HY220
Εργαστήριο Ψηφιακών Κυκλωμάτων

Χειμερινό Εξάμηνο
2017-2018

Interconnects: AXI Protocol

AXI

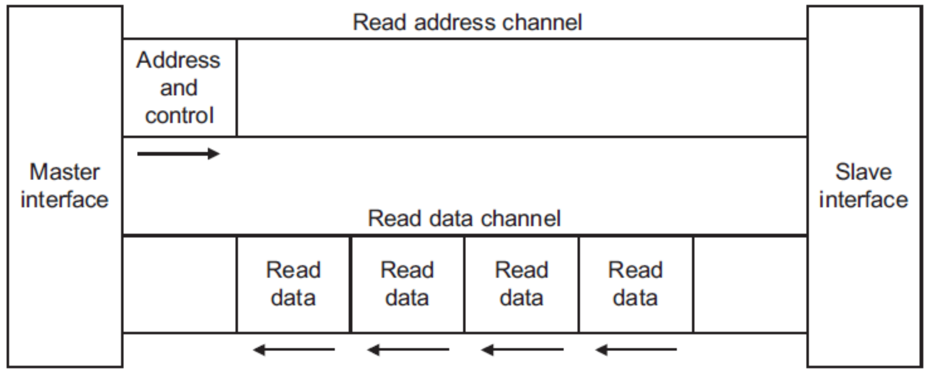
- AMBA AXI protocol is targeted at high-performance, high-frequency system designs

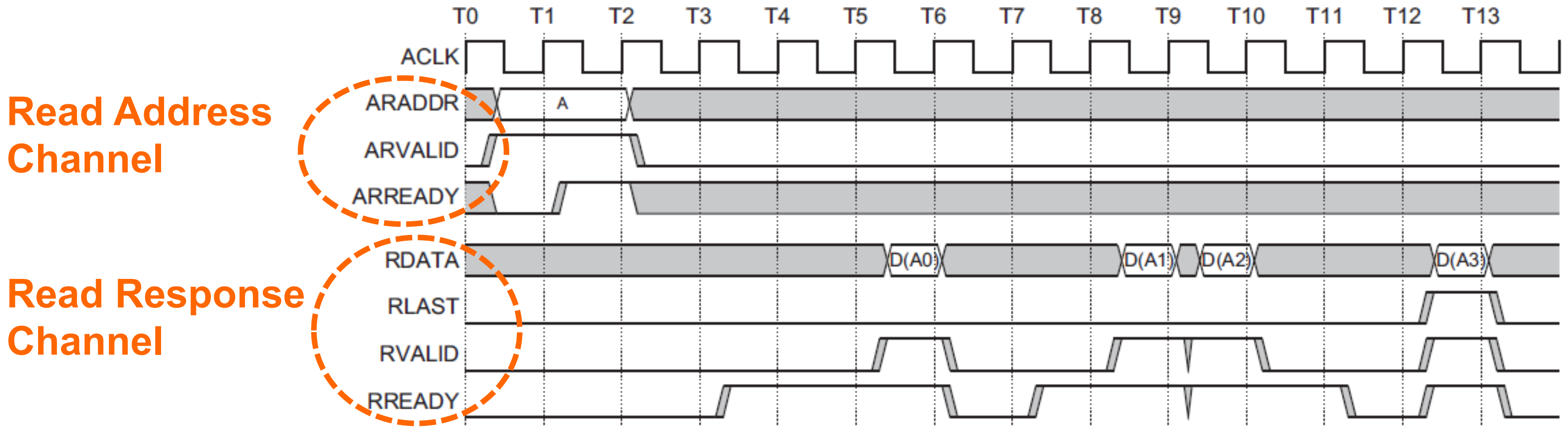
- AXI key features
  - Separate address/control and data phases
  - Support for unaligned data transfers using byte strobes
  - Separate read and write data channels to enable low-cost Direct Memory Access (DMA)
  - Ability to issue multiple outstanding addresses
  - Out-of-order transaction completion
  - Easy addition of register stages to provide timing closure

5 Independent Channels

- Read address channel and Write address channel
  - Variable length burst: 1 ~ 16 data transfers
  - Burst with a transfer size of 8 ~ 1024 bits (1B ~ 1KB)
- Read data channel
  - Convey data and any read response info.
  - Data bus can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits
- Write data channel
  - Data bus can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits
- Write response channel
  - Write response info.

AXI Read Operation





RREADY: From master, indicate that master can accept the read data and response info.

Sources: ARM and Xilinx

AXI Write Operation

Write Address Channel

Write Data Channel

Write Response Channel

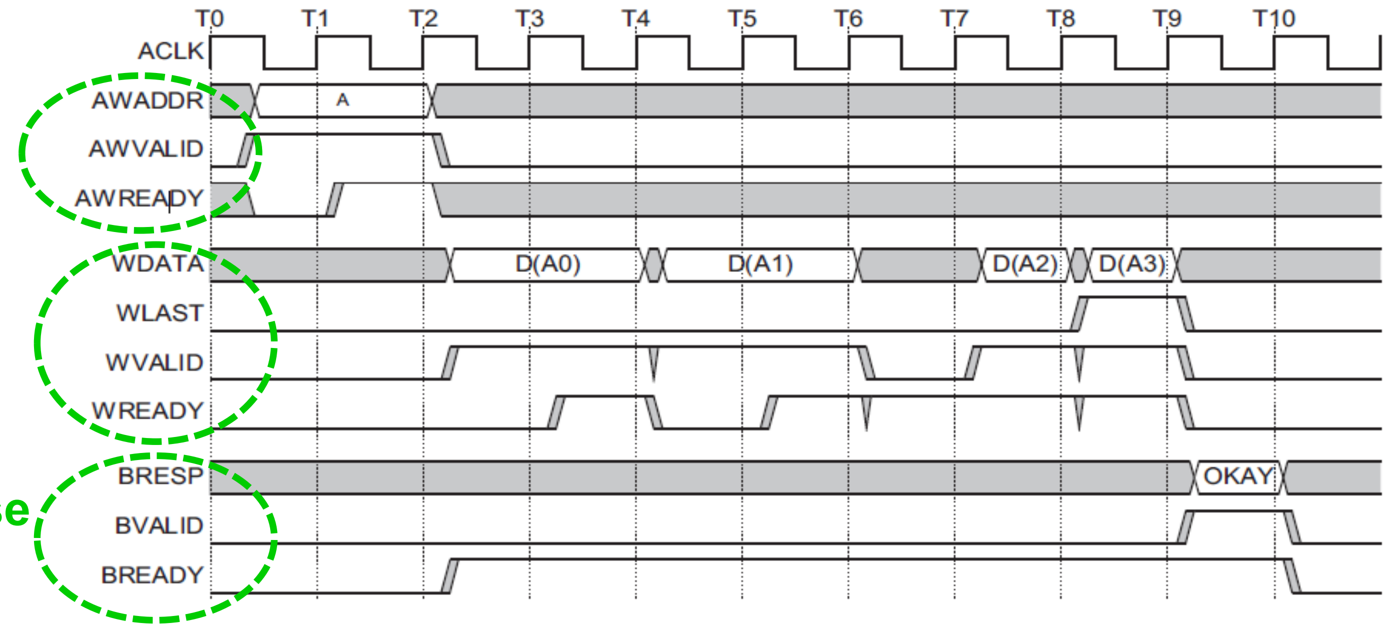


Figure 1-6 Write burst

WVALID Source: Master

WREADY Source: Slave

BVALID Source: Slave

BREADY Source: Master

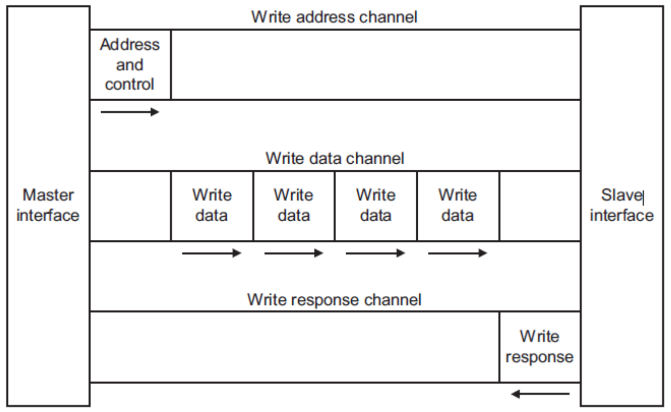


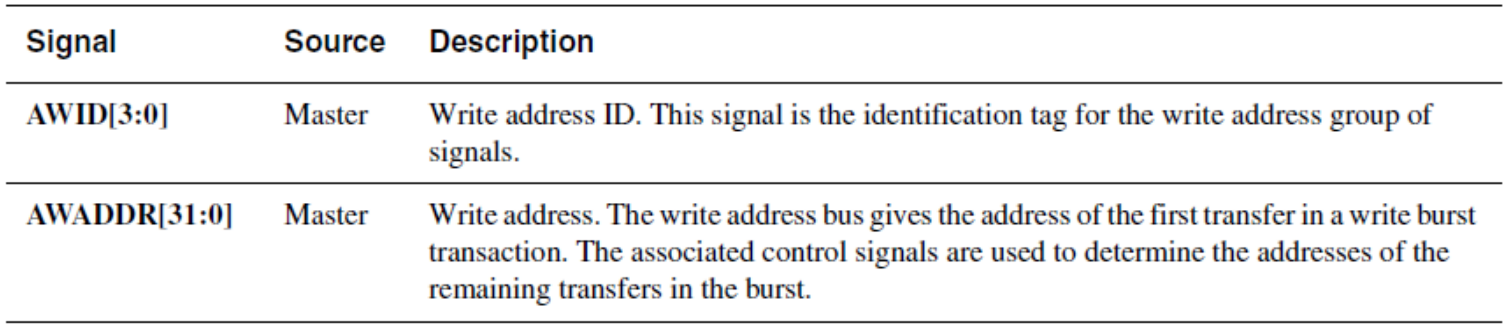
Figure 1-2 Channel architecture of writes

Sources: ARM and Xilinx

Out-of-order Completion

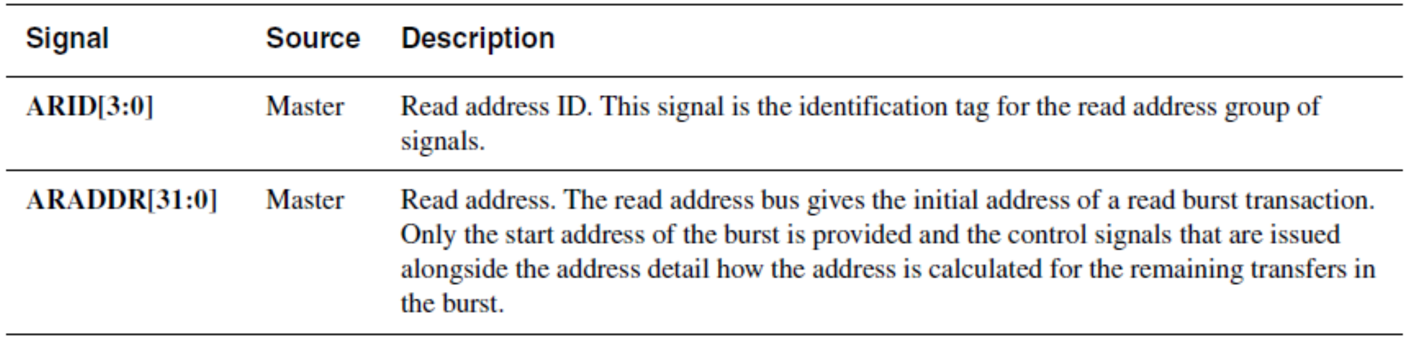
- AXI gives an ID tag to every transaction
  - Transactions with the same ID are completed in order
  - Transactions with different IDs can be completed out of order

Table 2-2 Write address channel signals



Signal	Source	Description
AWID[3:0]	Master	Write address ID. This signal is the identification tag for the write address group of signals.
AWADDR[31:0]	Master	Write address. The write address bus gives the address of the first transfer in a write burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst.

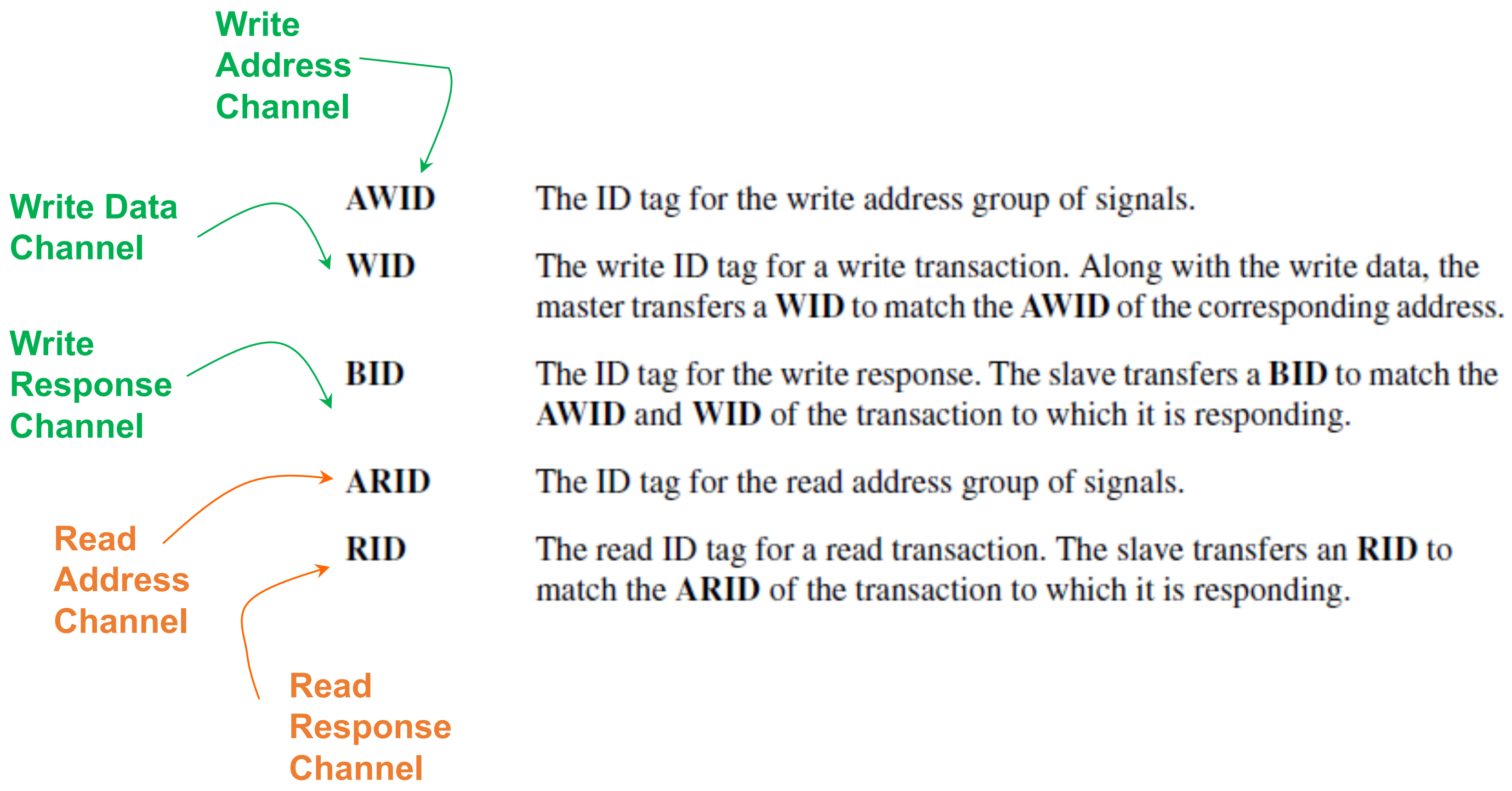
Table 2-5 Read address channel signals



Signal	Source	Description
ARID[3:0]	Master	Read address ID. This signal is the identification tag for the read address group of signals.
ARADDR[31:0]	Master	Read address. The read address bus gives the initial address of a read burst transaction. Only the start address of the burst is provided and the control signals that are issued alongside the address detail how the address is calculated for the remaining transfers in the burst.

Sources: ARM and Xilinx

ID Signals


Write Address Channel
Write Data Channel
Write Response Channel
Read Address Channel
Read Response Channel

AWID	The ID tag for the write address group of signals.
WID	The write ID tag for a write transaction. Along with the write data, the master transfers a **WID** to match the **AWID** of the corresponding address.
BID	The ID tag for the write response. The slave transfers a **BID** to match the **AWID** and **WID** of the transaction to which it is responding.
ARID	The ID tag for the read address group of signals.
RID	The read ID tag for a read transaction. The slave transfers an **RID** to match the **ARID** of the transaction to which it is responding.

Sources: ARM and Xilinx

Out-of-order Completion

- Out-of-order transactions can improve system performance in 2 ways
  - Fast-responding slaves respond in advance of earlier transactions with slower slaves
  - Complex slaves can return data out of order
    - A data item for a later access might be available before the data for an earlier access is available

- If a master requires that transactions are completed in the same order that they are issued, they must all have the same ID tag

- It is not a required feature
  - Simple masters and slaves can process one transaction at a time in the order they are issued

Addition of Register Slices

- AXI enables the insertion of a register slice in any channel at the cost of an additional cycle latency
  - Trade-off between latency and maximum frequency

- It can be advantageous to use
  - Direct and fast connection between a processor and high-performance memory
  - Simple register slices to isolate a longer path to less performance-critical peripherals

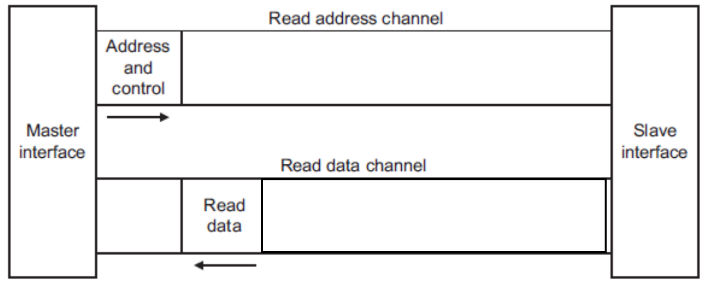
AXI-Lite

- The AXI4-Lite interface is a subset of the AXI4 interface intended for communication with control registers in components
- The aim of AXI4-Lite is to allow simple component interfaces to be built that are smaller and also require less design and validation effort
- Having a defined subset of the full AXI4 interface allows many different components to be built using the same subset and also allows a single common conversion component to be used to move between AXI4 and AXI4-Lite interfaces

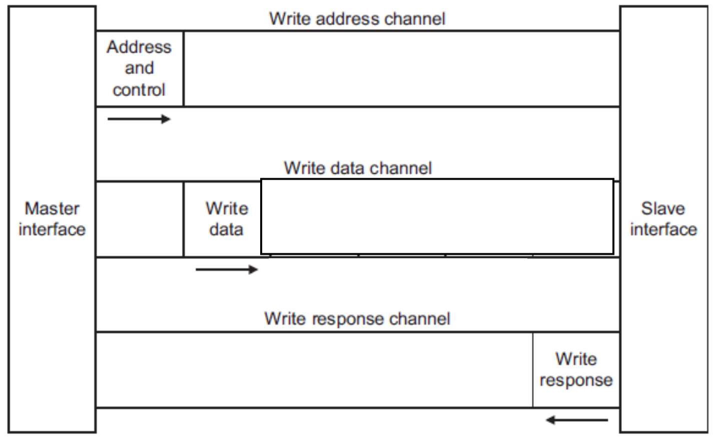
Sources: ARM and Xilinx

AXI-Lite

- No burst
- Data width 32 or 64 only
- Simple "logic" to connect AXI4 master to AXI4-Lite slave
  - Reflect master's transaction ID
- This is best for simple systems with minimal peripherals



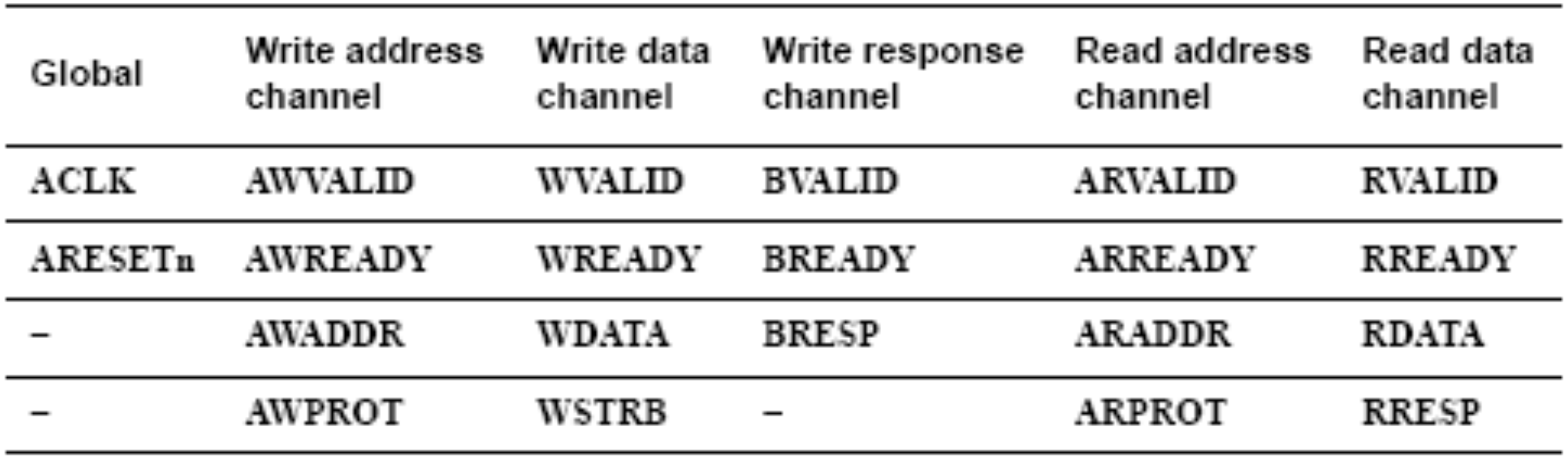
AXI4-Lite Read



AXI4-Lite Write

AXI-Lite Signal list

- Subset of AXI signal set
- Simple traditional signaling
- Targeted applications: simple, low-performance peripherals
  - GPIO
  - Uart Lite

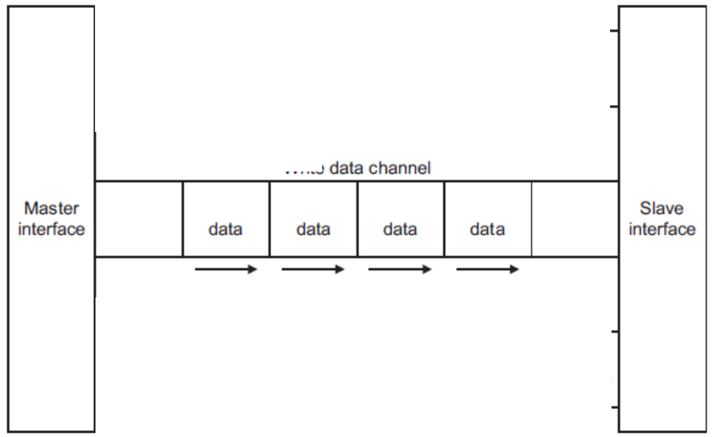


Global	Write address channel	Write data channel	Write response channel	Read address channel	Read data channel
ACLK	AWVALID	WVALID	BVALID	ARVALID	RVALID
ARESETn	AWREADY	WREADY	BREADY	ARREADY	RREADY
–	AWADDR	WDATA	BRESP	ARADDR	RDATA
–	AWPROT	WSTRB	–	ARPROT	RRESP

Sources: ARM and Xilinx

AXI-Stream

- No address channel
- Not read and write, always master to slave
- Unlimited burst length



AXI4-Streaming Transfer

AXI Additional Features

- ID fields for each of the five channels facilitate overlapped transactions
  - Provides for a transaction tag
- Transaction burst type determines address bus behavior
  - Fixed, increment, or wrap
- Optional address **Lock** signals facilitates exclusive and atomic access protection
- System cache support
- Protection unit support
- Error support
- Unaligned address

Documentation

- ARM specifications
  - AMBA AXI Protocol Version 2.0
  - AMBA 4 AXI4-Stream Protocol Version 1.0
  - http://infocenter.arm.com/help/topic/com.arm.doc.set.amba
- Xilinx AXI Reference Guide, UG761
  - AXI Usage in Xilinx FPGAs
    - Introduce key concepts of the AXI protocol
    - Explains what features of AXI Xilinx has adopted