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, safety mechanisms, and integrated 4KB FWFT FIFO with skid buffer for maximum bandwidth.

RTL Designer: Aritra Manna

Contents

1 Overview

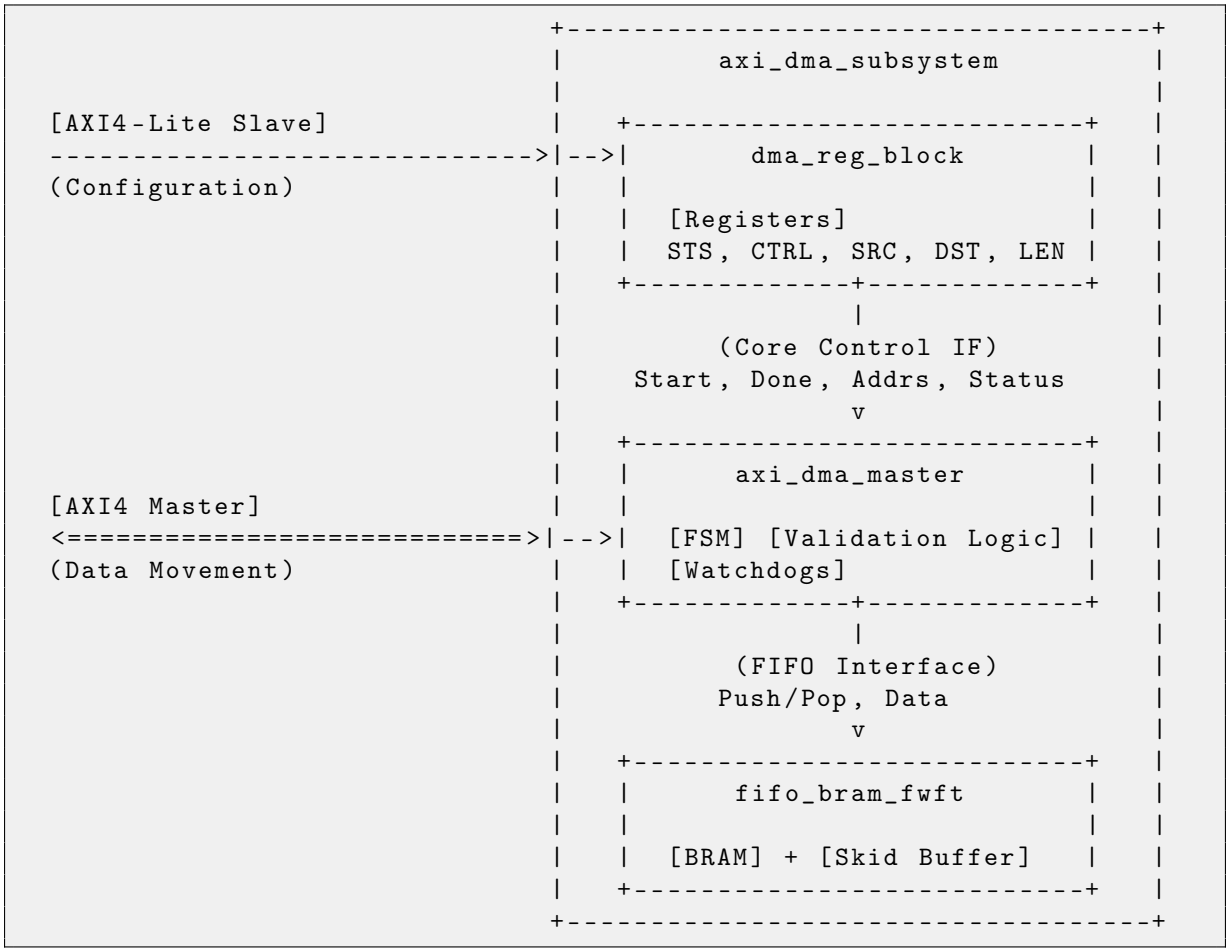
The **AXI4 DMA Subsystem** is a high-performance, single-channel Direct Memory Access (DMA) controller. It bridges an AXI4-Lite control plane with a high-bandwidth AXI4 data plane to move data between memory regions without CPU intervention.

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 Architecture Block Diagram

Listing 1: Architecture Block Diagram



2 Top-Level Module: axi_dma_subsystem

Wrapper module that integrates the register block and the DMA core.

2.1 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.

2.2 Ports

Clock and Reset

Port Name	Dir	Width	Description
clk	In	1	System Clock.
rst_n	In	1	Active-Low Reset.

AXI4-Lite Slave Interface (Configuration)

Port Name	Dir	Width	Description
cfg_s_axi_awaddr	In	32	Write Address.
cfg_s_axi_awvalid	In	1	Write Address Valid.
cfg_s_axi_awready	Out	1	Write Address Ready.
cfg_s_axi_wdata	In	32	Write Data.
cfg_s_axi_wstrb	In	4	Write Strobes.
cfg_s_axi_wvalid	In	1	Write Data Valid.
cfg_s_axi_wready	Out	1	Write Data Ready.
cfg_s_axi_bresp	Out	2	Write Response.
cfg_s_axi_bvalid	Out	1	Write Response Valid.
cfg_s_axi_bready	In	1	Write Response Ready.
cfg_s_axi_araddr	In	32	Read Address.
cfg_s_axi_arvalid	In	1	Read Address Valid.
cfg_s_axi_arready	Out	1	Read Address Ready.
cfg_s_axi_rdata	Out	32	Read Data.
cfg_s_axi_rresp	Out	2	Read Response.
cfg_s_axi_rvalid	Out	1	Read Data Valid.
cfg_s_axi_rready	In	1	Read Data Ready.

AXI4 Master Interface (Data Movement)

Used for DMA transfers. Defaults: AXI_ID_W=4, AXI_DATA_W=128.

Port Name	Dir	Width	Description
Read Address Channel (AR)			
m_axi_arid	Out	4	Read Address ID.

Port Name	Dir	Width	Description
m_axi_araddr	Out	32	Read Address.
m_axi_arlen	Out	8	Read Burst Length.
m_axi_arsize	Out	3	Read Burst Size.
m_axi_arburst	Out	2	Read Burst Type.
m_axi_arvalid	Out	1	Read Address Valid.
m_axi_arready	In	1	Read Address Ready.
Read Data Channel (R)			
m_axi_rid	In	4	Read ID.
m_axi_rdata	In	128	Read Data.
m_axi_rresp	In	2	Read Response.
m_axi_rlast	In	1	Read Last Beat.
m_axi_rvalid	In	1	Read Data Valid.
m_axi_rready	Out	1	Read Data Ready.
Write Address Channel (AW)			
m_axi_awid	Out	4	Write Address ID.
m_axi_awaddr	Out	32	Write Address.
m_axi_awlen	Out	8	Write Burst Length.
m_axi_awsize	Out	3	Write Burst Size.
m_axi_awburst	Out	2	Write Burst Type.
m_axi_awvalid	Out	1	Write Address Valid.
m_axi_awready	In	1	Write Address Ready.
Write Data Channel (W)			
m_axi_wdata	Out	128	Write Data.
m_axi_wstrb	Out	16	Write Strobes.
m_axi_wlast	Out	1	Write Last Beat.
m_axi_wvalid	Out	1	Write Data Valid.
m_axi_wready	In	1	Write Data Ready.
Write Response Channel (B)			
m_axi_bid	In	4	Write Response ID.
m_axi_bresp	In	2	Write Response.
m_axi_bvalid	In	1	Write Response Valid.
m_axi_bready	Out	1	Write Response Ready.

Transaction Ordering & Sidebands

1. **Ordering**: The DMA core issues at most one outstanding AXI Read transaction and one outstanding AXI Write transaction at any time.
2. **ID Usage**: All transfers use a fixed AXI ID per channel. The core assumes in-order responses and checks IDs strictly.
3. **Sidebands**: All AXI sideband signals not listed (CACHE, PROT, LOCK, QOS) are tied to constant, implementation-defined safe values (typically 0).

Interrupt Output

Port Name	Dir	Width	Description
intr_pend	Out	1	Interrupt Pending (Active High).

2.3 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.

3 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.

3.1 Parameters

Parameter	Default	Description
`AXI_ADDR_W`	32	Width of AXI addresses.

3.2 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.

3.3 Functional Requirements

1. **Register Decode**: The module must decode the defined address space (0x04 to 0x14) and return `SLVERR` response for any access to undefined addresses.

2. **Sticky Status**: Status bits (DONE/ERROR) must remain set until explicitly cleared by software (Write-1-to-Clear).
3. **Start Logic**:
   - A write to the `START` bit must generate a single-cycle start pulse to the core.
   - **Re-arm Protection**: A new START command is accepted only when `intr_pend` is 0 (i.e., previous DONE/ERROR must be cleared).
   - If a START is written while `intr_pend == 1`, the command is ignored.

4 Sub-Module: axi_dma_master

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

4.1 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	Internal FIFO depth (matched to 4KB).
TIMEOUT_SRC_CYCLES	128	Source Read Timeout cycles (Default internal).
TIMEOUT_DST_CYCLES	128	Source Write Timeout cycles (Default internal).

4.2 Ports

Port Name	Dir	Width	Description
clk, rst_n	In	1	System Clock/Reset.
DMA Control Interface			
dma_start	In	1	1-cycle Start Pulse.
dma_src_addr	In	32	Source Address.
dma_dst_addr	In	32	Destination Address.
dma_length	In	32	Length in bytes.
dma_done	Out	1	Completion Pulse.
dma_completion_status	Out	4	Error code (0=OK). Valid on dma_done.
dma_busy	Out	1	1 when State != IDLE.

AXI4 Master Interface

The `axi_dma_master` module uses `axi_*` **prefix** for its AXI ports. The wrapper `axi_dma_subsystem` connects these to its external `m_axi_*` ports.

Port Name	Dir	Width	Description
Read Address Channel (AR)			
axi_arid	Out	4	Read Address ID.

Port Name	Dir	Width	Description
axi_araddr	Out	32	Read Address.
axi_arlen	Out	8	Burst Length (0-255).
axi_arsize	Out	3	Burst Size (0x4 = 16 bytes).
axi_arburst	Out	2	Burst Type (01 = INCR).
axi_arvalid	Out	1	Read Address Valid.
axi_arready	In	1	Read Address Ready.
Read Data Channel (R)			
axi_rid	In	4	Read ID (Must match ARID).
axi_rdata	In	128	Read Data.
axi_rresp	In	2	Read Response.
axi_rlast	In	1	Read Last Beat.
axi_rvalid	In	1	Read Data Valid.
axi_rready	Out	1	Read Data Ready.
Write Address Channel (AW)			
axi_awid	Out	4	Write Address ID.
axi_awaddr	Out	32	Write Address.
axi_awlen	Out	8	Burst Length.
axi_awsize	Out	3	Burst Size.
axi_awburst	Out	2	Burst Type (01 = INCR).
axi_awvalid	Out	1	Write Address Valid.
axi_awready	In	1	Write Address Ready.
Write Data Channel (W)			
axi_wdata	Out	128	Write Data.
axi_wstrb	Out	16	Write Strobes (Always All-Ones).
axi_wlast	Out	1	Write Last Beat.
axi_wvalid	Out	1	Write Data Valid.
axi_wready	In	1	Write Data Ready.
Write Response Channel (B)			
axi_bid	In	4	Write Response ID.
axi_bresp	In	2	Write Response.
axi_bvalid	In	1	Write Response Valid.
axi_bready	Out	1	Write Response Ready.

4.3 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 independent counters (`src_timer`, `dst_timer`) increment when `VALID=1 && READY=0`.
    - **Reset Condition**: Resets to 0 on any successful handshake (`VALID=1 && READY=1`) OR any FSM state change.
    - **Timeout**: If counter > `TIMEOUT_CYCLES`, buffer aborts to `DONE` with `ERR_TIMEOUT`.
4. **FIFO Control**: `RD_DATA` state drives `fifo_wr_en`. `WR_DATA` state drives `fifo_rd_en` based on `wready`.
5. **FIFO Soft-Reset**: When the FSM reaches the `DONE` state (either on successful completion or error), the FIFO must be flushed/soft-reset. This ensures any stale or incomplete data from timeout conditions, AXI errors, or aborted transfers is discarded.

5 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.

5.1 Parameters

Parameter	Default	Description
`DATA_W`	128	Width of data port (Must match `AXI_DATA_W`).
`DEPTH`	1024	FIFO Depth (Number of items).

5.2 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 (includes BRAM + Skid).
`dout`	Out	128	Read Data (Available immediately if !empty).
`empty`	Out	1	Empty Status (0 = Data valid on `dout`).

5.3 Functional Requirements

1. **Buffering**: The module must provide elastic buffering to decouple the Source Read rate from the Destination Write rate.
2. **First-Word Fall-Through (FWFT)**: The FIFO must present valid data on the output port (`dout`) immediately when available, without waiting for a read request (`rd_en`). This is critical for maximizing AXI Write channel bandwidth.
3. **Backpressure**: It must correctly assert `full` to prevent overflow and `empty` to indicate data availability.

4. **Throughput**: The design must support continuous back-to-back read/write cycles (100% throughput) when not empty/full.

6 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.

7 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 (Hardware does not split).
6	`ERR_4K_DST`	Destination address range crosses 4KB boundary (Hardware does not split).
7	`ERR_LEN_LARGE`	`LEN > 4096`.
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` (0x2) or `DECERR` (0x3).

8 Interrupt Architecture

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

8.1 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`).

8.2 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 (`STATUS.DONE` or `STATUS.ERROR`) is set to **1**.

8.3 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.

9 Design Guarantees & Assumptions

1. **AXI-Lite Timing**: The Slave interface may exert backpressure (`AWREADY/WREADY/ARREADY` low). Software must not assume single-cycle completion for register accesses.
2. **Performance Contract**: The AXI Master interface is required to support **1 transfer per clock cycle (100% throughput)** during active bursts to meet bandwidth expectations.
3. **Reset Observability**: Reset is asynchronous. Software observing the core via JTAG/Debug during a reset event will see `BUSY` drop to 0 immediately. `DONE` and `ERROR` pulses are strictly suppressed during reset to prevent false completion reports.