AXI4-DMA

AXI4 to/from AXI-Stream DMA

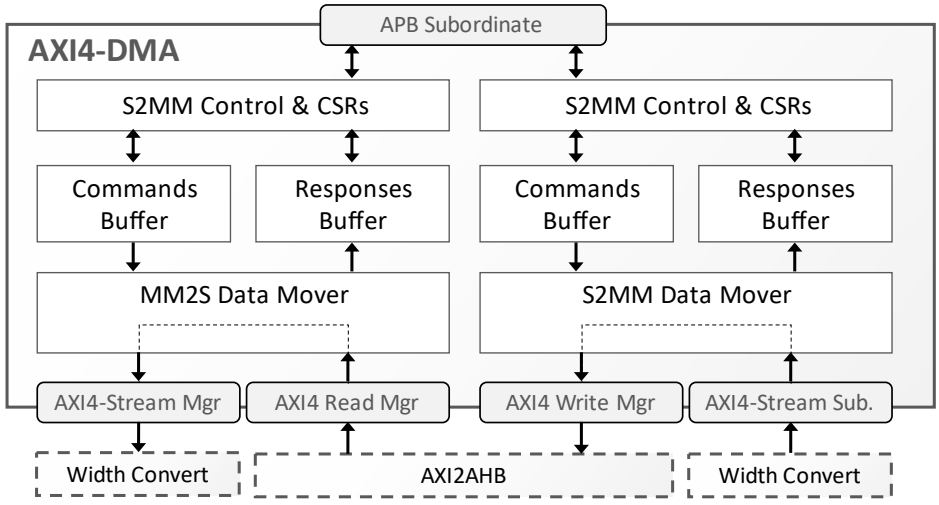
The AXI4-DMA IP core implements a Direct Memory Access (DMA) engine that efficiently moves data between AXI4-Stream peripherals and a memory-mapped AXI4 bus.

The core implements two independent paths: One transfers data from the read manager memory-mapped interface to the manager stream (MM2S) interface. The other transfers data from the subordinate stream interface to the write manager memory-map (S2MM) interface. Each path consists of a data mover block, the control-and-status registers, and buffers for DMA commands and responses. The host system can program the commands and receives responses via the control and status registers (CSR) interface. The address offset for each DMA command is either 32- or 64-bits wide, and the length of the transfer is 16 bits, allowing up to 64kB to be transferred with a single command. The MM2S commands control the behavior of the TLAST and TUSER signals of the manager stream interface, while TLAST on the subordinate stream interface causes an early termination of the transaction. The data-movers use the response descriptors to report possible errors and the actual transfer length for the S2MM path.

Designed for ease of integration, the AXI4-DMA supports configuration options and optional modules allowing the adoption of the core to the requirements of each specific design. The data-bus and address-bus widths of the AXI4 interfaces and the data-bus width of the AXI4-stream interfaces are synthesis-time configurable and can be 32 or 64 bits. External AXI-Stream data width converters can connect to peripherals with wider or narrower data-buses, The core is delivered with such sample modules converting from 32 bits to 64 bits and vice-versa. Furthermore, integration with a 32-bit AHB bus is possible with the AXI-to-AHB bridge that is also included.

The AXI4-DMA core is rigorously verified, LINT-clean, and scan-ready. It is available in synthesizable Verilog and FPGA netlist forms and includes everything required for successful implementation, including a UVM testbench, simulation and synthesis scripts, and comprehensive user documentation.

Block Diagram



Applications

The AXI4-DMA core can be used in any SoC integrating streaming-capable peripherals that need to receive input or store outputs in the system memory. The core is especially suited as a companion to the CAN bus controller and TSN Ethernet cores available from CAST.

FEATURES

Memory-Mapped to Stream & Stream to Memory-Mapped DMA

- Independent stream-to-memory, and memory-to-stream paths
- Up to 16 commands per direction with 32-bit or 64-bit address offset and 16-bit transfer length

Interfaces

- Host (Memory-Mapped) Data Interface
  - AXI4 Manager port used to access data and descriptors
  - Run-time programmable burst size can be limited to 16 beats for AXI3 compliance.
  - Aligned accesses only
  - Stream manager's TLAST and TUSER behavior controlled by the DMA command
- Peripheral (Stream) Data Interface
  - AXI4-Stream subordinate port for memory-to-stream, and manager port for stream-to-memory
  - Synthesis-time configurable data-bus width (32 or 64 bits)
  - Stream subordinate's TLAST causes early transaction termination
  - S2MM channel reports TLAST assertions and actual transfer length
- Management/CSR Interface
  - APB subordinate port
  - Provides access to control and status registers and DMA command & response buffers
- Maskable interrupt notifies host for completion of each command
- AXI4-to-AHB bridge can optionally be instantiated (supports only 32-bit data & address width)

Synthesis-Time Configuration Options

- Data-bus width (32 or 64 bits)
- Address-bus width (32 or 64 bits)
- Memory-to-stream, and stream-to-memory FIFO depth
- Memory-to-stream and stream-to-memory path instantiation

Support

The AXI4-DMA as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Deliverables

The core includes everything required for successful implementation:

- RTL (System Verilog) source code or targeted FPGA netlist
- System Verilog Testbench
- Sample Simulation and Synthesis scripts
- Comprehensive Documentation