AXI4 DMA Subsystem

Product Specification

Version 2.0
December 15, 2025

Abstract

The AXI4 DMA Subsystem is a high-performance, single-channel Direct Memory Access (DMA) controller designed for high-bandwidth data movement between memory regions without CPU intervention. It features an AXI4-Lite control plane and a high-throughput AXI4 data plane with strict protocol compliance and safety mechanisms.

Designer: Aritra Manna

Contents

1 Overview

The **AXI4 DMA Subsystem** is a high-performance, single-channel Direct Memory Access (DMA) controller. It bridges an AXI4-Lite control plane with a high-bandwidth AXI4 data plane to move data between memory regions without CPU intervention.

1.1 Key Features

- **High Performance**: AXI4 Master with 128-bit data path, Single-cycle throughput (100%).
- **Robust Architecture**: Store-and-Forward mechanism for data integrity and deadlock avoidance.
- **Strict Compliance**: Enforces 4KB boundary checks and 16-byte alignment.
- **Safety**: Independent Source/Destination Watchdog Timers.
- **Control**: Simple AXI4-Lite Slave interface with Status/Error reporting.
- **Interrupts**: Configurable interrupt support for Completion and Error events.
- **Elastic Buffering**: Integrated 4KB FWFT FIFO with skid buffer for maximum bandwidth.

1.2 Block Diagram

Listing 1: Architecture Block Diagram

```
                              +------------------------------------+
                              |         axi_dma_subsystem          |
                              |                                    |
[AXI4-Lite Slave]             |   +----------------------------+   |
----------------------------->|-->|       dma_reg_block        |   |
(Configuration)               |   |                            |   |
                              |   |  [Registers]               |   |
                              |   |  STS, CTRL, SRC, DST, LEN  |   |
                              |   +-------------+--------------+   |
                              |                 |                  |
                              |         (Core Control IF)          |
                              |    Start, Done, Addrs, Status      |
                              |                 v                  |
                              |   +----------------------------+   |
                              |   |      axi_dma_master        |   |
[AXI4 Master]                 |   |                            |   |
<============================>|-->|  [FSM] [Validation Logic]  |   |
(Data Movement)               |   |  [Watchdogs]               |   |
                              |   +-------------+--------------+   |
                              |                 |                  |
                              |          (FIFO Interface)          |
                              |         Push/Pop, Data             |
                              |                 v                  |
                              |   +----------------------------+   |
                              |   |      fifo_bram_fwft        |   |
                              |   |                            |   |
                              |   |  [BRAM] + [Skid Buffer]    |   |
                              |   +----------------------------+   |
                              +------------------------------------+
```

2 Module Specifications

2.1 Top-Level Module: axi_dma_subsystem

This wrapper module integrates the register block and the DMA core.

2.1.1 Parameters

Parameter	Default	Description
AXI_ADDR_W	32	Width of AXI addresses.
AXI_DATA_W	128	Width of AXI data path (Master).
AXI_ID_W	4	Width of AXI ID signals.
FIFO_DEPTH	256	Depth of internal buffer (256 * 128b = 4KB).
TIMEOUT_SRC	100000	Cycles before source read times out.
TIMEOUT_DST	100000	Cycles before destination write times out.

2.1.2 Ports

- **Clock/Reset**: clk, rst_n.
- **AXI Slave**: cfg_s_axi_* (32-bit data).
- **AXI Master**: m_axi_* (128-bit data).
- **Interrupt**: intr_pend (Active High).

2.1.3 Reset Semantics

On de-assertion of rst_n (Active Low):

1. All AXI VALID outputs must de-assert immediately/asynchronously.
2. The internal FSM returns to IDLE.
3. FIFO contents are invalidated (pointers reset).
4. No AXI completion is reported (no spurious DONE/ERROR).
5. STATUS registers reset to default values.

2.2 Sub-Module: dma_reg_block

Handles the AXI4-Lite Slave interface, maintains Configuration/Status registers, and generates the Interrupt. It synchronizes control signals to the core.

2.2.1 Parameters

Parameter	Default	Description
AXI_ADDR_W	32	Width of AXI addresses.

2.2.2 Ports

Port Name	Dir	Width	Description
clk, rst_n	In	1	System Clock/Reset.
AXI4-Lite Slave Interface			
cfg_s_axi_*	In/Out	-	Standard AXI4-Lite Slave Interface.
Core Control Interface			
core_start	Out	1	Pulse. Asserts for 1 cycle when CTRL[0] is written.
core_src_addr	Out	32	Static value from SRC_ADDR register.
core_dst_addr	Out	32	Static value from DST_ADDR register.
core_len	Out	32	Static value from LEN register.
core_done	In	1	Pulse. Indicates transfer completion.
core_busy	In	1	Level. 1=Core is active. Mapped to STATUS[1].
core_status	In	4	Error Code. Valid when core_done is high.
Interrupt Interface			
intr_pend	Out	1	(sts_done || sts_error) && ctrl_int_en. Active High.

2.3 Sub-Module: axi_dma_master

The brain of the operation. Contains the Main FSM, Validation Logic, and AXI Master protocol handlers.

2.3.1 AXI4 Master Interface Ports

Signal Name	Dir	Width	Description
Read Address Channel (AR)			
m_axi_arid	Output	4	Read Address ID.
m_axi_araddr	Output	32	Read Address.
m_axi_arlen	Output	8	Burst Length (0-255).
m_axi_arsize	Output	3	Burst Size (0x4 = 16 bytes).
m_axi_arburst	Output	2	Burst Type (01 = INCR).
m_axi_arvalid	Output	1	Read Address Valid.
m_axi_arready	Input	1	Read Address Ready.
Read Data Channel (R)			
m_axi_rid	Input	4	Read ID (Must match ARID).
m_axi_rdata	Input	128	Read Data.
m_axi_rresp	Input	2	Read Response.
m_axi_rlast	Input	1	Read Last Beat.
m_axi_rvalid	Input	1	Read Data Valid.
m_axi_rready	Output	1	Read Data Ready.
Write Address Channel (AW)			
m_axi_awid	Output	4	Write Address ID.
m_axi_awaddr	Output	32	Write Address.
m_axi_awlen	Output	8	Burst Length.

Signal Name	Dir	Width	Description
`m_axi_awsize`	Output	3	Burst Size.
`m_axi_awburst`	Output	2	Burst Type (01 = INCR).
`m_axi_awvalid`	Output	1	Write Address Valid.
`m_axi_awready`	Input	1	Write Address Ready.
Write Data Channel (W)			
`m_axi_wdata`	Output	128	Write Data.
`m_axi_wstrb`	Output	16	Write Strobes.
`m_axi_wlast`	Output	1	Write Last Beat.
`m_axi_wvalid`	Output	1	Write Data Valid.
`m_axi_wready`	Input	1	Write Data Ready.
Write Response Channel (B)			
`m_axi_bid`	Input	4	Write Response ID.
`m_axi_bresp`	Input	2	Write Response.
`m_axi_bvalid`	Input	1	Write Response Valid.
`m_axi_bready`	Output	1	Write Response Ready.

2.3.2 Functional Description

1. **Transfer Coordination**: The core must wait for a **Start Pulse** (`dma_start`) while in the Idle state. Upon receiving a start command, it must capture and **validate configurations** (`SRC`, `DST`, `LEN`). If validation passes, the core must autonomously orchestrate the data movement in a **Store-and-Forward** manner:
   - **Read Phase**: Issue AXI Read command and buffer the entire burst into the internal FIFO.
   - **Write Phase**: Once the read burst is complete and data is secured, issue the AXI Write command to drain the FIFO to the destination.
2. **Exact Burst Formation**: For a valid transfer, `LEN` must be a multiple of `AXI_DATA_W/8` (16 bytes). The DMA always issues exactly one full-length INCR burst where: `ARLEN = AWLEN = (LEN / 16) - 1`.
3. **Watchdog Timer**: Two counters (`src_timer`, `dst_timer`) increment when `VALID=1 && READY=0`. If counter > `TIMEOUT_CYCLES`, buffer aborts to `DONE` with `ERR_TIMEOUT`. Registers reset on any successful handshake.

2.4 Sub-Module: `fifo_bram_fwft`

A specialized FIFO designed for high-bandwidth bursting. It uses a “Skid Buffer” (Pipeline Register) on the output to break timing paths and ensure First-Word Fall-Through (FWFT) behavior.

2.4.1 Parameters

Parameter	Default	Description
`DATA_W`	128	Width of data port.
`DEPTH`	1024	FIFO Depth.

2.4.2 Ports

Port Name	Dir	Width	Description
clk, rst_n	In	1	System Clock/Reset.
wr_en	In	1	Write Enable.
din	In	128	Write Data.
rd_en	In	1	Read Enable (Pop).
full	Out	1	Full Status.
dout	Out	128	Read Data.
empty	Out	1	Empty Status.

3 Register Map

Base Address: Defined by system interconnect (e.g. 0x4000_0000).

Offset	Register	Access	Reset	Bits	Description
0x04	**CTRL**	RW	0x0	1 0	INT_EN: 1=Enable Interrupts. START: Write 1 to start transfer. (Self-clearing).
0x08	**STATUS**	MIX	0x0	7:4 3 2 1 0	ERR_CODE (RO): Last error code. INTR_VAL (RO): Live interrupt status. ERROR (W1C): 1=Transfer Failed. Write 1 to clear. BUSY (RO): 1=DMA Active. DONE (W1C): 1=Transfer Success. Write 1 to clear.
0x0C	**SRC_ADDR**	RW	0x0	31:0	Source Address. **Must be 16-byte aligned.**
0x10	**DST_ADDR**	RW	0x0	31:0	Destination Address. **Must be 16-byte aligned.**
0x14	**LEN**	RW	0x0	31:0	Length in bytes. **Must be 16-byte aligned.** Max 4096.

4 Error Codes

Values read from STATUS[7:4].

Hex	Name	Description
0	ERR_NONE	No error.
1	ERR_ALIGN_SRC	SRC_ADDR[3:0] != 0.
2	ERR_ALIGN_DST	DST_ADDR[3:0] != 0.
3	ERR_ALIGN_LEN	LEN[3:0] != 0.
4	ERR_ZERO_LEN	LEN == 0.
5	ERR_4K_SRC	Source address range crosses 4KB boundary.
6	ERR_4K_DST	Destination address range crosses 4KB boundary.
7	ERR_LEN_LARGE	LEN > 4096.

Hex	Name	Description
8	ERR_TIMEOUT_SRC	Source AXI Read Stalled > TIMEOUT **consecutive** cycles.
9	ERR_TIMEOUT_DST	Destination AXI Write Stalled > TIMEOUT **consecutive** cycles.
F	ERR_AXI_RESP	AXI Slave returned SLVERR or DECERR.

5 Interrupt Architecture

The subsystem provides a single level-sensitive interrupt output (intr_pend).

5.1 Sources

The interrupt is asserted when **either** of the following sticky bits in the STATUS register are set:

1. DONE (Bit 0): Asserted on successful completion.
2. ERROR (Bit 2): Asserted on any error condition (ERR_CODE != 0).

5.2 Masking

The intr_pend output is qualified by the Global Interrupt Enable bit (CTRL[1]). It is asserted active high if and only if:

1. The Global Interrupt Enable bit (CTRL[1]) is set to **1**, **AND**
2. At least one of the sticky status bits is set to **1**.

5.3 Clearance (W1C)

The interrupt is **Active High** and **Level Sensitive**.

1. Read STATUS register to determine the cause.
2. Write 1 to the respective bit (STATUS[0] or STATUS[2]) to clear it.
3. The intr_pend line de-asserts immediately when both bits are zero.