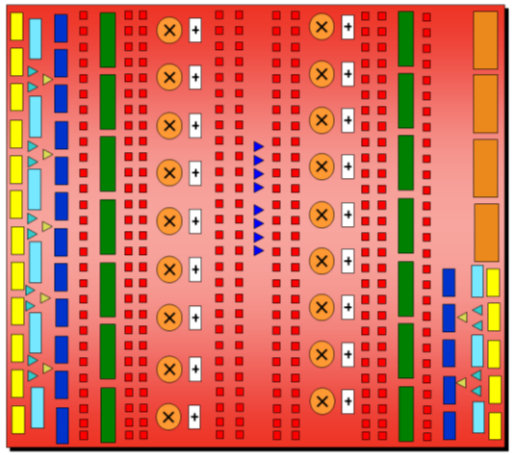
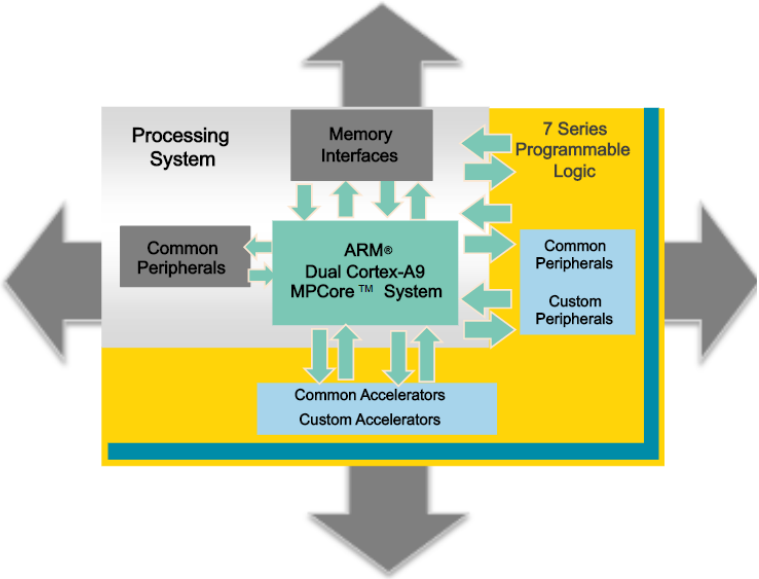
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

A2A
Algorithms to Architecture

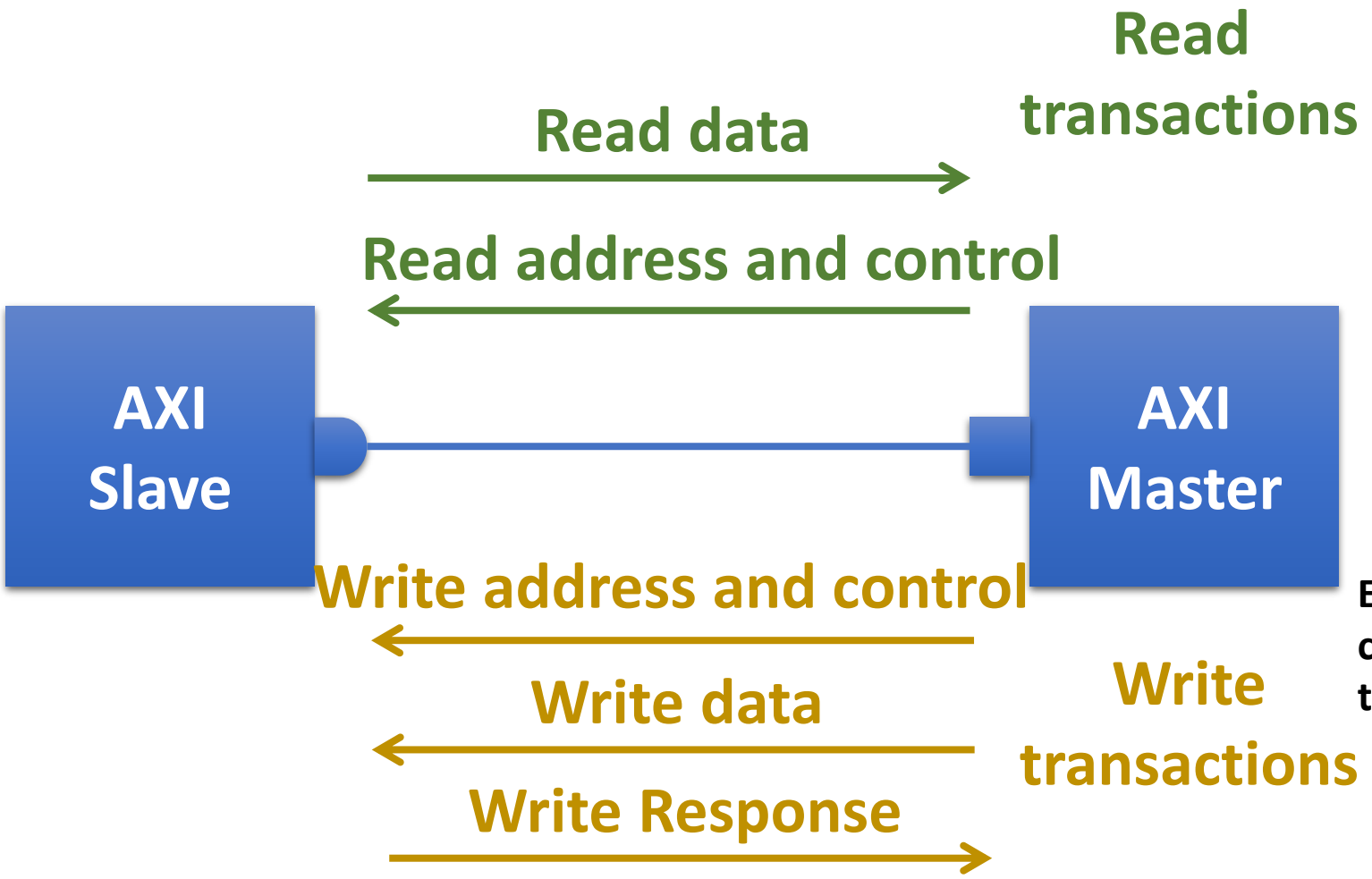
ECE 270: Embedded Logic Design





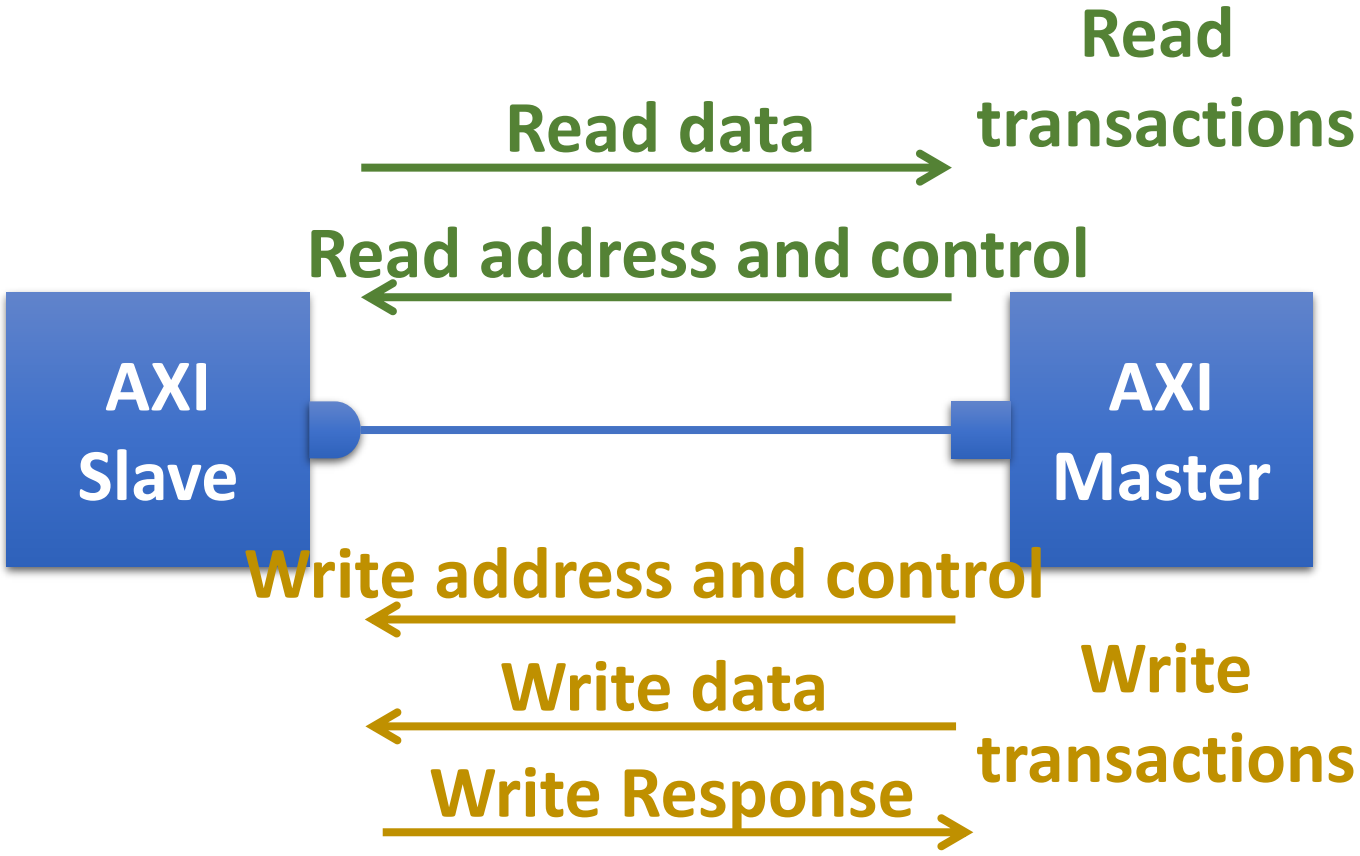
AXI Memory Mapped

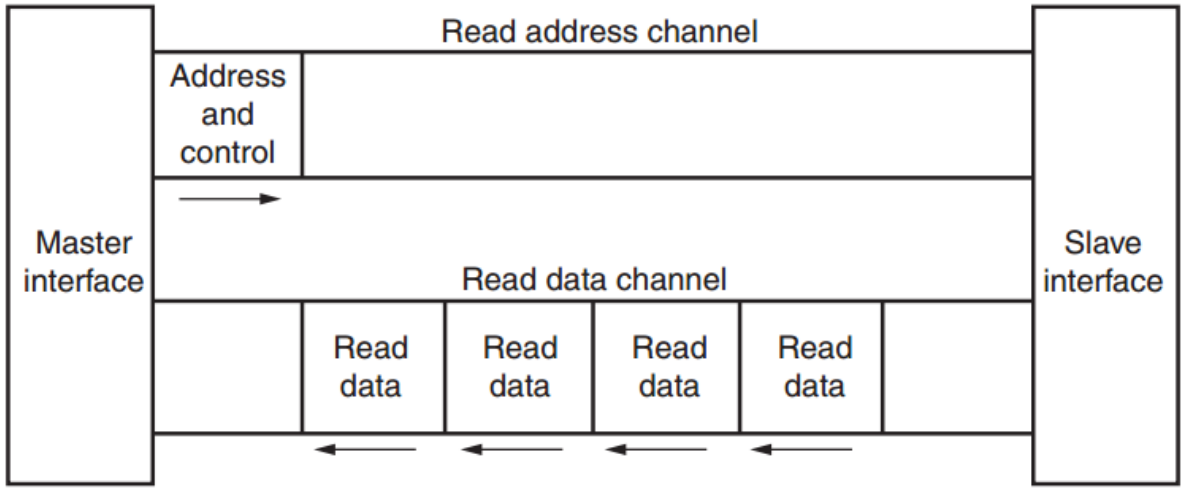
AXI Memory Mapped: Channels

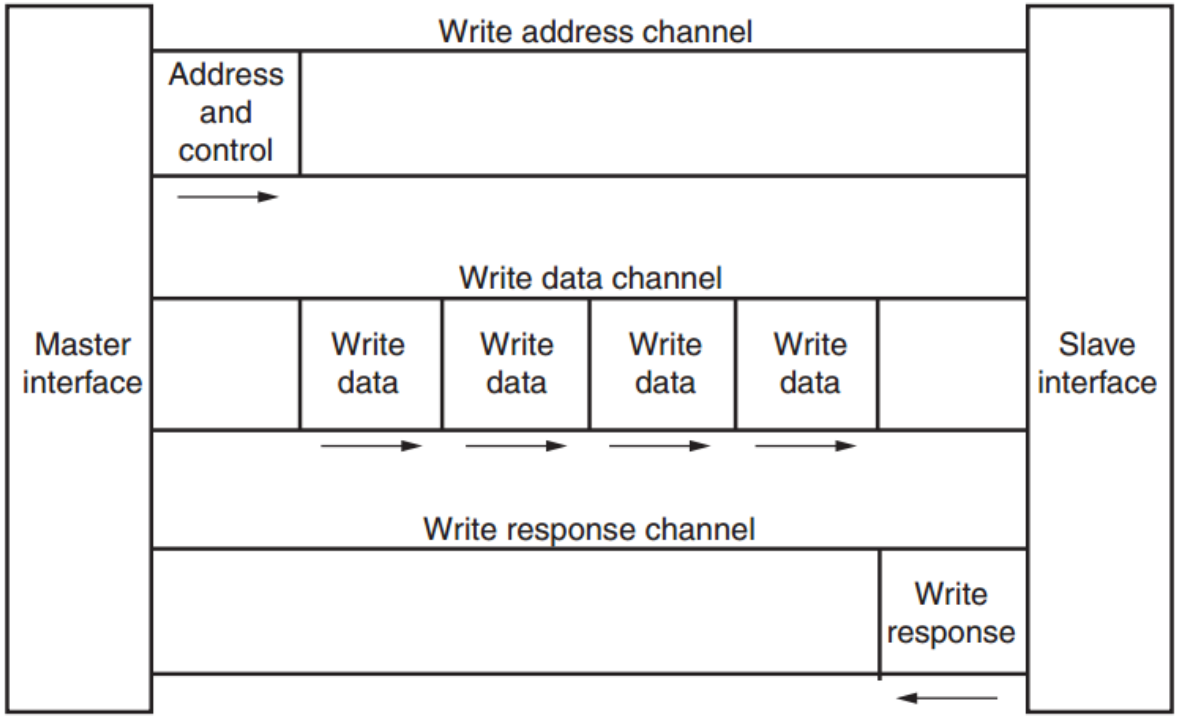


Each channel may carry additional control information that describes the parameters of the transactions

AXI Memory Mapped

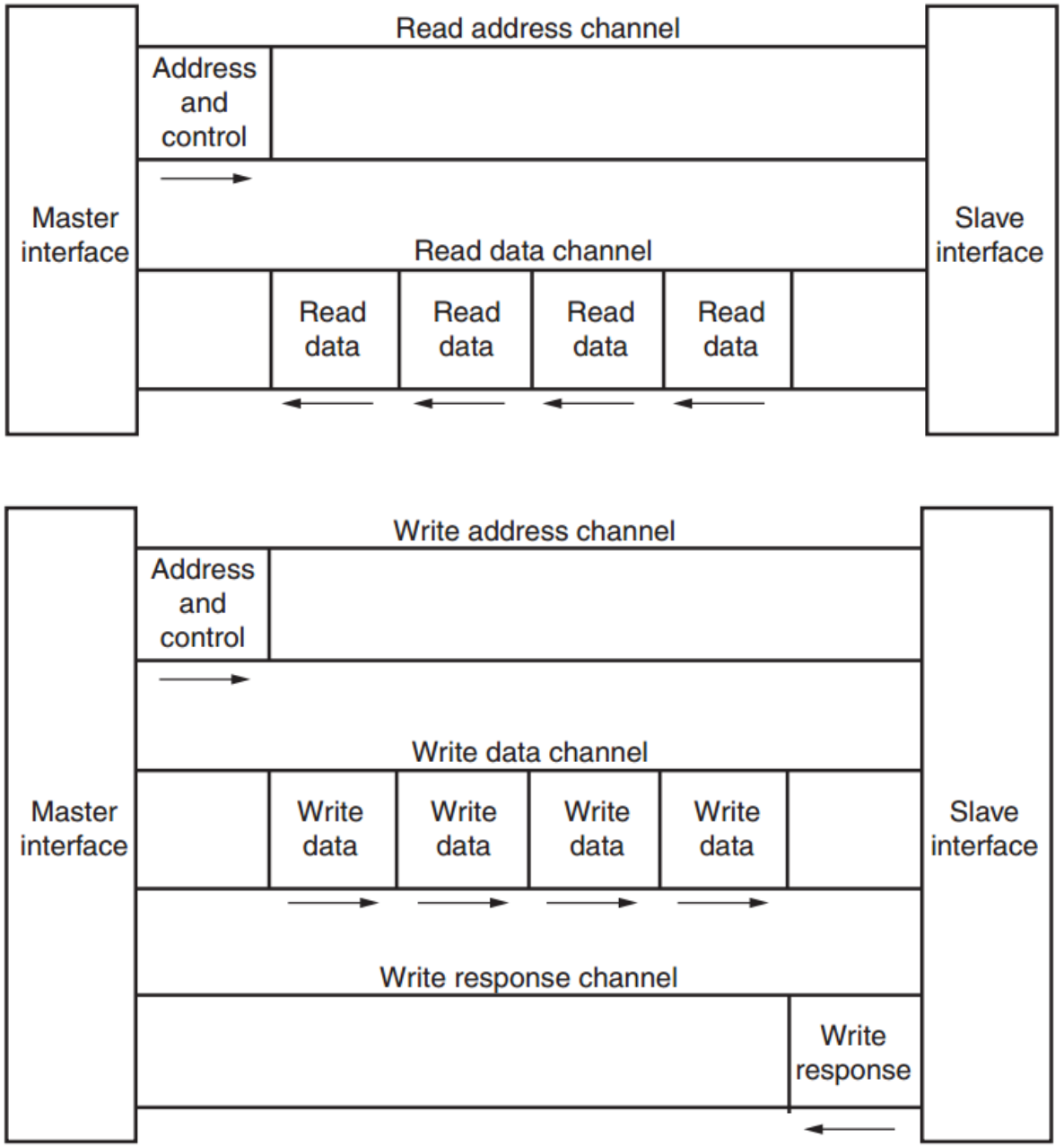






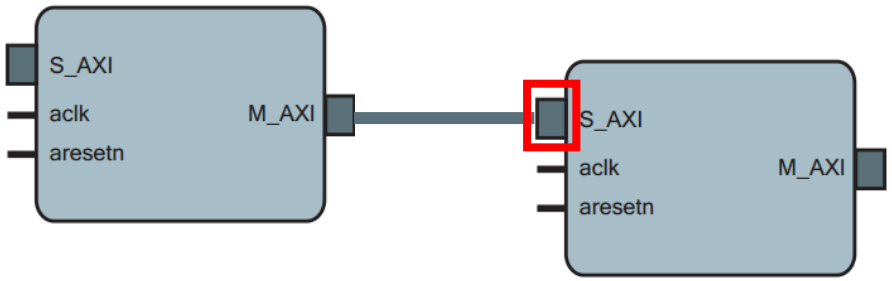
AXI Memory Mapped

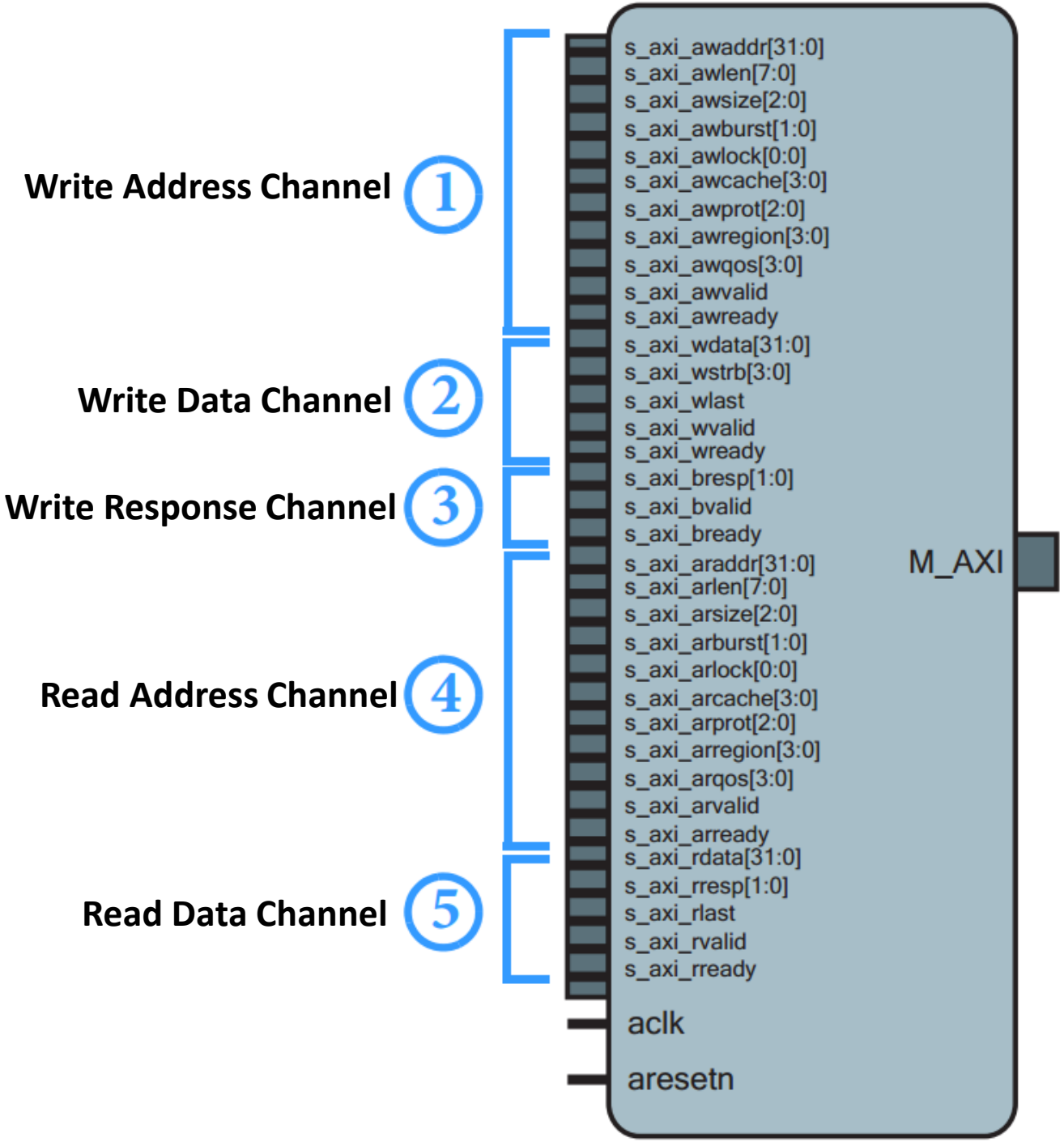
- Bursts varying from 1 to 256 data transfers per burst
- Each burst transfer size can be 1,2,4,8..128 bytes
- **Byte lane strobe signal** for every eight data bits, indicating which bytes of the data are valid



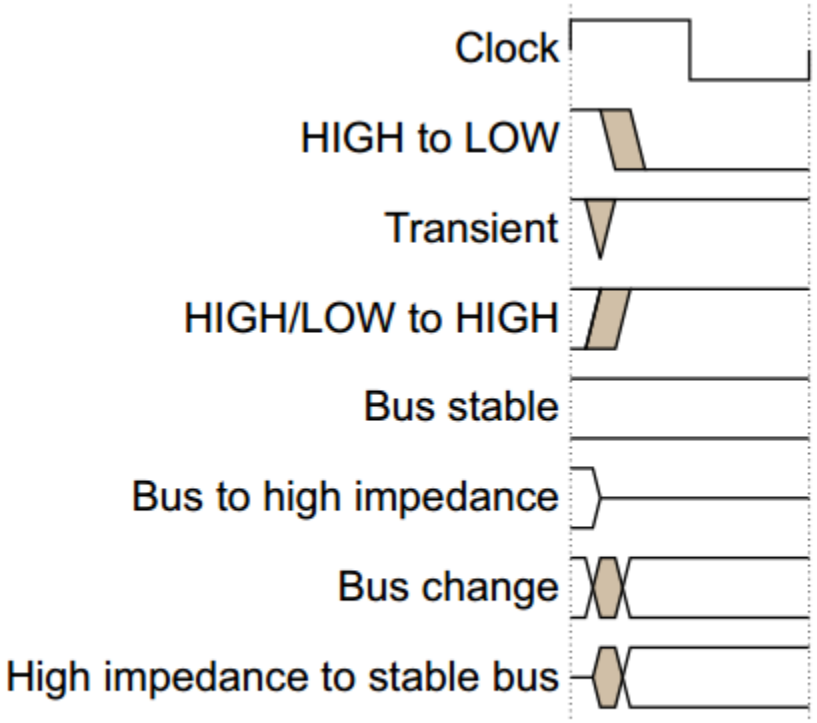
AXI Memory Mapped: Channels

- AXI4 provides separate data and address/control connections for Reads and Writes, which allows simultaneous, bidirectional data transfer.
- Whether you're doing a read or write, an AXI4 transaction is the same. An address comes out, then the data is either written by the AXI master or provided by the AXI slave.
- The extra channel, Write Response, allows the AXI slave to tell the AXI master, in a write transaction, status on the transaction.

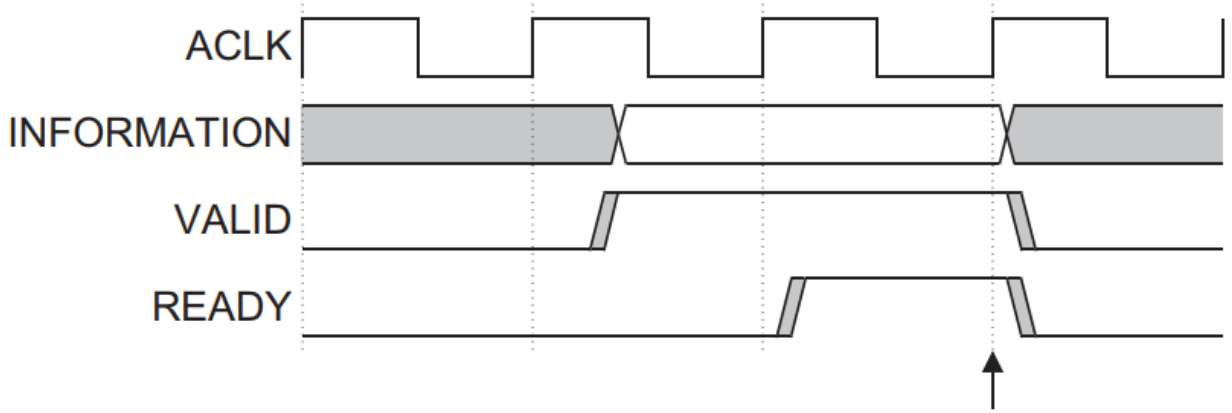

S_AXI
aclk
aresetn
M_AXI
S_AXI
aclk
aresetn
M_AXI


Write Address Channel
1
Write Data Channel
2
Write Response Channel
3
Read Address Channel
4
Read Data Channel
5
s_axi_awaddr[31:0]
s_axi_awlen[7:0]
s_axi_awsize[2:0]
s_axi_awburst[1:0]
s_axi_awlock[0:0]
s_axi_awcache[3:0]
s_axi_awprot[2:0]
s_axi_awregion[3:0]
s_axi_awqos[3:0]
s_axi_awvalid
s_axi_awready
s_axi_wdata[31:0]
s_axi_wstrb[3:0]
s_axi_wlast
s_axi_wvalid
s_axi_wready
s_axi_bresp[1:0]
s_axi_bvalid
s_axi_bready
s_axi_araddr[31:0]
s_axi_arlen[7:0]
s_axi_arsize[2:0]
s_axi_arburst[1:0]
s_axi_arlock[0:0]
s_axi_arcache[3:0]
s_axi_arprot[2:0]
s_axi_arregion[3:0]
s_axi_arqos[3:0]
s_axi_arvalid
s_axi_arready
s_axi_rdata[31:0]
s_axi_rresp[1:0]
s_axi_rlast
s_axi_rvalid
s_axi_rready
aclk
aresetn
M_AXI

AXI Memory Mapped

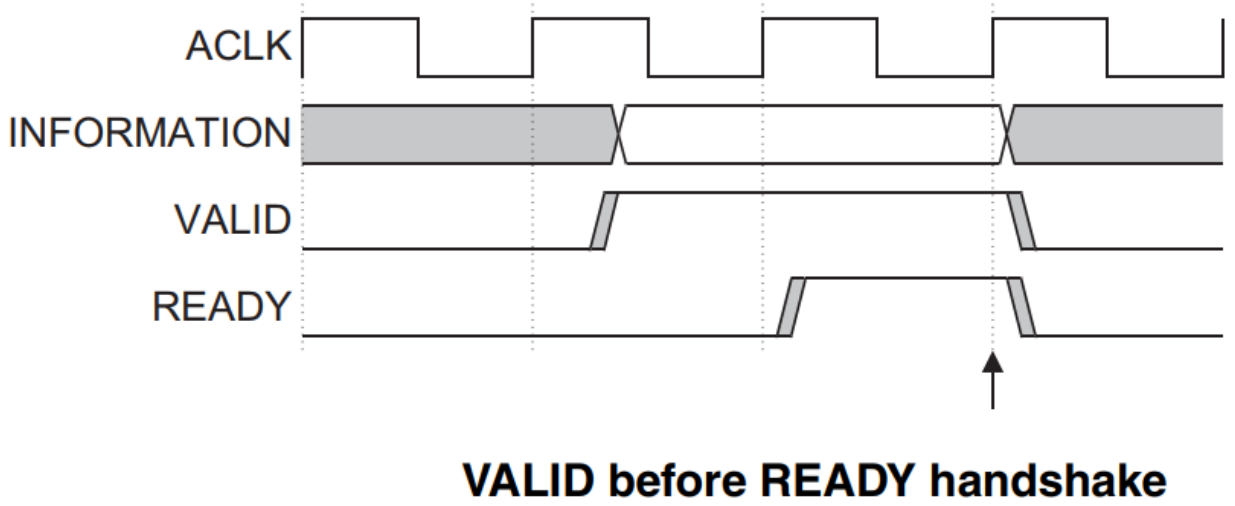


Key to timing diagram conventions



VALID before READY handshake

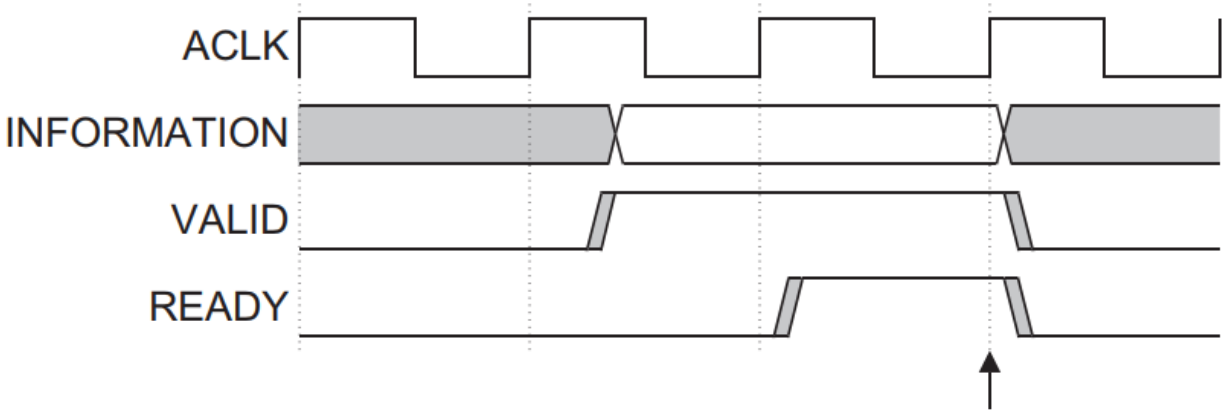
AXI Memory Mapped



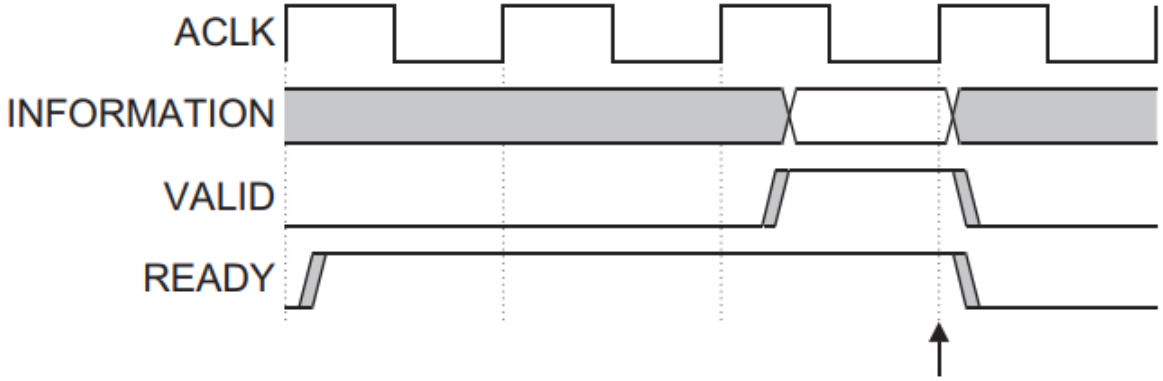
VALID before READY handshake

- The *source* generates the **VALID** signal to indicate when the address, data or control information is available.
- The *destination* generates the **READY** signal to indicate that it can accept the information.
- Transfer occurs only when *both* the **VALID** and **READY** signals are HIGH

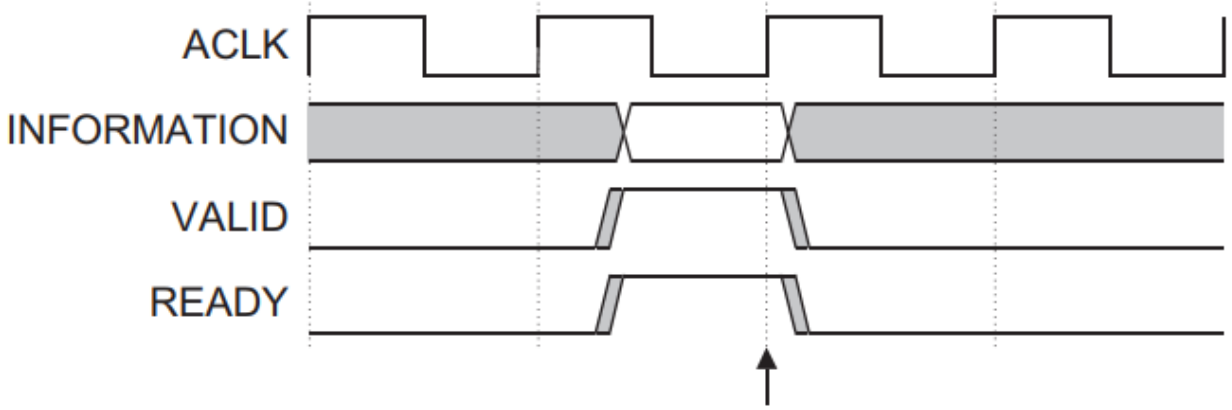
AXI Memory Mapped



VALID before READY handshake

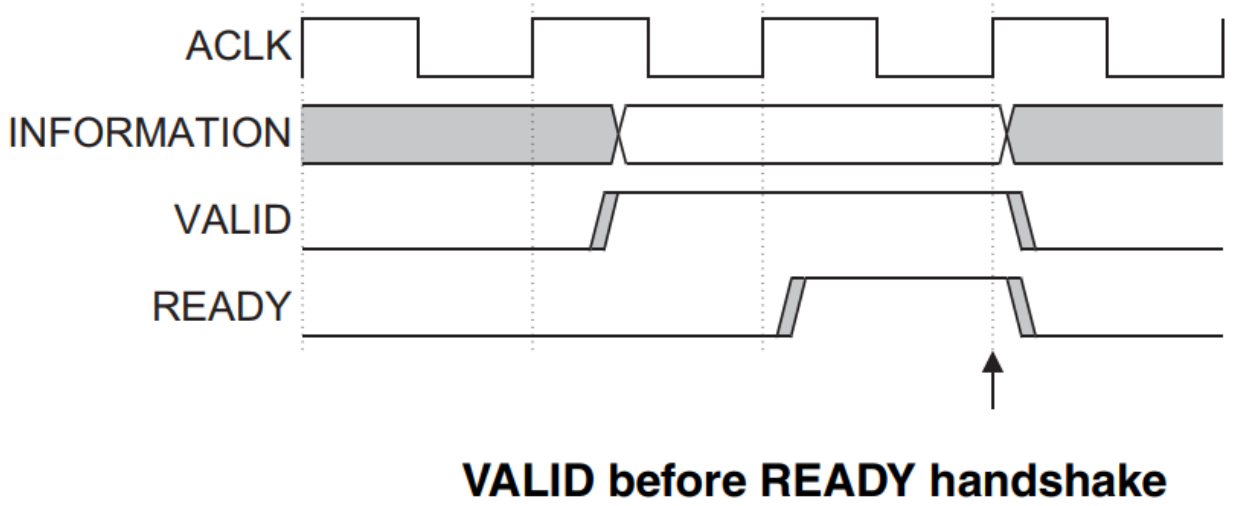


READY before VALID handshake



VALID with READY handshake

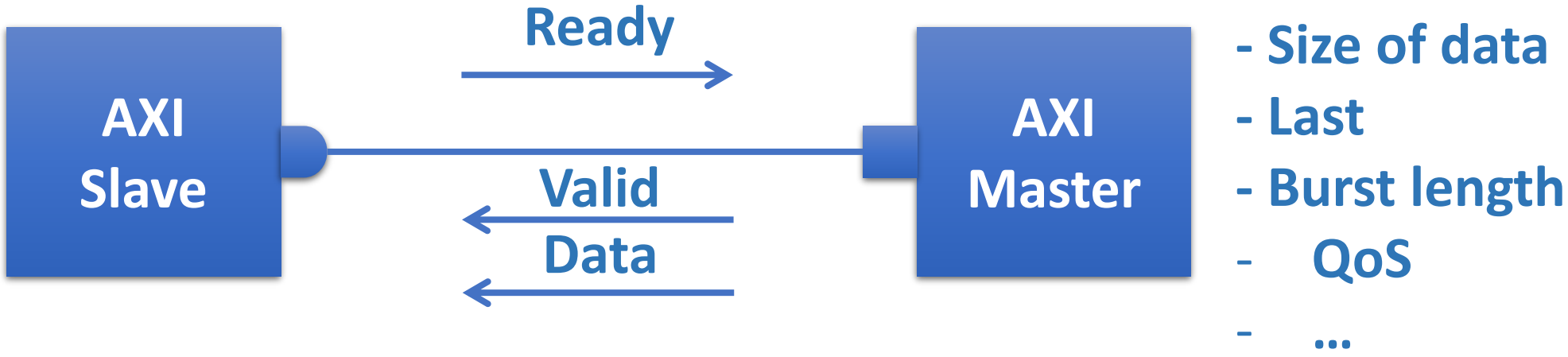
AXI Memory Mapped



VALID before READY handshake

- A source is NOT permitted to wait until READY is asserted before asserting VALID
- Once VALID is asserted it must remain asserted until the handshake occurs, at a rising clock edge at which VALID and READY are both asserted.
- A destination is permitted to wait for **VALID** to be asserted before asserting the corresponding **READY**.
- If **READY** is asserted, it is permitted to deassert **READY** before **VALID** is asserted.

AXI Memory Mapped



- Each of the independent channels consists of a set of information signals along with **VALID** and **READY** signals that provide a two-way handshake mechanism.
- The information source uses the **VALID** signal to show when valid address, data or control information is available on the channel.
- The destination uses the **READY** signal to show when it can accept the information.
- Both the read data channel and the write data channel also include a **LAST** signal to indicate the transfer of the final data item in a transaction.

AXI Memory Mapped

Transaction channel	Handshake pair
Write address channel	**AWVALID, AWREADY**

Write address channel: The master can assert the AWVALID signal only when it drives valid address and control information. When asserted, AWVALID must remain asserted until the rising clock edge after the slave asserts AWREADY.

AXI Memory Mapped

Transaction channel	Handshake pair
Write address channel	**AWVALID, AWREADY**
Write data channel	**WVALID, WREADY**

Write data channel: During a write burst, the master can assert the WVALID signal only when it drives valid write data. When asserted, WVALID must remain asserted until the rising clock edge after the slave asserts WREADY.

AXI Memory Mapped

Transaction channel	Handshake pair
Write address channel	**AWVALID, AWREADY**
Write data channel	**WVALID, WREADY**
Write response channel	**BVALID, BREADY**

Write response channel: The slave can assert the BVALID signal only when it drives a valid write response. When asserted, BVALID must remain asserted until the rising clock edge after the master asserts BREADY.

AXI Memory Mapped

Transaction channel	Handshake pair
Write address channel	**AWVALID, AWREADY**
Write data channel	**WVALID, WREADY**
Write response channel	**BVALID, BREADY**
Read address channel	**ARVALID, ARREADY**
Read data channel	**RVALID, RREADY**

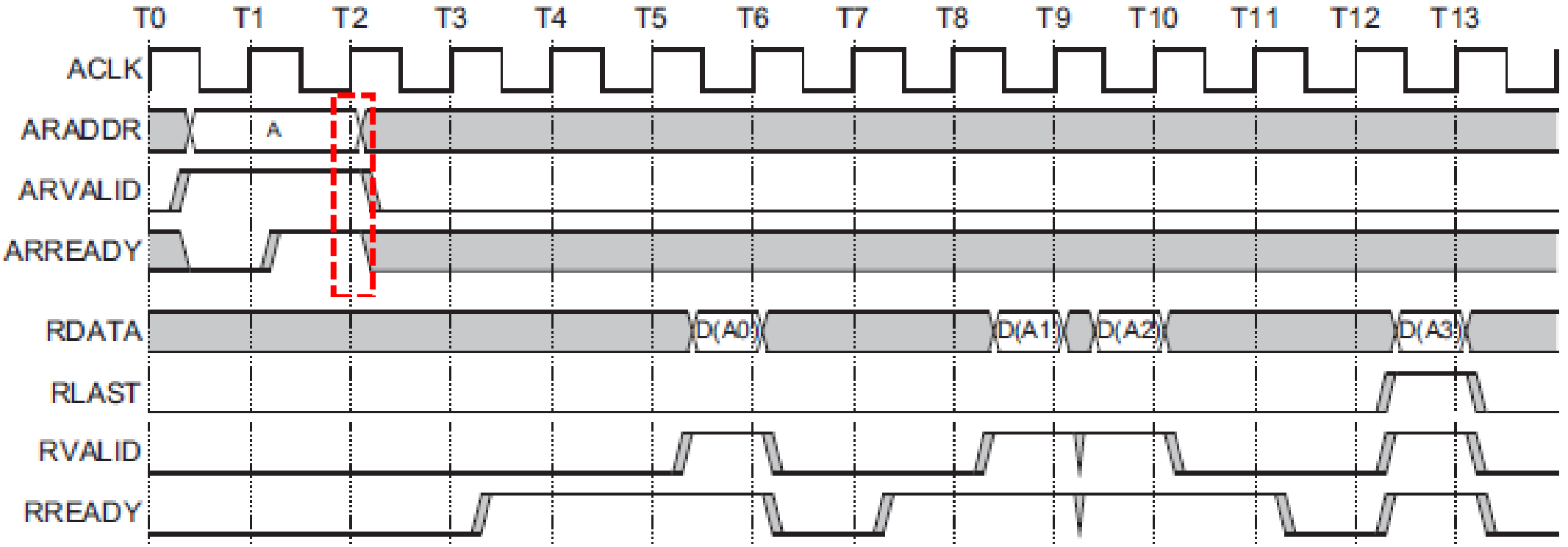
Read address channel: The master can assert the ARVALID signal only when it drives valid address and control information. When asserted, ARVALID must remain asserted until the rising clock edge after the slave asserts the ARREADY signal.

AXI Memory Mapped

Transaction channel	Handshake pair
Write address channel	**AWVALID, AWREADY**
Write data channel	**WVALID, WREADY**
Write response channel	**BVALID, BREADY**
Read address channel	**ARVALID, ARREADY**
Read data channel	**RVALID, RREADY**

Read data channel: The slave can assert the RVALID signal only when it drives valid read data. When asserted, RVALID must remain asserted until the rising clock edge after the master asserts RREADY. The master interface uses the RREADY signal to indicate that it accepts the data.

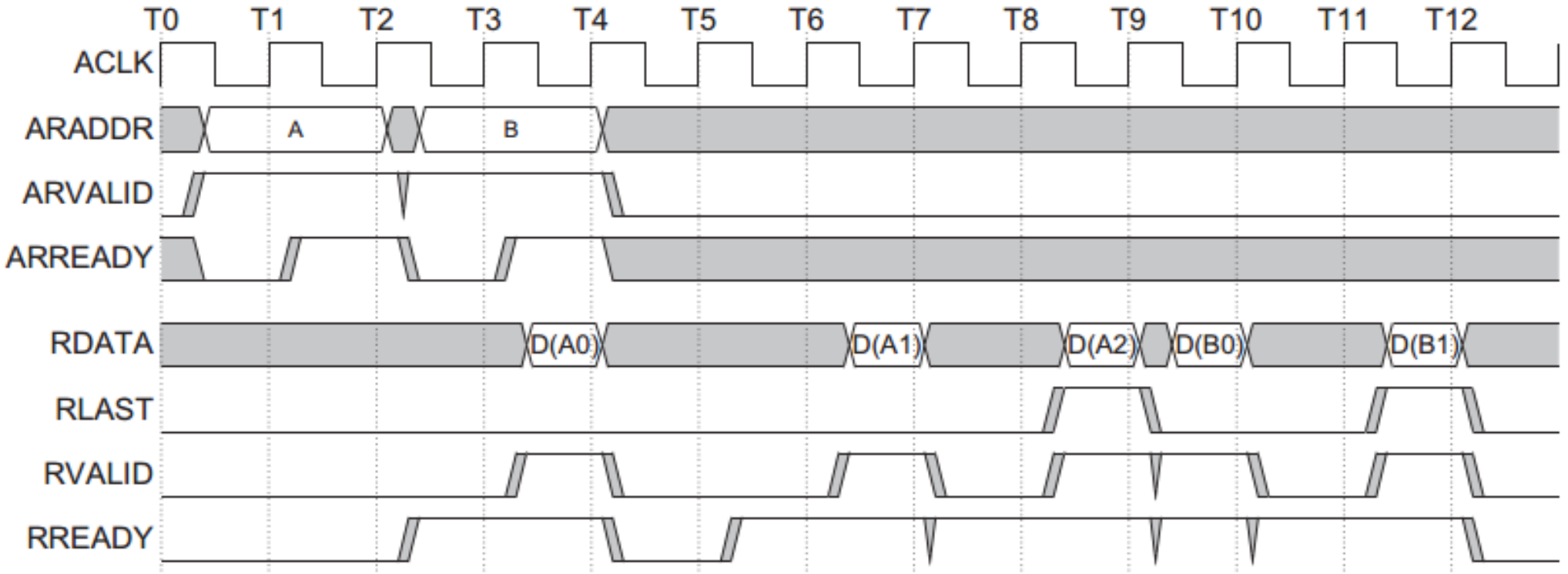
AXI Interface: Read



AXI – Burst Read

*Other control signals are not discussed here: ARID, ARLEN, ARSIZE, RID, RRESP,.....

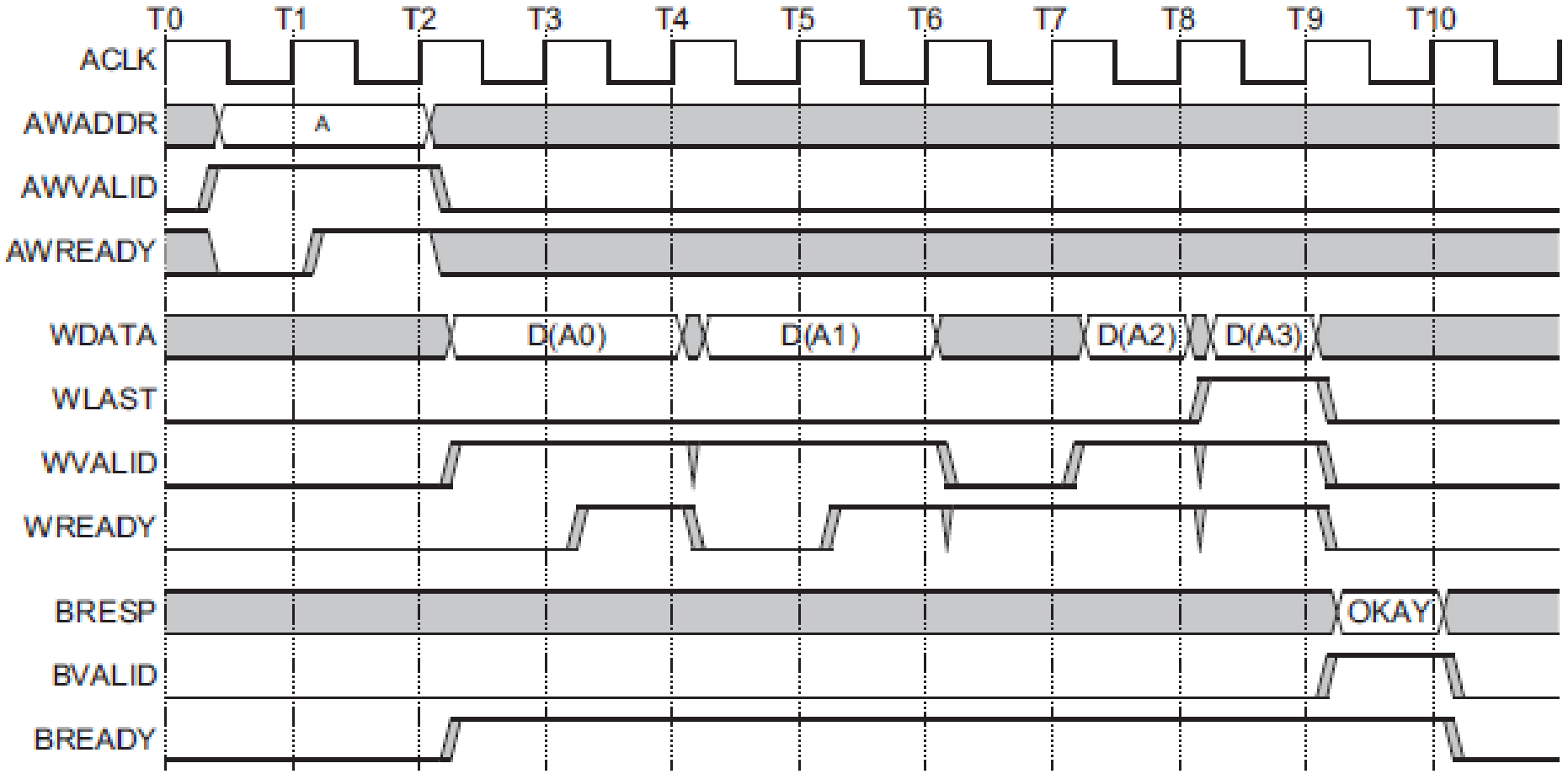
AXI Interface: Read



AXI – Burst Read

*Other control signals are not discussed here: ARID, ARLEN, ARSIZE, RID, RRESP,…..

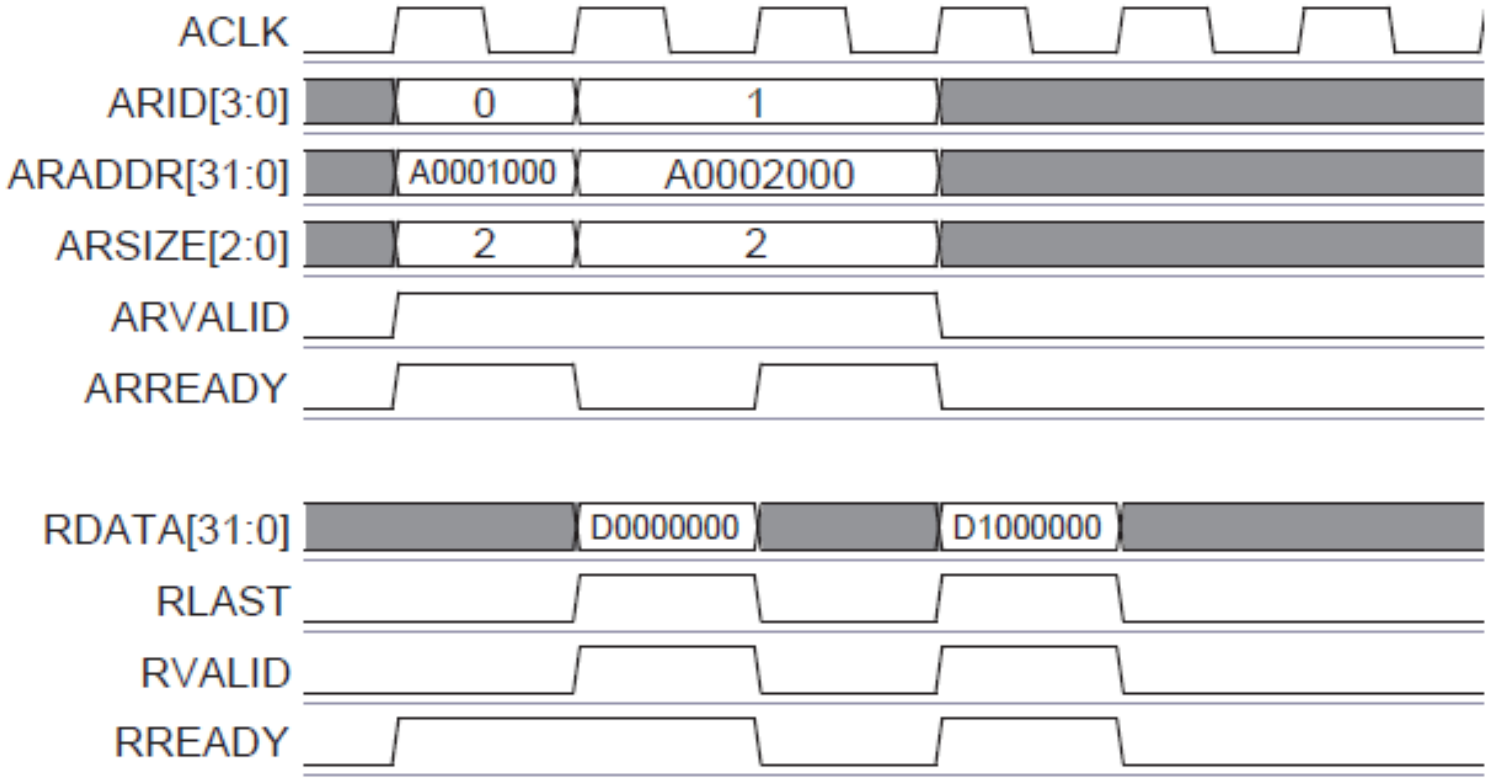
AXI Interface: Write



AXI Burst Write

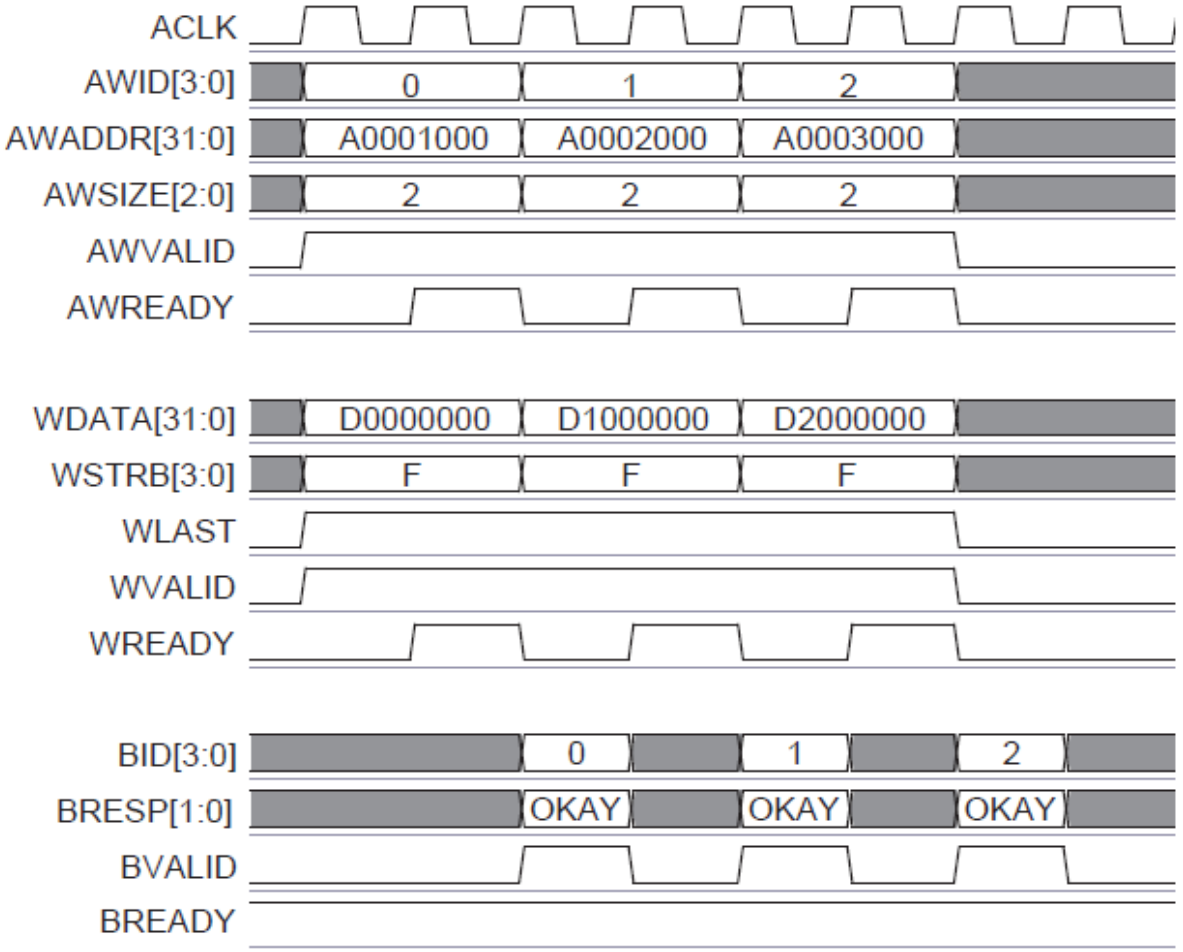
AXI Interface: Read

AxSIZE[2:0]	Bytes in transfer
0b000	1
0b001	2
0b010	4
0b011	8
0b100	16
0b101	32
0b110	64
0b111	128



AXI Interface: Write

AxSIZE[2:0]	Bytes in transfer
0b000	1
0b001	2
0b010	4
0b011	8
0b100	16
0b101	32
0b110	64
0b111	128



Example of AXI4 multiple write transactions

AXI Interface: Burst

AXI Interface: Burst

AXI Interface: Burst

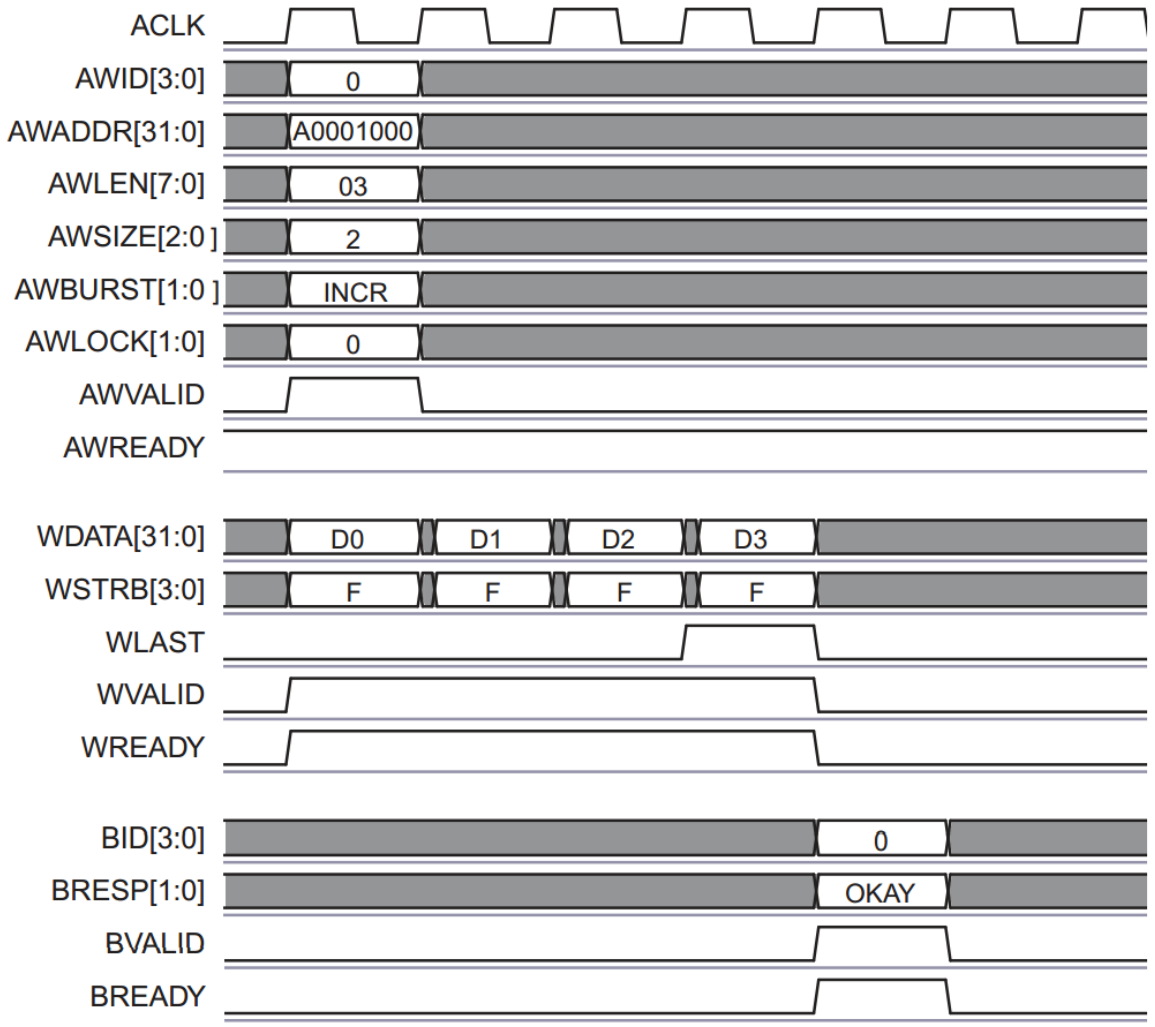
ARBURST[1:0] AWBURST[1:0]	Burst type	Description	Access
b00	FIXED	Fixed-address burst	FIFO-type
b01	INCR	Incrementing-address burst	Normal sequential memory
b10	WRAP	Incrementing-address burst that wraps to a lower address at the wrap boundary	Cache line
b11	Reserved	-	-

AXI Interface: Write

AxSIZE[2:0]	Bytes in transfer
0b000	1
0b001	2
0b010	4
0b011	8
0b100	16
0b101	32
0b110	64
0b111	128

The burst length for AXI4 is defined as,

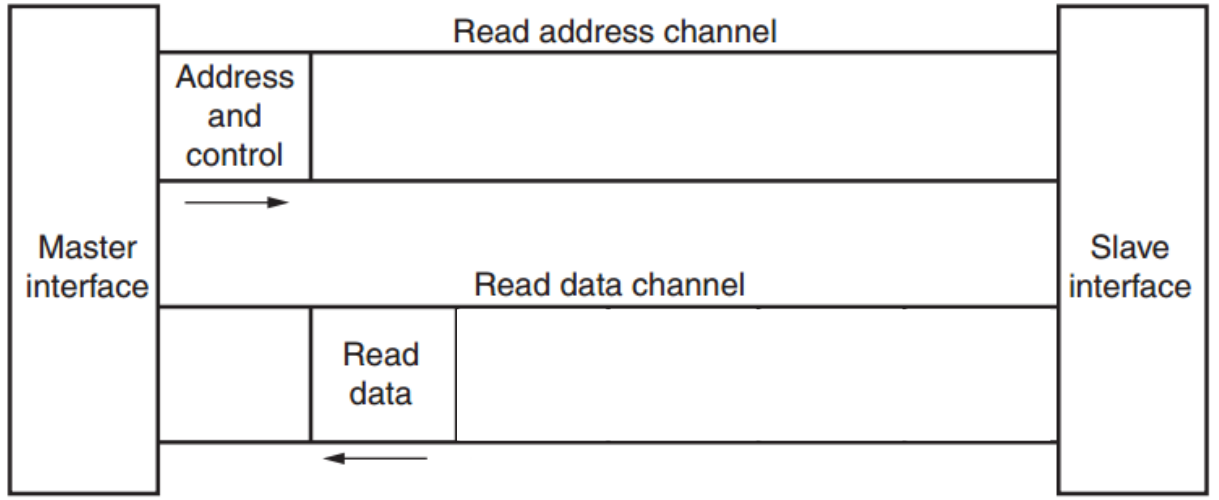
Burst_Length = **AxLEN[7:0]** + 1

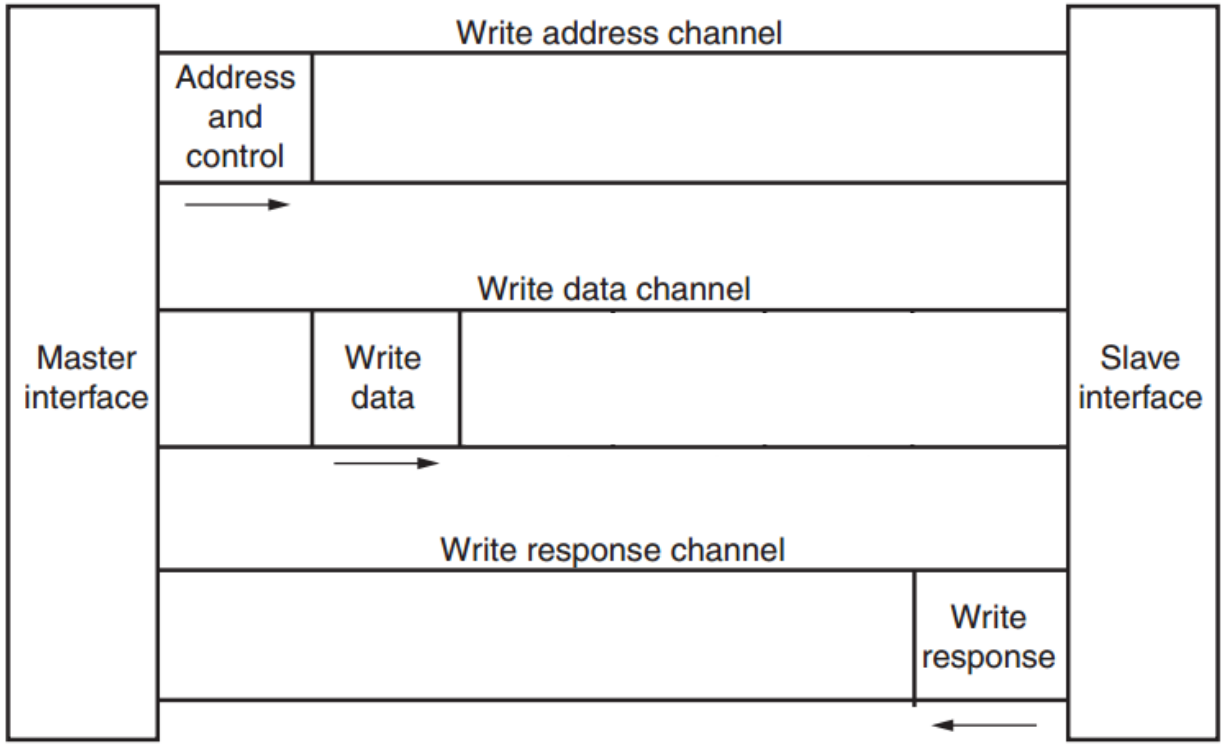


Example of AXI4 write burst transaction

AXI Lite

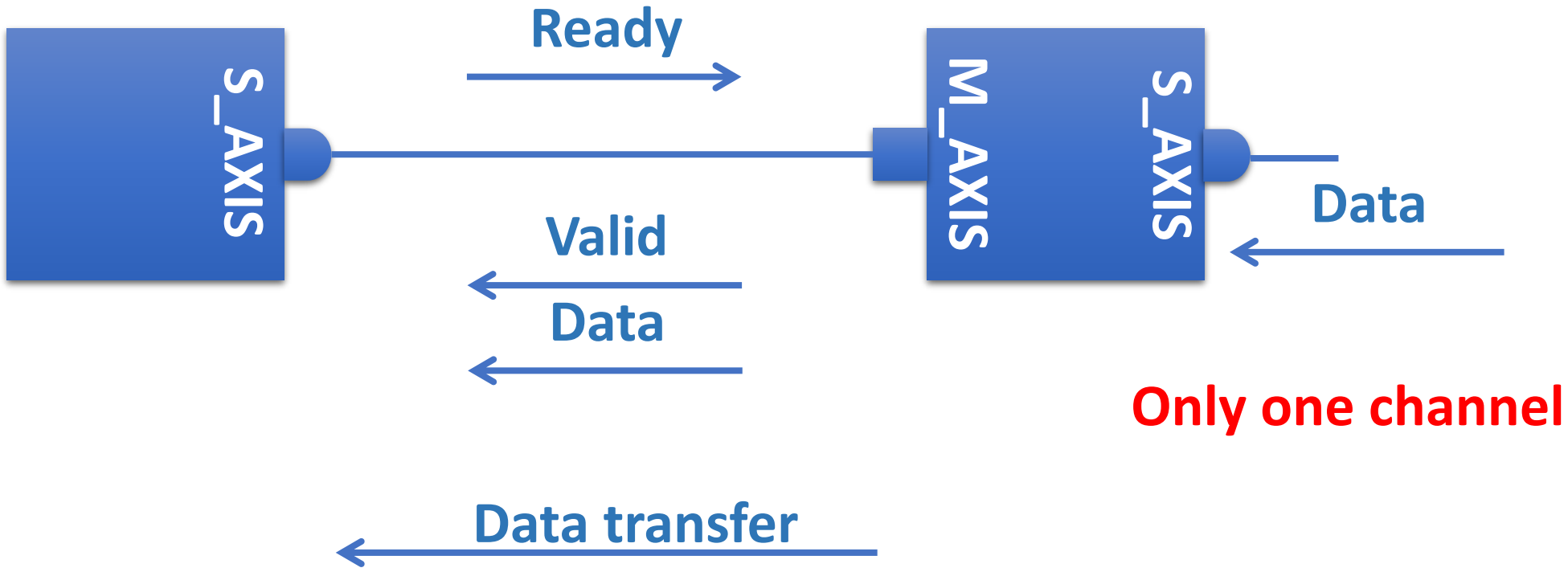
- Bursting is not supported
- Subset of the AXI4 interface intended for communication with control registers and have small footprint



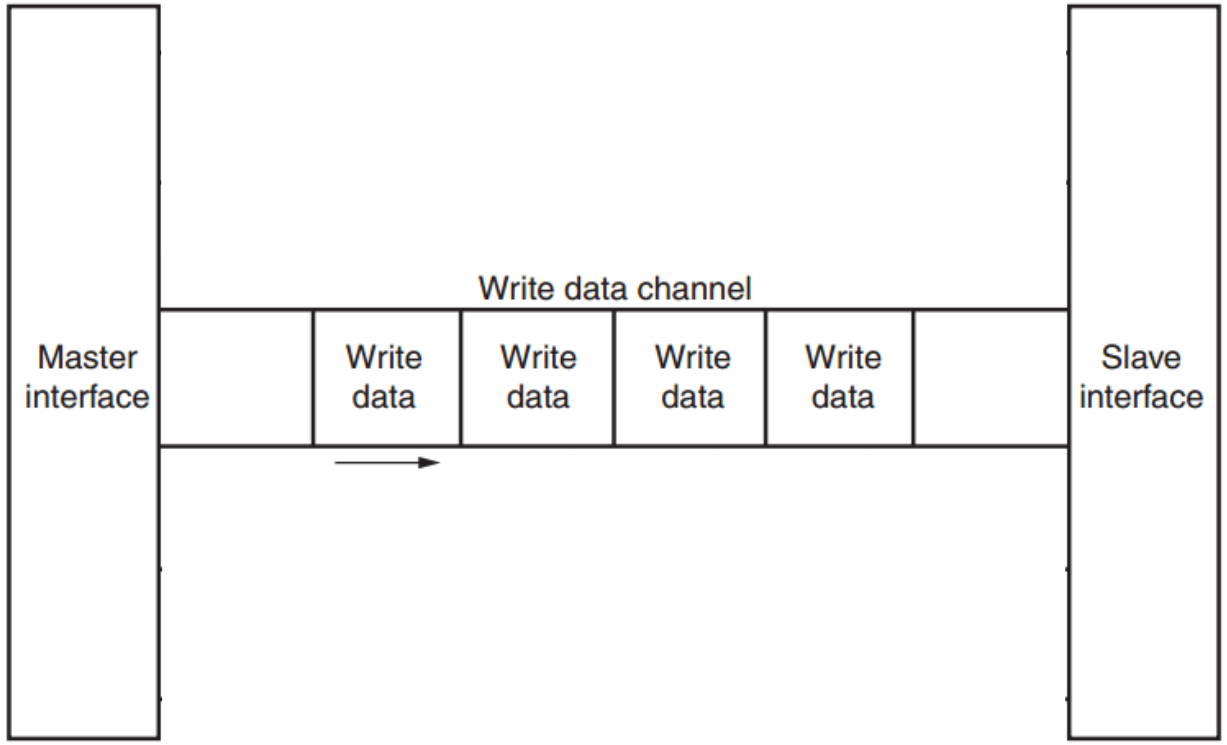


AXI Stream

- The AXI4-Stream protocol defines a single channel for transmission of streaming data (unlimited burst).

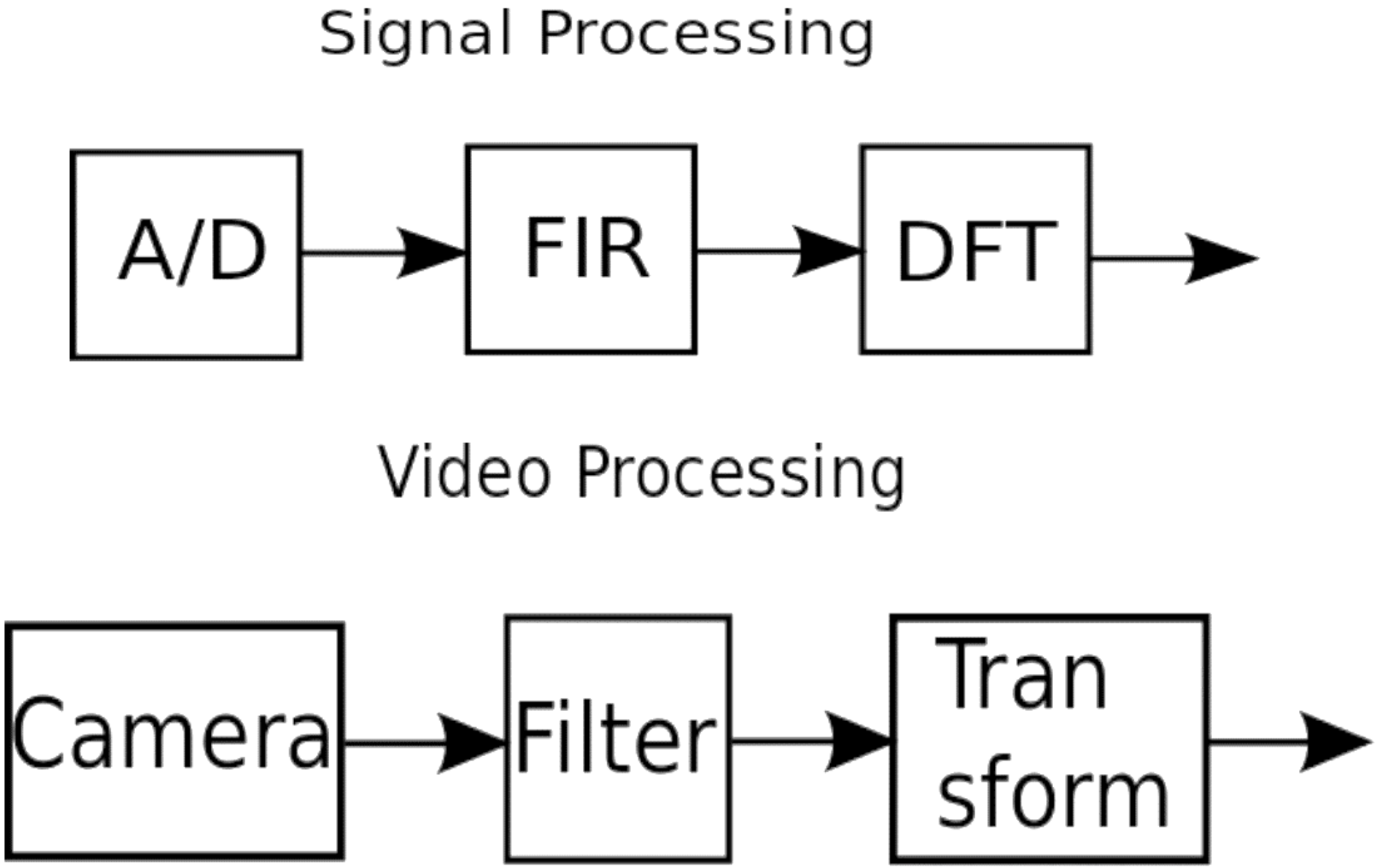


AXI Stream



- The AXI4-Stream channel is modeled after the Write Data channel of the AXI4.
- Unlike AXI4, AXI4-Stream interfaces can burst an **unlimited** amount of data.

AXI Stream


Signal Processing
A/D
FIR
DFT
Video Processing
Camera
Filter
Tran
sform