor are presented on ARCACHE [3:0] bus during address cycle. Default value of this field is 0011.
		• Bits 31:28 – ARUSER – sideband signals used for user-defined information. ARUSER values from Tx descriptor are presented on ARUSER [3:0]. ARUSER values and their interpretations are user-defined. You can keep ARUSER static for the entire packet by programming the same values in all the descriptors with in a chain.

Table 3-14: **Tx Descriptor Fields *(Cont'd)***

Address Space Offset	Name	Description
14h	STRIDE_VSIZE	• Bits 15:0 - Stride Control. It is the address distance between the first address of successive "horizontal" reads. Reads would start at the Buffer Address and read HSIZE bytes, then skip STRIDE-HSIZE addresses and read HSIZE bytes, and so on. This would continue until VSIZE lines have been read. On AXI4-Stream this is transmitted out on the m_axis_mm2s_ interface as one contiguous packet and is terminated with a single assertion of TLAST on the last data beat of the transfer. • Bits 18:16 - Reserved • Bits 31:19 – Number of "horizontal lines" for strided access. Can represent two-dimensional data access of video or into a 2-D matrix. VSIZE number of transfers, each HSIZE bytes long, are expected to be transmitted for each packet.
18h	HSIZE	• Bits 15:0 – Number of bytes to transfer in each "horizontal line" from successive contiguous byte addresses. Can represent a portion of a video line or a portion of a matrix row when the matrix is read in row major order. C_SG_LENGTH_WIDTH specifies the bit width of the HSIZE parameter. • Bits 25:16 - Reserved • Bit 26 – TXEOP – End of packet flag. It indicates the buffer associated with this descriptor is transmitted last. This flag is set by the CPU. 0 – Not end of packet. 1 – End of packet • Bit 27 – TXSOP – Start of packet flag. It indicates the buffer associated with this descriptor is transmitted first. This flag is set by the CPU. 0 – Not start of packet. 1 – Start of packet • Bits 31:28 - Reserved

Table 3-14: **Tx Descriptor Fields *(Cont'd)***

Address Space Offset	Name	Description
1Ch	MC_STS	Multichannel Control bits. • Bits 27:0 – Reserved • Bit 28 – IE – DMA Internal Error due to under-run or over-run conditions. 0 – No DMA Internal Errors 1 – DMA Internal Error detected. DMA Engine halts • Bit 29 – SE – DMA Slave Error. This error occurs if the slave read from the Memory Map interface issues a Slave Error. 0 – No DMA Slave Errors 1 – DMA Slave Error detected. DMA Engine halts • Bit 30 – DE – DMA Decode Error. This error occurs if the address request is to an invalid address. 0 – No DMA Decode Errors 1 – DMA Decode Error detected. DMA Engine halts • Bit 31 – Cmp – Completed. This indicates to the software that the DMA engine has completed the transfer. 0 – Descriptor not completed 1 – Descriptor completed

1. The ARCACHE, ARUSER values are important from the AXI read prospective. These values should be specified in the descriptor as needed. For normal operation ARCACHE should be set to "0011" while ARUSER can be set to "0000".
2. A value of '0' on VSIZE is illegal and results in Multi Channel DMA not functioning as expected.

S2MM (Rx) Descriptor

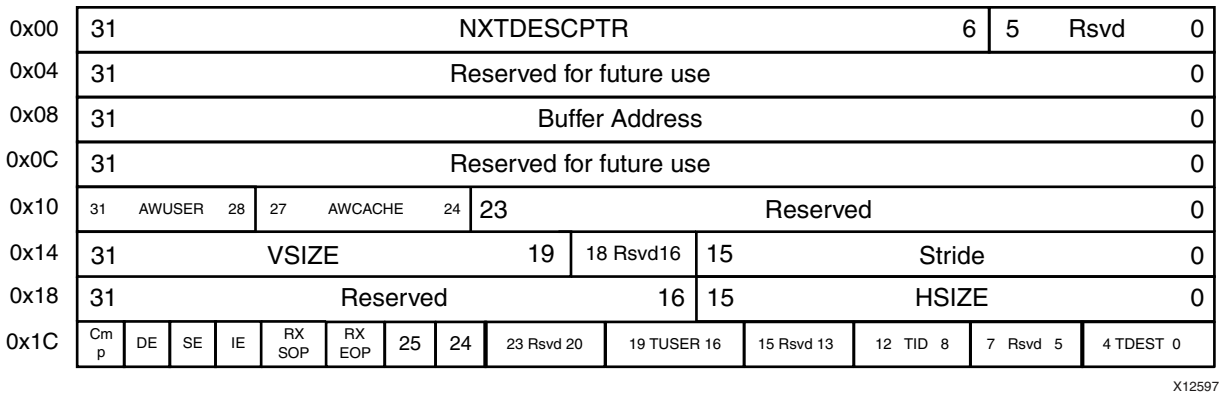

Figure 3-17: **Rx Descriptor**

Table 3-15: **Rx Descriptor Fields**

Address Space Offset	Name	Description
00h	NXTDESC	Bits 5:0 - Reserved Bits 31:6 - Next Descriptor Pointer
04h	RESERVED	Bits 31:0 - Reserved
08h	BUFFER_ADDRESS	Bits 31:0 – Buffer Address Provides the location of the buffer space available to store data transferred from Stream to Memory Map
0Ch	RESERVED	Bits 31:0 - Reserved
10h	CACHE_USER_CTL	Bit 23:0 – Reserved
		Bit 27:24 – AWCACHE – Cache type. This signal provides additional information about the cacheable characteristics of the transfer. See the AMBA *AXI Protocol Specification* for a different decoding mechanism. AWCACHE values from Rx descriptor are presented on AWCACHE [3:0] bus during address cycle. Default value of this field should be 0011.
		Bits 31:28 – AWUSER – sideband signals used for user-defined information. AWUSER values from Rx descriptor are presented on AWUSER [3:0]. AWUSER values and their interpretations are user-defined. You can keep AWUSER static for entire packet by programming same values in all the descriptors within a chain.
14h	STRIDE_VSIZE	Bits 15:0 - Stride Control. It is the address distance between the first address of successive “horizontal” writes. Writes start at the Buffer Address and write HSIZE bytes, then skip STRIDE-HSIZE addresses and write HSIZE bytes, and so on. This continues until VSIZE has been written. On AXI4-Stream this is received on the s_axis_s2mm_ interface as one contiguous packet and is terminated with a single assertion of TLAST on the last data beat of the transfer.
		Bits 18:16 – Reserved
		Bits 31:19 – Number of “horizontal lines” for strided access. Can represent two-dimensional data access of video or into a 2-D matrix. VSIZE number of transfers, each HSIZE bytes long, are expected to be received for each packet.
18h	HSIZE	Bits 15:0 – Number of bytes to transfer in each “horizontal line” from successive contiguous byte addresses. Can represent a portion of a video line or a portion of a matrix row when matrix is stored in row major order. C_SG_LENGTH_WIDTH specifies the bit width of the HSIZE parameter. Bits 31:16 - Reserved

Table 3-15: **Rx Descriptor Fields *(Cont'd)***

Address Space Offset	Name	Description
1Ch	MC_STS	Multichannel Status bits. Bits 4:0 – TDEST provides routing information for the data stream. TDEST values are static for entire packet. TDEST values are captured from incoming stream and updated in this field
		Bits 7:5 – Reserved
		Bits 12:8 - TID provides a stream identifier. TID values are static for entire packet. TID values are captured from incoming stream and updated in this field.
		Bits 15:13 – Reserved
		Bits 19:16 – TUSER – sideband signals used for user-defined information. TUSER values are static for entire packet. TUSER values are captured from incoming stream and updated in this field.
		Bits 25:20 – Reserved
		Bits 25:24 – Reserved
		Bit 26 – RXEOP – End of packet flag. It indicates the buffer associated with this descriptor contains the last part of packet. This flag is set by AXI DMA. • 0 – Not end of packet • 1 – End of packet
		Bit 27 – RXSOP – Start of packet flag. It indicates the buffer associated with this descriptor contains the start of the packet. This flag is set by AXI DMA. • 0 – Not start of packet • 1 – Start of packet
		Bit 28 – IE – DMA Internal Error due to under-run or over-run conditions. • 0 – No DMA Internal Errors • 1 – DMA Internal Error detected. DMA Engine halts
		Bit 29 – SE – DMA Slave Error. This error occurs if the slave read from the Memory Map interface issues a Slave Error. • 0 – No DMA Slave Errors • 1 – DMA Slave Error detected. DMA Engine halts.
		Bit 30 – DE – DMA Decode Error. This error occurs if the address request is to an invalid address. • 0 – No DMA Decode Errors • 1 – DMA Decode Error detected. DMA Engine halts.

Table 3-15: **Rx Descriptor Fields *(Cont'd)***

Address Space Offset	Name	Description
1Ch (Continued)	MC_STS (Continued)	Bit 31 – Cmp – Completed. This indicates to the software that the DMA engine has completed the transfer. • 0 – Descriptor not completed • 1 – Descriptor completed

1. The AWCACHE, AWUSER values are important from the AXI read prospective. These values should be specified in the descriptor as needed. For normal operation AWCACHE should be set to "0011" while AWUSER can be set to "0000".
2. A value of '0' on VSIZE is illegal and results in Multi Channel DMA not functioning as expected.

AXI DMA Multichannel Operation

This section describes the end-to-end control and data flow of descriptors and associated data for both MM2S side and S2MM side.

MM2S

It is similar to normal AXI DMA operation. When MM2S_CURDESC and MM2S_TAILDESC are programmed by software, AXI DMA fetches a chain of descriptors and processes until it reaches tail descriptor. In AXI DMA v6.01.a, TDEST, TID, and TUSER fields are assumed to remain constant for an entire packet as defined in the descriptors. That is, each packet transfer across a logical channel defined by (TDEST, TID, TUSER) runs to completion before the DMA transfers another packet. Although packet transfers for multiple channels can be interleaved, after started, each must run to completion before another transfer can occur. It is the user's responsibility to avoid deadlock scenarios under this assumption. The AXI DMA does not signal error conditions if the (TDEST, TID, TUSER) fields within the descriptors do not adhere to these assumptions. It is up to the software to maintain consistency.

Tx Descriptor contains control and status fields in multichannel mode. There is no status update on this descriptor for the number of bytes transferred at the end of the transaction. Error information is captured in registers along with the current descriptor pointer of the failing descriptor. The completion of a transfer for a chain can be known by polling the IDLE bit of MM2S_DMASR or through an interrupt by enabling it in MM2S_DMACR.

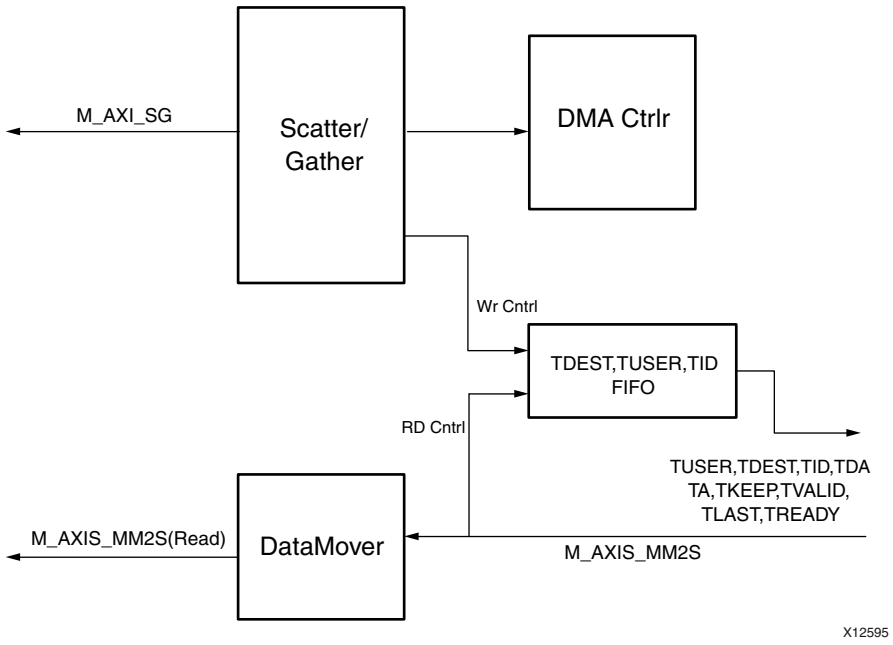

Figure 3-18: **MM2S Control and Datapath Flow**

S2MM

TDEST signals are sampled from incoming stream. This value is used by the DMA controller to read the corresponding S2MM_CURDESC and S2MM_TAILDESC and present to the Scatter Gather module. This module in turn fetches a chain of descriptors for that TDEST. AXI DMA holds the streaming data by deasserting `TREADY` until the corresponding descriptors are fetched. Then, AXI DMA writes would start at the Buffer Address and continue until it receives TLAST from streaming side.

TDEST, TID, TUSER values are captured from incoming streams and stored internally. These values are not expected to change in the middle of a packet. These values are updated in each descriptor of the chain after the completion of data writes for that descriptor onto the MM side along with other status bits.

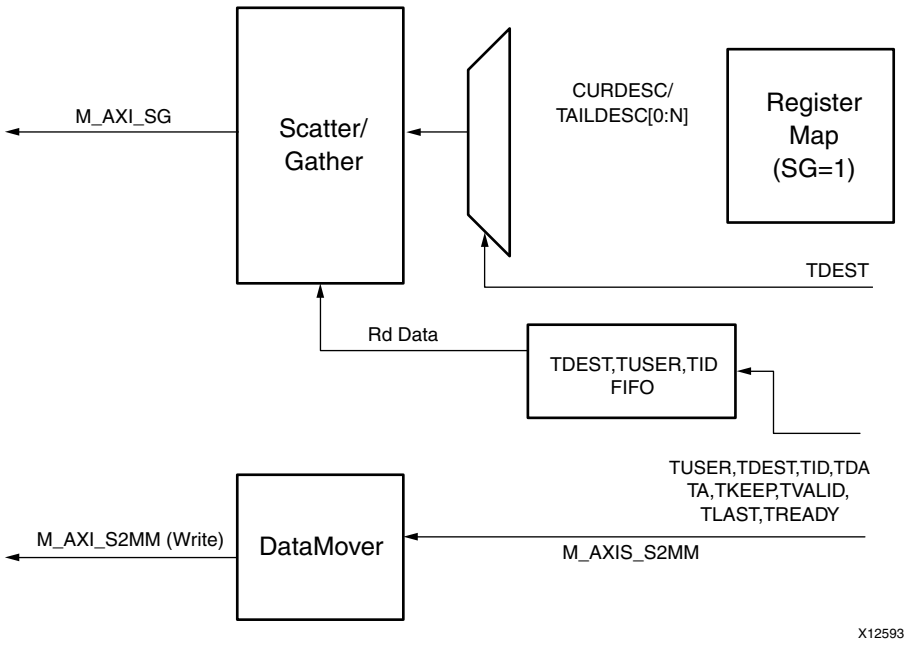

Figure 3-19: **S2MM Control and Datapath Flow**

2D Transfers

In Multichannel Mode, AXI DMA v6.01.a supports two dimensional (2-D) memory access patterns to be efficiently transferred with an AXI4-Stream channel. Memory access patterns index into a one-dimensional address space in 'row major' order, where location L[row][column] = ADDRESS[row * NUM_COLUMNS + col]'.

Access patterns are controlled with descriptor fields HSIZE, VSIZE, and STRIDE, which enable the transfer of sub-blocks within the (implicit) 2-D array. HSIZE is specified between starting addresses for successive 'row' sub-blocks. For 2D transfers, the HSIZE, VISZE and STRIDE should be byte aligned. Having unaligned values of HSIZE, VSIZE or STRIDE will cause unexpected behavior.

Each read (MM2S) or write (S2MM) transfer consists of VSIZE transfers, each of size HSIZE. The starting address of each successive transfer is STRIDE address from the starting address of the previous transfer (initially, the BaseAddr of the packet transfer). Figure 3-20 shows the example of the two-dimensional data format:

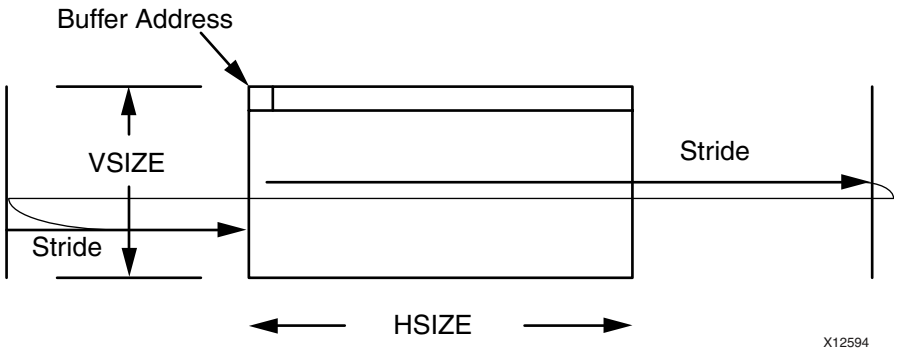

Figure 3-20: **2D Data Format**

MM2S

Reads start at the Buffer Address and consists of VSIZE read bursts, each having HSIZE bytes. The starting address of each read burst is a STRIDE address greater than the starting address of the previous burst read.

Example: Buffer Address = 08, VSIZE = 06, HSIZE = 256 bytes and Stride = 512 bytes

In this case, Reads start at buffer address location 08 and continue to read HSIZE (256) bytes. The second line starts at Buffer address+Stride = 512+8 = 520. It continues to read HSIZE (256) bytes. The third line starts at 520+512 = 1032 and the fourth line starts at 1032+512 = 1544. Likewise it reads VSIZE lines.

On AXI4-Stream this is transmitted out on `m_axis_mm2s_interface` as one contiguous packet and is terminated with the assertion of `TLAST` on the last data beat of the transfer.

S2MM

Writes start at the Buffer Address and consist of VSIZE write bursts, each having HSIZE bytes. The starting address of each write burst is a STRIDE address greater than the starting address of the previous burst write. On AXI4-Stream this is received in on `s_axis_s2mm_interface` as one contiguous packet and is terminated with a single assertion of `TLAST` on the last data beat of the transfer.

Limitations with Multi Channel Mode

- Unable to support descriptor queuing in S2MM path for multichannel mode
- Unable to support small packet sizes for S2MM path (back to back packets of 4 or less data beats)
- 2D access is supported only for aligned addresses and Hsize.

Register Map for Multichannel (SG = 1)

Register map is modified to support up to 16 TDEST on the S2MM path.

00	MM2S_DMACR
04	MM2S_DMASR
08	MM2S_CURDESC0
0C	Reserved
10	MM2S_TAILDESC0
14	Reserved
18	Reserved
1C	Reserved
20	Reserved
24	Reserved
28	Reserved
2C	SG_CTL
30	S2MM_DMACR
34	S2MM_DMASR
38	S2MM_CURDESC0
3C	Reserved
40	S2MM_TAILDESC0
44	Reserved
48	Reserved
4C	Reserved
50	Reserved
54	Reserved
58	Reserved
5C	Reserved
60	Reserved
64	Reserved
68	Reserved
6C	Reserved
70	S2MM_CURDESC1
74	Reserved
78	S2MM_TAILDESC1
7C	Reserved
80	Reserved
84	Reserved
88	Reserved
8C	Reserved
90	S2MM_CURDESC2
94	Reserved
98	S2MM_TAILDESC2
9C	Reserved
A0	Reserved
A4	Reserved
A8	Reserved
AC	Reserved
B0	S2MM_CURDESC3
B4	Reserved
B8	S2MM_TAILDESC3
BC	Reserved

C0	Reserved
C4	Reserved
C8	Reserved
CC	Reserved
D0	S2MM_CURDESC4
D4	Reserved
D8	S2MM_TAILDESC4
DC	Reserved
E0	Reserved
E4	Reserved
E8	Reserved
EC	Reserved
F0	S2MM_CURDESC5
F4	Reserved
F8	S2MM_TAILDESC5
FC	Reserved
100	Reserved
104	Reserved
108	Reserved
10C	Reserved
110	S2MM_CURDESC6
114	Reserved
118	S2MM_TAILDESC6
11C	Reserved
120	Reserved
124	Reserved
128	Reserved
12C	Reserved
130	S2MM_CURDESC7
134	Reserved
138	S2MM_TAILDESC7
13C	Reserved
140	Reserved
144	Reserved
148	Reserved
14C	Reserved
150	S2MM_CURDESC8
154	Reserved
158	S2MM_TAILDESC8
15C	Reserved
160	Reserved
164	Reserved
168	Reserved
16C	Reserved
170	S2MM_CURDESC9
174	Reserved
178	S2MM_TAILDESC9
17C	Reserved

180	Reserved
184	Reserved
188	Reserved
18C	Reserved
190	S2MM_CURDESC10
194	Reserved
198	S2MM_TAILDESC10
19C	Reserved
1A0	Reserved
1A4	Reserved
1A8	Reserved
1AC	Reserved
1B0	S2MM_CURDESC11
1B4	Reserved
1B8	S2MM_TAILDESC11
1BC	Reserved
1C0	Reserved
1C4	Reserved
1C8	Reserved
1CC	Reserved
1D0	S2MM_CURDESC12
1D4	Reserved
1D8	S2MM_TAILDESC12
1DC	Reserved
1E0	Reserved
1E4	Reserved
1E8	Reserved
1EC	Reserved
1F0	S2MM_CURDESC13
1F4	Reserved
1F8	S2MM_TAILDESC13
1FC	Reserved
200	Reserved
204	Reserved
208	Reserved
20C	Reserved
210	S2MM_CURDESC14
214	Reserved
218	S2MM_TAILDESC14
21C	Reserved
220	Reserved
224	Reserved
228	Reserved
22C	Reserved
230	S2MM_CURDESC15
234	Reserved
238	S2MM_TAILDESC15

X12591

Figure 3-21: **Register Map for Multichannel Support in SG = 1**

SG_CTL

New register SG_CTL added at 0x2C. It contains `SG_CACHE[3:0]` and `SG_USER[3:0]` that are wired to the corresponding signals on the AXI SG interface.

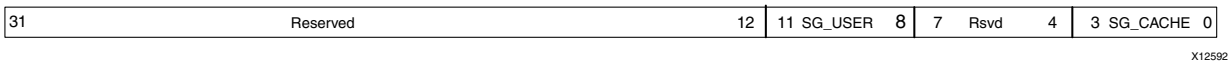

Figure 3-22: **SG Control Register**

AXI DMA System Configuration

System configuration should be taken into consideration when using AXI DMA. Due to the high performance and low latency design of AXI DMA, throttling or back pressure on the AXI4-Stream ports of AXI DMA (MM2S and S2MM) subsequently applies back pressure on the associated channel's AXI4 Memory Map side. Depending on a user's system configuration this back pressure can lead to a deadlock situation where, for example, a write transfer on S2MM to a single ported memory controller cannot complete because of a throttled read transfer on MM2S. A loopback type system where the MM2S stream interface is looped back to S2MM stream interface, such as when loopback is turned on in AXI Ethernet, can present such a deadlock scenario.

The AXI Interconnect interfacing to the AXI DMA AXI4 Memory Map ports on MM2S and S2MM can be configured to prevent the deadlock scenario described previously. Read and Write Data FIFOs can be turned on in the AXI Interconnect to allow read and/or write data to be buffered up. Enabling the FIFOs along with limiting the number of outstanding read and write requests accepted by AXI Interconnect guarantees that all requested data to be transferred can be accepted by the AXI Interconnect preventing deadlock.

To enable these AXI Interconnect features, four non-hdl parameters are provided:

- C_INTERCONNECT_M_AXI_MM2S
- C_INTERCONNECT_M_AXI_MM2S_READ_ISSUING
- C_INTERCONNECT_M_AXI_MM2S_READ_FIFO_DEPTH
- C_INTERCONNECT_M_AXI_S2MM_WRITE_ISSUING
- C_INTERCONNECT_M_AXI_S2MM_WRITE_FIFO_DEPTH

Setting these parameters correctly configures the AXI Interconnect interfaced to the Memory Map Ports of the AXI DMA. For AXI DMA *_ISSUING multiplied by the associated channels *_BURST_SIZE must be less than the *_FIFO_DEPTH to prevent the deadlock scenario. For example, if C_MM2S_BURST_SIZE is set to 16 and C_INTERCONNECT_M_AXI_MM2S_READ_ISSUING is set to 4, then the product of these two values would be 16 X 4 = 64.

Therefore the setting C_INTERCONNECT_M_AXI_MM2S_READ_FIFO_DEPTH = 512 would be sufficient to satisfy the requirements (16 X 4 = 64 which is less than 512). Here is the formula presented for clarity:

*_BURST_SIZE X *_ISSUING < *_FIFO_DEPTH

If building AXI DMA using the EDK tool suite, the issuing parameter is set automatically based on the AXI DMA burst size parameter and the fifo depth parameter is set to 512. When looking at FPGA resource utilization in an EDK system with AXI DMA, the AXI Interconnect instantiates FIFOs for both MM2S and S2MM channels of AXI DMA.

Errors

Any detected error results in the AXI DMA gracefully halting. When an error is detected, the errored channel's DMACR.RS bit is set to 0. Per AXI protocol, all AXI transfers must complete. Therefore, the AXI DMA completes all pending transactions before setting the errored channel's DMASR.Halted bit. When the DMASR.Halted bit is set to 1, the AXI DMA engine is truly halted.

For Scatter / Gather Mode the pointer to the descriptor associated with the errored transfer is updated to the channels CURDESC_PTR register. On the rare occurrence that more than one error is detected, only the CURDESC_PTR register for one of the errors is logged. To resume operations, a reset must be issued to the AXI DMA.

The following is a list of possible errors:

- **DMAIntErr**. DMA Internal Error indicates that an internal error in the AXI DataMover was detected. For Scatter / Gather Mode (C_INCLUDE_SG = 1) this can occur under two conditions. First, for MM2S and S2MM channels, it can occur when a BTT = 0 is written to the primary AXI DataMover. This would happen if a descriptor is fetched with the buffer length = 0.

  Secondly, for S2MM channels, the internal error can occur if C_SG_USE_STSAPP_LENGTH = 1 and an underflow condition occurs on the S2MM primary stream interface. This indicates that a failure in the system has occurred. For example, a soft reset issued in the stream source interrupts the receive stream, or a reported length on the status stream does not match actual data received.

  For Simple DMA Mode (C_INCLUDE_SG = 0) this error can occur on the S2MM side when the data received from the S2MM stream is more than what is specified in the BTT field of the register.

- **DMASlvErr**. DMA Slave Error occurs when the slave to or from which data is transferred responds with a SLVERR.

- **DMADecErr**. DMA Decode Error occurs when the address request is targeted to an address that does not exist.
- **SGIntErr**. Scatter Gather Internal Error occurs when a BTT = 0. This error also occurs if a fetched descriptor already has the Complete bit set. This condition indicates to the AXI DMA that the descriptor has not been processed by the CPU, and therefore is considered stale. This error is for Scatter / Gather Mode only (C_INCLUDE_SG = 1).
- **SGSlvErr**. Scatter Gather Slave Error occurs when the slave to or from which descriptors are fetched and updated responds with a SLVERR. This error is for Scatter / Gather Mode only (C_INCLUDE_SG = 1).
- **SGDecErr**. Scatter Gather Decode Error occurs when the address request is targeted to an address that does not exist. This error is for Scatter / Gather Mode only (C_INCLUDE_SG = 1).

Note: Scatter Gather error bits are unable to be updated to the descriptor in remote memory. They are only captured in the associated channel DMASR where the error occurred.

Interconnect Parameters

The AXI DMA Interconnect Parameters are described in Table 3-16.

Table 3-16: **AXI DMA Interconnect Parameters**

Non-HDL Parameter	Allowable Values	Description
C_INTERCONNECT_M_AXI_MM2S_READ_ISSUING	1, 2, 4	This parameter sets the number of outstanding read requests the AXI Interconnect accepts from the AXI DMA MM2S Read Master. This parameter configures the AXI Interconnect slave port connected to the MM2S Read Master of the AXI DMA and is set automatically by the EDK Tool Suite.
C_INTERCONNECT_M_AXI_S2MM_WRITE_ISSUING	1, 2, 4	This parameter sets the number of outstanding write requests the AXI Interconnect accepts from the AXI DMA S2MM Write Master. This parameter configures the AXI Interconnect slave port connected to the S2MM Write Master of AXI DMA and is set automatically by the EDK Tool Suite.
C_INTERCONNECT_M_AXI_MM2S_READ_FIFO_DEPTH	0, 32, 512	This parameters sets the read data FIFO depth (in elements) for AXI DMA MM2S Read Master. This parameter configures the AXI Interconnect slave port connected to the MM2S Read Master of the AXI DMA and is set automatically by the EDK Tool Suite.
C_INTERCONNECT_M_AXI_S2MM_WRITE_FIFO_DEPTH	0, 32, 512	This parameters sets the write data First In First Out (FIFO) depth (in elements) for AXI DMA S2MM Write Master. This parameter configures the AXI Interconnect slave port connected to the S2MM Write Master of AXI DMA and is set automatically by the EDK Tool Suite.

Allowable Parameter Combinations

The AXI DMA Allowable Parameters Combinations are described in Table 3-17.

Table 3-17: **Allowable Parameter Combination**

Parameter Name	Affects Parameter	Relationship Description
C_INCLUDE_MM2S	C_INCLUDE_MM2S_DRE C_MM2S_BURST_SIZE, C_NUM_MM2S_CHANNELS	Affected Parameters are ignored when C_INCLUDE_MM2S = 0
C_INCLUDE_S2MM	C_INCLUDE_S2MM_DRE C_S2MM_BURST_SIZE, C_NUM_S2MM_CHANNELS	Affected Parameters are ignored when C_INCLUDE_S2MM = 0
C_INCLUDE_SG	C_SG_USE_STSAPP_LENGTH C_SG_INCLUDE_STSCNTRL_ STRM, C_ENABLE_MULTI_CHANNEL	Affected Parameters are ignored when C_INCLUDE_SG = 0
C_SG_INCLUDE_STSCNTRL_STRM	C_SG_USE_STSAPP_LENGTH	Affected Parameter is ignored when C_SG_INCLUDE_STSCNTRL_STRM = 0
C_M_AXI_MM2S_DATA_WIDTH	C_M_AXIS_MM2S_TDATA_ WIDTH	Affected Parameter must be less than or equal to C_M_AXI_MM2S_DATA_WIDTH
C_M_AXI_S2MM_DATA_WIDTH	C_S_AXIS_S2MM_TDATA_ WIDTH	Affected Parameter must be less than or equal to C_M_AXI_S2MM_DATA_WIDTH

SECTION II: VIVADO DESIGN SUITE

Customizing and Generating the Core

Constraining the Core

Detailed Example Design

Customizing and Generating the Core

The AXI DMA can be found in the following places in the Vivado™ IP Catalog.:

- **AXI_Infrastructure, Communication_&_Networking\Ethernet**
- **Embedded_Processing\AXI_Infrastructure\DMA**

To access the AXI DMA, do the following:

1. Open an existing project or create a new project using Vivado tools.
2. Open the IP catalog and navigate to any of the above taxonomies.
3. Double-click **AXI Direct Memory Access** to bring up the AXI DMA Graphical User Interface (GUI.)

Vivado System Parameter Screen

The AXI DMA GUI displays information about the core, allows configuration of the core, and provides the ability to generate the core. See Figure 4-1 and Figure 4-2.

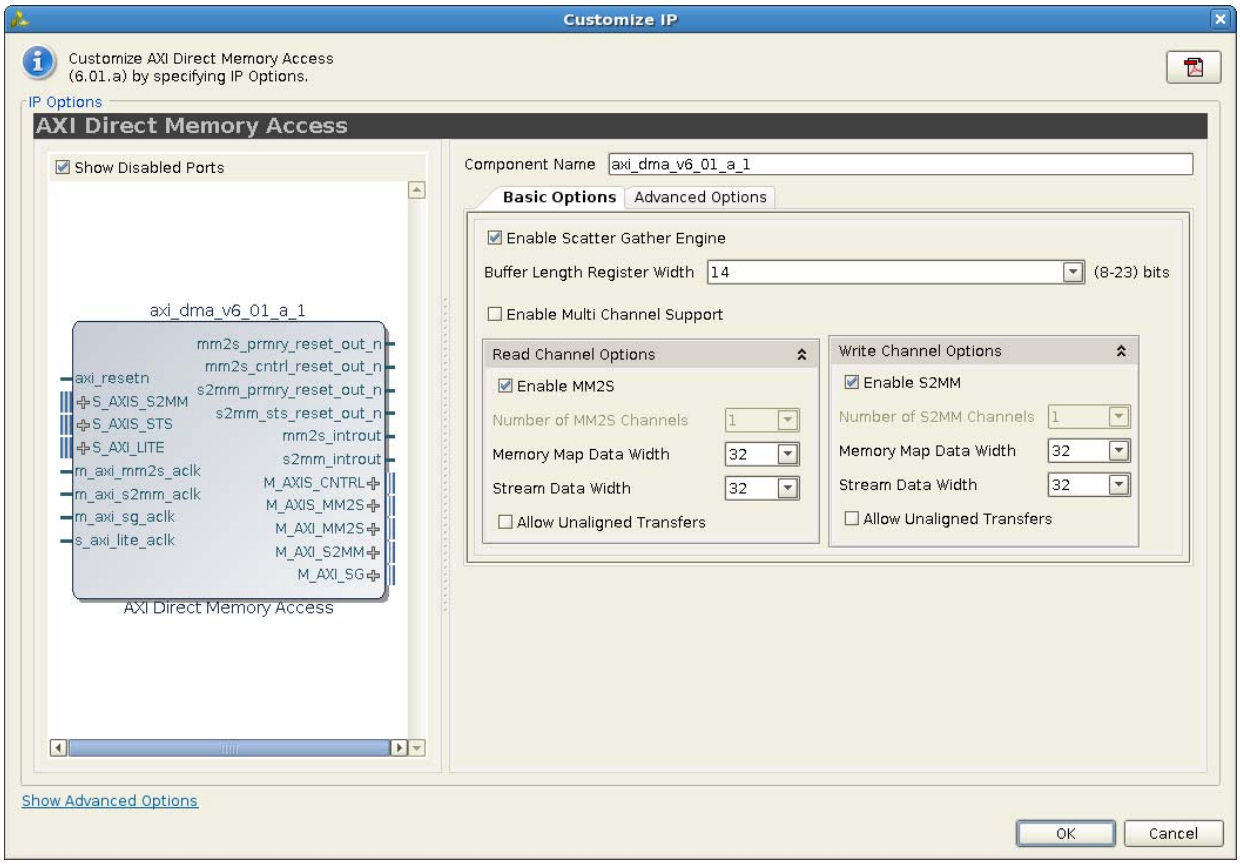

Figure 4-1: **Main GUI (Basic Options tab)**

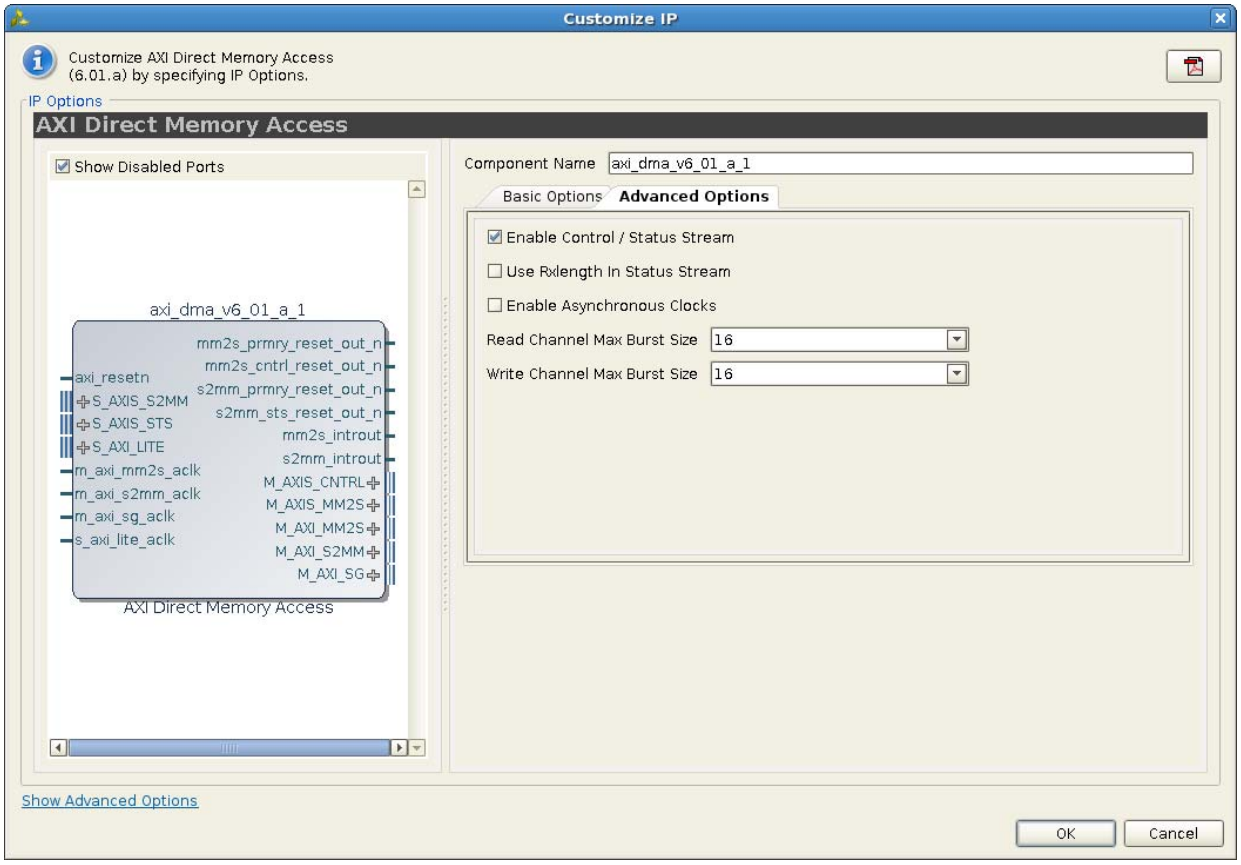

Figure 4-2: **Main GUI (Advanced Options tab)**

In most cases the DMA can be configured with the option specified on the **Basic Option** tab. All the options available in this GUI are essentially the same as that in the CORE Generator™ GUI. The GUI has been split into two tabs for the sake of simplicity and ease-of-use.

Component Name

The base name of the output files generated for the core. Names must begin with a letter and can be composed of any of the following characters: a to z, 0 to 9, and "_".

Basic Options

The following describe the fundamental options that affect both channels of the AXI DMA core.

Enable Scatter Gather Engine

Checking this option enables Scatter Gather Mode operation and includes the Scatter Gather Engine in AXI DMA. Unchecking this option enables Simple DMA Mode operation, excluding the Scatter Gather Engine from AXI DMA. Disabling the Scatter Gather Engine causes all output ports for the Scatter/Gather engine to be tied to zero and all of the input ports to be left open.

Enable Multi Channel DMA

Checking this option enables multichannel capability of DMA and lets you choose the number of channels for both MM2S and S2MM channels. See Multichannel DMA Support in Chapter 3 for details.

Buffer Length Register Width

This integer value specifies the number of valid bits used for the Control field buffer length and Status field bytes transferred in the Scatter/Gather descriptors. It also specifies the number of valid bits in the RX Length of Status Stream App4 field when Use Rxlength is enabled. For Simple DMA mode, it specifies the number of valid bits in the MM2S_LENGTH and S2MM_LENGTH registers. The length width directly correlates to the number of bytes being specified in a Scatter/Gather descriptor or number of bytes being specified in App4.RxLength, MM2S_LENGTH, or S2MM_LENGTH. The number of bytes is equal to 2Length Width. So a Length Width of 23 gives a byte count of 8388608 Bytes or 8 MegaBytes.

MM2S Channel Options

The following subsections describe options that affect only the MM2S Channel of the AXI Direct Memory Access (DMA) core.

Enable Channel

This option enables or disables the MM2S Channel. Enabling the MM2S Channel allows read transfers from memory to AXI4-Stream to occur. Disabling the MM2S Channel excludes the logic from the AXI DMA core. Outputs for MM2S channel are tied to zero and inputs are ignored by AXI DMA.

Number of MM2S Channels

This option enables you to choose a number of channels from 1 to 16.

Memory Map Data Width

Data width in bits of the AXI MM2S Memory Map Read data bus. Valid values are 32, 64, 128, 256, 512 and 1024.

Stream Data Width

Data width in bits of the AXI MM2S AXI4-Stream Data bus. This value must be equal or less than the Memory Map Data Width. Valid values are 8, 16, 32, 64, 128, 512 and 1024.

Allow Unaligned Transfers

Enables or disables the MM2S Data Realignment Engine (DRE). When checked, the DRE is enabled and allows data realignment to the byte (8 bits) level on the MM2S Memory Map datapath. For the MM2S channel, data is read from memory. If the DRE is enabled, data reads can start from any Buffer Address byte offset, and the read data is aligned such that the first byte read is the first valid byte out on the AXI4-Stream.

Note: If DRE is disabled for the respective channel, unaligned Buffer, Source, or Destination Addresses are not supported. Having an unaligned address with DRE disabled produces undefined results. DRE Support is only available for the AXI4-Stream data width setting of 64-bits and under.

S2MM Channel Options

The following describe options that affect only the S2MM Channel of the AXI DMA core.

Enable Channel

This setting enables or disables the S2MM Channel. Enabling the S2MM Channel allows write transfers from AXI4-Stream to memory to occur. Disabling the S2MM Channel excludes the logic from the AXI DMA core. Outputs for S2MM channel are tied to zero and inputs are ignored by AXI DMA.

Number of S2MM Channels

This option enables you to choose a number of channels from 1 to 16.

Memory Map Data Width

Data width in bits of the AXI S2MM Memory Map Write data bus. Valid values are 32, 64, 128, 256, 512 and 1024.

Stream Data Width

Data width in bits of the AXI S2MM AXI4-Stream Data bus. This value must be equal or less than the Memory Map Data Width. Valid values are 8, 16, 32, 64, 128, 512 and 1024.

Allow Unaligned Transfers

Enables or disables the S2MM Data Realignment Engine (DRE). When checked, the DRE is enabled and allows data realignment to the byte (8 bits) level on the S2MM Memory Map datapath. For the S2MM channel, data is written to memory. If the DRE is enabled, data writes can start from any Buffer Address byte offset, and the write data is aligned such that the first valid byte received on S2MM AXI4-Stream is written to the specified unaligned address offset.

Note: If DRE is disabled for the respective channel, unaligned Buffer, Source, or Destination Addresses are not supported. Having an unaligned address with DRE disabled produces undefined results. DRE Support is only available for AXI4-Stream data width setting of 64-bits and under

Advance Options

The following subsections describe some of the advance options that affect both channels of the AXI DMA core.

Enable Control / Status Stream

Checking this option enables the AXI4 Control and Status Streams. The AXI4 Control stream allows user application metadata associated with the MM2S channel to be transmitted to a target IP. User application fields 0 through 4 of an MM2S Scatter / Gather Start Of Frame (SOF) descriptor Transmit Start Of Frame (TXSOF =1) are transmitted on the `m_axis_mm2s_cntrl` stream interface along with an associated packet being transmitted on `m_axis_mm2s` stream interface. The AXI4 Status stream allows user application metadata associated with the S2MM channel to be received from a target IP. The received status packet populates user application fields 0 to 4 of an S2MM Scatter / Gather End Of Frame (EOF) descriptor. That is the descriptor associated with the end of packet. This is indicated by a Receive End Of Frame (RXEOF = 1) in the status word of the updated descriptor.

Note: This feature requires the inclusion of the Scatter Gather Engine.

Use RxLength In Status Stream

If the Control / Status Stream is enabled, checking this allows AXI DMA to use a receive length field that is supplied by the S2MM target IP in App4 of the status packet. This gives AXI DMA a pre-determined receive byte count, allowing AXI DMA to command the exact number of bytes to be transferred. This provides for a higher bandwidth solution for systems needing greater throughput. In this configuration the AXI DMA removes the S2MM store and forward feature so you must make sure the S2MM target IP can supply all data bytes specified in the receive length field of status packet APP4.

Enable Asynchronous Clocks

This setting provides the ability to operate the MM2S interface m_axi_mm2s_aclk, S2MM interface `m_axi_s2mm_aclk`, AXI4-Lite control interface `s_axi_lite_aclk`, and the Scatter Gather Interface `m_axi_sg_aclk` asynchronously from each other. When Asynchronous Clocks are enabled, the frequency of `s_axi_lite_aclk` must be less than or equal to `m_axi_sg_aclk`. Likewise `m_axi_sg_aclk` must be less than or equal to the slower of `m_axi_mm2s_aclk` and `m_axi_s2mm_aclk`. When Asynchronous Clocks are disabled, all clocks must be at the same frequency and from the same source.

Maximum Burst Size (MM2S and S2MM)

This setting specifies the maximum size of the burst cycles on the AXI4-Memory Map. In other words, this setting specifies the granularity of burst partitioning.

Valid values are 16, 32, 64, 128, and 256.

Output Generation (Vivado)

The output hierarchy when the core is generated from Vivado tool is as shown below:

Here the component name of the IP generated is axi_dma_v6_01_a_0.

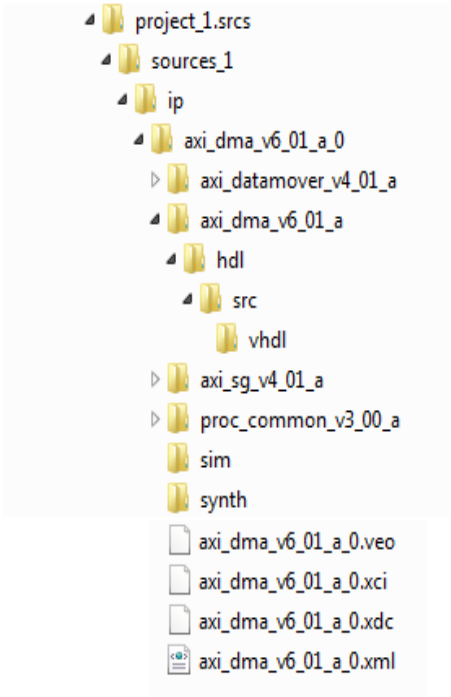

Figure 4-3: **Output Hierarchy**

Vivado tool delivers the RTL files of the axi_dma core as well as all its helper cores. The axi_datamover_v4_01_a, axi_sg_v4_01_a and proc_common_v3_00_a are the helper cores used by the axi_dma.

The RTL files of axi_dma are delivered under /ip/axi_dma_v6_01_a/hdl/src/vhdl. The sim and synth folders contain the wrappers for simulation and synthesis respectively. The tool also delivers the instantiation template file .veo/.vho and the xdc constraints file.

Constraining the Core

Necessary XDC constraints are delivered along with the core generation.

Detailed Example Design

The IP does not provide any example design.

SECTION III: ISE DESIGN SUITE

Customizing and Generating the Core

Constraining the Core

Detailed Example Design

Customizing and Generating the Core

This chapter includes information on using Xilinx tools to customize and generate the core.

Generating the Core

The AXI DMA can be found in **AXI_Infrastructure** and also in **Communication_&_Networking\Ethernet** in the CORE Generator™ graphical user interface (GUI) View by Function pane.

To access the AXI DMA, do the following:

1. Open a project by selecting **File** then **Open Project** or create a new project by selecting **File** then **New Project**.
2. With an open project, choose **AXI_Infrastructure** in the **View by Function** pane.
3. Double-click on **AXI Direct Memory Access** to bring up the AXI DMA GUI.

CORE Generator System Parameter Screen

The AXI DMA GUI contains one screen (Figure 7-1) providing information about the core, allows for configuration of the core, and provides the ability to generate the core.

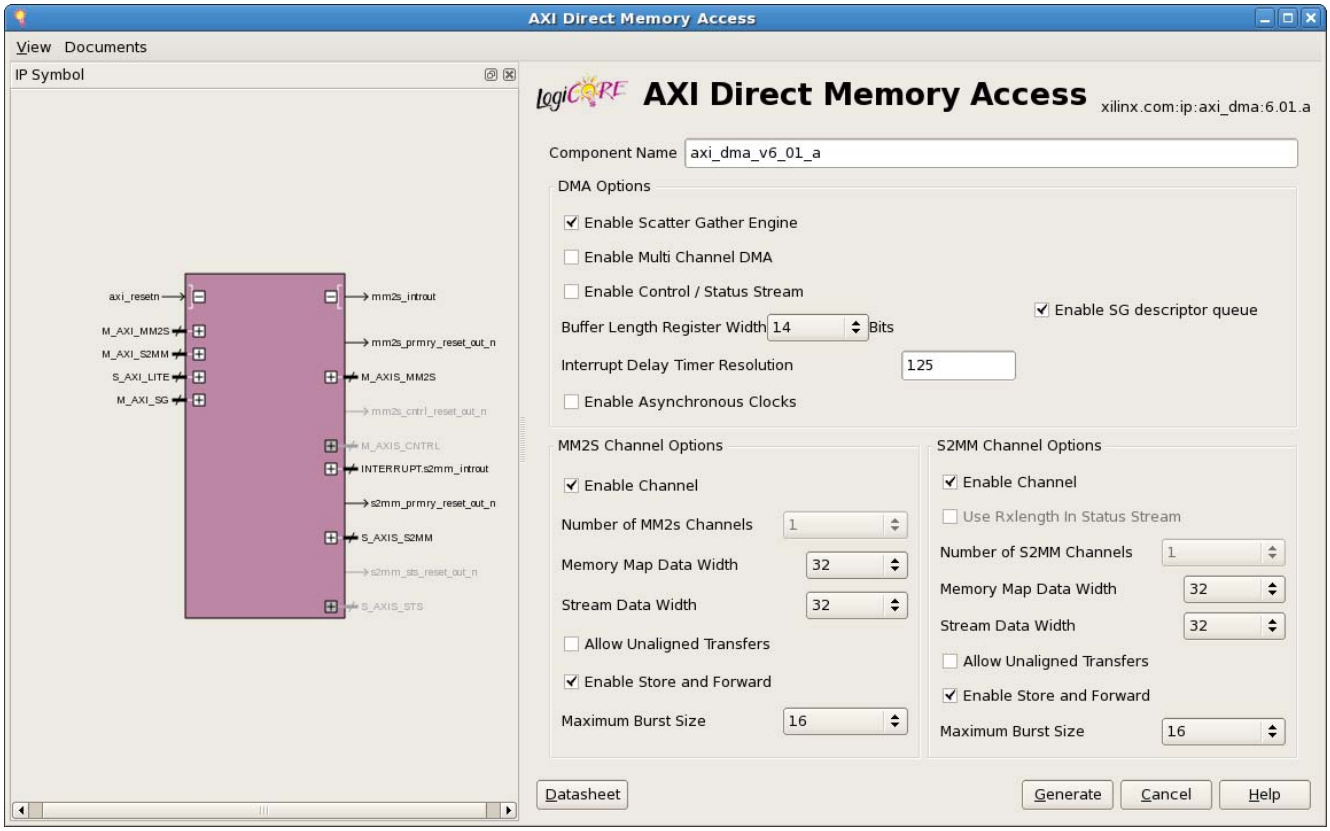

Figure 7-1: **AXI DMA GUI**

Component Name

The base name of the output files generated for the core. Names must begin with a letter and can be composed of any of the following characters: a to z, 0 to 9, and "_".

DMA Options

The following describe options that affect both channels of the AXI DMA core.

Enable Scatter Gather Engine

Checking this option enables Scatter Gather Mode operation and includes the Scatter Gather Engine in AXI DMA. Unchecking this option enables Simple DMA Mode operation, excluding the Scatter Gather Engine from AXI DMA. Disabling the Scatter Gather Engine causes all output ports for the Scatter/Gather engine to be tied to zero, and all of the input ports to be left open.

Enable Multi Channel DMA

Checking this option enables multichannel capability of DMA and lets you choose the number of channels for both MM2S and S2MM channels. See Multichannel DMA Support in Chapter 3 for details.

Enable Control / Status Stream

Checking this option enables the AXI4 Control and Status Streams. The AXI4 Control stream allows user application metadata associated with the MM2S channel to be transmitted to a target IP. User application fields 0 through 4 of an MM2S Scatter / Gather Start Of Frame (SOF) descriptor Transmit Start Of Frame (TXSOF =1) are transmitted on the `m_axis_mm2s_cntrl` stream interface along with an associated packet being transmitted on `m_axis_mm2s` stream interface. The AXI4 Status stream allows user application metadata associated with the S2MM channel to be received from a target IP. The received status packet populates user application fields 0 to 4 of an S2MM Scatter / Gather End Of Frame (EOF) descriptor. That is the descriptor associated with the end of packet. This is indicated by a Receive End Of Frame (RXEOF = 1) in the status word of the updated descriptor.

Note: This feature requires the inclusion of the Scatter Gather Engine.

Enable SG Descriptor Queue

Checking this option enables the queue in Scatter/Gather module so a set of descriptors can be fetched and stored in queue.

Buffer Length Register Width

This integer value specifies the number of valid bits used for the Control field buffer length and Status field bytes transferred in the Scatter / Gather descriptors. It also specifies the number of valid bits in the RX Length of Status Stream App4 field when Use Rxlength is enabled. For Simple DMA mode it specifies the number of valid bits in the MM2S_LENGTH and S2MM_LENGTH registers. The length width directly correlates to the number of bytes being specified in a Scatter/Gather descriptor or number of bytes being specified in App4.RxLength, MM2S_LENGTH, or S2MM_LENGTH. The number of bytes is equal to $2^{\text{Length Width}}$. So a Length Width of 23 gives a byte count of 8388608 Bytes or 8 MegaBytes.

Interrupt Delay Timer Resolution

This integer value sets the resolution of the Interrupt Delay Counter. Values specify the number of clock cycles between each tick of the delay counter. If Scatter Gather Engine is enabled then clock cycles are based on the `m_axi_sg_aclk` clock input. When in Simple DMA Mode, that is Scatter Gather Engine is disabled, then the delay interrupt is not used by AXI DMA.

Enable Asynchronous Clocks

This setting provides the ability to operate the MM2S interface `m_axi_mm2s_aclk`, S2MM interface `m_axi_s2mm_aclk`, AXI4-Lite control interface `s_axi_lite_aclk`, and the Scatter Gather Interface `m_axi_sg_aclk` asynchronously from each other. When Asynchronous Clocks are enabled, the frequency of `s_axi_lite_aclk` must be less than or equal to `m_axi_sg_aclk`. Likewise `m_axi_sg_aclk` must be less than or equal to the slower of `m_axi_mm2s_aclk` and `m_axi_s2mm_aclk`. When Asynchronous Clocks are disabled, all clocks must be at the same frequency and from the same source.

MM2S Channel Options

The following describe options that affect only the MM2S Channel of the AXI Direct Memory Access (DMA) core.

Enable Channel

This option enables or disables the MM2S Channel. Enabling the MM2S Channel allows read transfers from memory to AXI4-Stream to occur. Disabling the MM2S Channel excludes the logic from the AXI DMA core. Outputs for MM2S channel are tied to zero and inputs are ignored by AXI DMA.

Number of MM2S Channels

This option enables you to choose a number of channels from 1 to 16.

Memory Map Data Width

Data width in bits of the AXI MM2S Memory Map Read data bus. Valid values are 32, 64, 128, 256, 512 and 1024.

Stream Data Width

Data width in bits of the AXI MM2S AXI4-Stream Data bus. This value must be equal or less than the Memory Map Data Width. Valid values are 8, 16, 32, 64, 128, 512 and 1024.

Allow Unaligned Transfers

Enables or disables the MM2S Data Realignment Engine (DRE). When checked, the DRE is enabled and allows data realignment to the byte (8 bits) level on the MM2S Memory Map datapath. For the MM2S channel, data is read from memory. If the DRE is enabled, data reads can start from any Buffer Address byte offset, and the read data is aligned such that the first byte read is the first valid byte out on the AXI4-Stream. What is considered aligned or unaligned is based on the stream data width C_M_AXIS_MM2S_TDATA_WIDTH.

For example, if C_M_AXIS_MM2S_TDATA_WIDTH = 32, data is aligned if it is located at word offsets (32-bit offset), that is 0x0, 0x4, 0x8, 0xC, and so forth. If C_M_AXIS_MM2S_TDATA_WIDTH = 64, data is aligned if it is located at double-word offsets (64-bit offsets), that is 0x0, 0x8, 0x10, 0x18, and so forth.

For use cases where all transfers are C_M_AXIS_MM2S_TDATA_WIDTH aligned, DRE can be disabled excluding DRE and saving FPGA resources.

Note: If DRE is disabled for the respective channel, unaligned Buffer, Source, or Destination Addresses are not supported. Having an unaligned address with DRE disabled produces undefined results. DRE Support is only available for AXI4-Stream data width setting of 64-bits and under.

Maximum Burst Size

This option specifies the maximum size of the burst cycles on the AXI4 MM2S Memory Map Read interface. In other words, this setting specifies the granularity of burst partitioning. For example, if the burst length is set to 16, the maximum burst on the memory map interface is 16 data beats. Smaller values reduce throughput whereas larger values increase throughput. Valid values are 16, 32, 64, 128, and 256.

S2MM Channel Options

The following describe options that affect only the S2MM Channel of the AXI DMA core.

Enable Channel

This setting enables or disables the S2MM Channel. Enabling the S2MM Channel allows write transfers from AXI4-Stream to memory to occur. Disabling the S2MM Channel excludes the logic from AXI DMA core. Outputs for S2MM channel are tied to zero and inputs are ignored by AXI DMA.

Number of S2MM Channels

This option enables you to choose a number of channels from 1 to 16.

Use RxLength In Status Stream

If the Control / Status Stream is enabled, then checking this allows AXI DMA to use a receive length field that is supplied by the S2MM target IP in App4 of the status packet. This gives AXI DMA a pre-determined receive byte count allowing AXI DMA to command the exact number of bytes to be transferred. This provides for a higher bandwidth solution for systems needing greater throughput. In this configuration the AXI DMA removes the S2MM store and forward feature so you must make sure the S2MM target IP can supply all data bytes specified in the receive length field of status packet APP4.

Clock Frequency

This setting specifies the clock frequency in hertz of the S2MM interface clock, `m_axi_s2mm_aclk`. This parameter is used when Asynchronous Clocks are enabled and configures the AXI DMA for proper clock domain crossings. When Asynchronous Clocks are disabled, this setting is ignored by AXI DMA.

Memory Map Data Width

Data width in bits of the AXI S2MM Memory Map Write data bus. Valid values are 32, 64, 128, 256, 512 and 1024.

Stream Data Width

Data width in bits of the AXI S2MM AXI4-Stream Data bus. This value must be equal or less than the Memory Map Data Width. Valid values are 8, 16, 32, 64, 128, 512 and 1024.

Allow Unaligned Transfers

Enables or disables the S2MM Data Realignment Engine (DRE). When checked, the DRE is enabled and allows data realignment to the byte (8 bits) level on the S2MM Memory Map datapath. For the S2MM channel, data is written to memory. If the DRE is enabled then data writes can start from any Buffer Address byte offset, and the write data is aligned such that the first valid byte received on S2MM AXI4-Stream is written to the specified unaligned address offset. What is considered aligned or unaligned is based on the stream data width C_S_AXIS_S2MM_TDATA_WIDTH. For example, if C_S_AXIS_S2MM_TDATA_WIDTH = 32, data is aligned if it is located at word offsets (32-bit offset), that is, 0x0, 0x4, 0x8, 0xC, and so forth.

If C_S_AXIS_S2MM_TDATA_WIDTH = 64, data is aligned if it is located at double-word offsets (64-bit offsets), that is, 0x0, 0x8, 0x10, 0x18, and so forth.

For use cases where all transfers are C_S_AXIS_S2MM_TDATA_WIDTH aligned, DRE can be disabled excluding DRE and saving FPGA resources.

Note: If DRE is disabled for the respective channel, unaligned Buffer, Source, or Destination Addresses are not supported. Having an unaligned address with DRE disabled produces undefined results. DRE Support is only available for AXI4-Stream data width setting of 64-bits and under.

Maximum Burst Size

This setting specifies the maximum size of the burst cycles on the AXI S2MM Memory Map Write interface. In other words, this setting specifies the granularity of burst partitioning. For example, if the burst length is set to 16, the maximum burst on the memory map interface is 16 data beats. Smaller values reduce throughput whereas larger values increase throughput. Valid values are 16, 32, 64, 128, and 256.

Generating the Core Using EDK

The AXI DMA can be found in **IP Catalog - EDK_Install/DMA and Timer** in the Xilinx Platform Studio tool graphical user interface (GUI).

To access the AXI DMA, do the following:

1. Invoke Xilinx Platform Studio and open a project by selecting **File > Open Project** or create a new project by selecting **File > New Project**.
2. With an open project, choose **EDK_Install/DMA and Timer**.
3. Double-click **AXI DMA** to display the AXI DMA GUI.

For a new project:

1. Invoke Xilinx Platform Studio and create a New Project using Base System Builder.
2. Select Interconnect Type (AXI System) and then select Board Name (based on 6/7 series FPGA)
3. After BSB is created, add AXI DMA from IP catalog (**EDK_Install/DMA and Timer**) by double clicking **AXI DMA 6.01.a**. This opens up an EDK GUI which is described in the next section.

EDK pCore GUI

The AXI VDMA EDK GUI provides information about the core, allows for configuration of the core, and provides the ability to generate the core. The pCore is generated with each option set to the default value. Figure 7-2 illustrates the EDK pCore GUI for the AXI DMA. All of the options in the EDK pCore GUI correspond to the same options in the CORE Generator™ tool GUI. EDK pCore GUI has an additional option, Interface a custom IP. Enable this checkbox to allow DMA connect to any generic protocol.

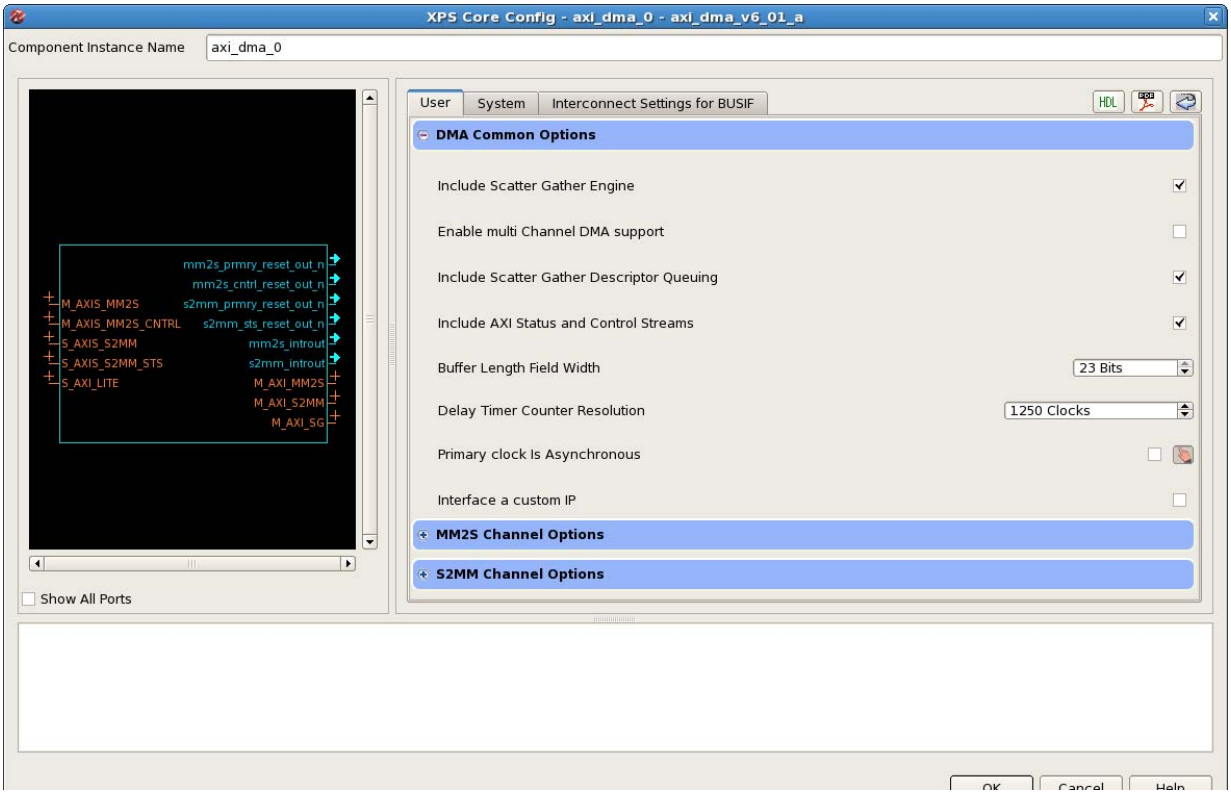

Figure 7-2: **EDK pCore GUI**

Core Implementation

Functional Simulation

VHDL source files for axi_dma_v6_01_a are provided un-encrypted for use in behavioral simulation within a simulation environment. Neither a test bench nor test fixture is provided with the AXI DMA core.

Synthesis

Synthesis of the AXI DMA can be performed with Xilinx Synthesis Technology (XST) or with the Vivado synthesis tool.

Xilinx Tools

See the LogiCORE™ IP Facts table.

Static Timing Analysis

Static timing analysis can be performed using *trce*, following *ngdbuild*, *map*, and *par*.

Output Generation

Figure 7-3 shows the output hierarchy when the core is generated from the CORE Generator tool.

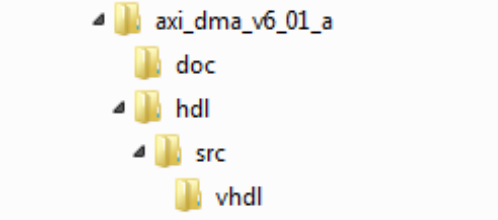

Figure 7-3: **Hierarchy**

- \doc contains readme.txt, changelog.html, vinfo.html, product guide and other files.
- \vhdl contains hdl rtl files

Constraining the Core

UCF constraints as applicable are generated along with the other core deliverables.

Detailed Example Design

This core does not provide any example design.

SECTION IV: APPENDICES

Migrating

Debugging

Additional Resources

Appendix A

Migrating

See *Vivado Design Suite Migration Methodology Guide* (UG911).

Also see the Vivado Design Suite - 2012.2 User Guides web page.

Appendix B

Debugging

See Solution Centers in Appendix C for information helpful to the debugging progress.

Appendix C

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

http://www.xilinx.com/support.

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

References

The *AMBA AXI4-Stream Protocol Specification* provides supplemental material useful with this product guide.

To search for Xilinx documentation, go to http://www.xilinx.com/support

For more information about Vivado™ tools, see Vivado Design Suite - 2012.2 User Guides web page.

Technical Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/19/11	1.0	Initial Xilinx release.
04/24/12	2.0	Summary of Core Changes • Added multichannel support • Added 2D transactions support • Added keyhole support • Added Cache and User controls for AXI memory side Interface
07/25/12	3.0	Summary of Core Changes • Added Vivado tools support and Zynq-7000 support

Notice of Disclaimer