N26F300
VLSI SYSTEM DESIGN
(GRADUATE LEVEL)

Bus Interface

Outline

- Processor
- Custom processor – GCD example
- Peripherals
- **Interfacing via bus**

3 Interfacing

Bus Overview

AHB Bus

AXI Bus

[Material partly adapted from 1) MOE SLD Course materials by Prof. I.G. Huang@ NSYSU; 2) and ARMB AXI protocol specification.]

Outline

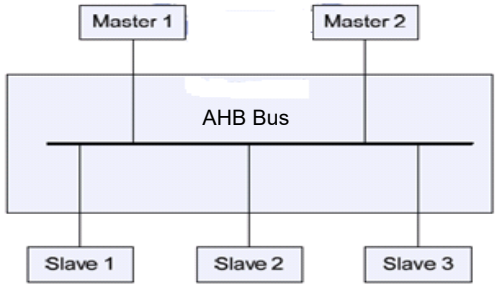
- Comparison between the AXI and the AHB
- AXI Introduction
- Basic transfer examples
- Channel handshake
- Additional features

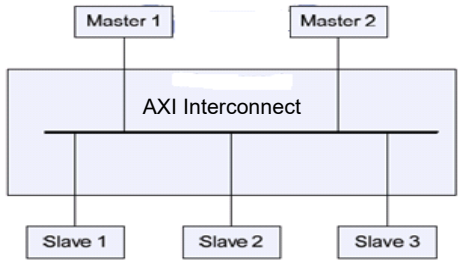
Comparison between the AXI and the AHB

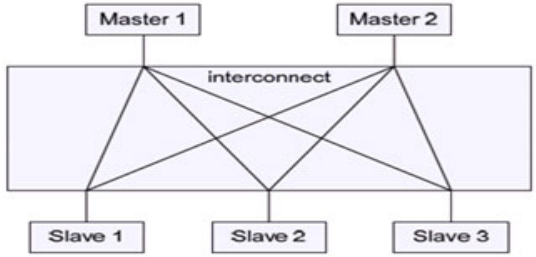
Name	AMBA 3 AXI	AMBA 2 AHB
Communication channel	5 difference channels for address, data and control signals transfer	One communication bus for address and data transfer
Transaction completion	Out-of-order transaction	In-order transaction
Transfer direction	Uni-direction	Bi-direction
Address issue	Multiple outstanding address	One time only can deal one transaction
Improve frequency	Easy addition of register stages to provide timing closure.	Hard to isolate timing
Atomic operations	Exclusive, Locked access	Locked access

Hardware cost

- Hardware architecture
  - AMBA 2.0v AHB
    - One communication bus for address control signals and data transfer
    - Bus bandwidth – 139 bits
  - AMBA 3.0v AXI(shared bus architecture)
    - 5 difference channels for address, data and control signals transfer
    - Bus bandwidth – 206 bits
  - AMBA 3.0v AXI(crossbar architecture)
    - 5 difference channels for address, data and control signals transfer
    - Bus bandwidth – 206 bits * 2 * 3







Outline

- Comparison between the AXI and the AHB
- AXI Introduction
- Basic transfer examples
- Channel handshake
- Additional features

AXI Introduction

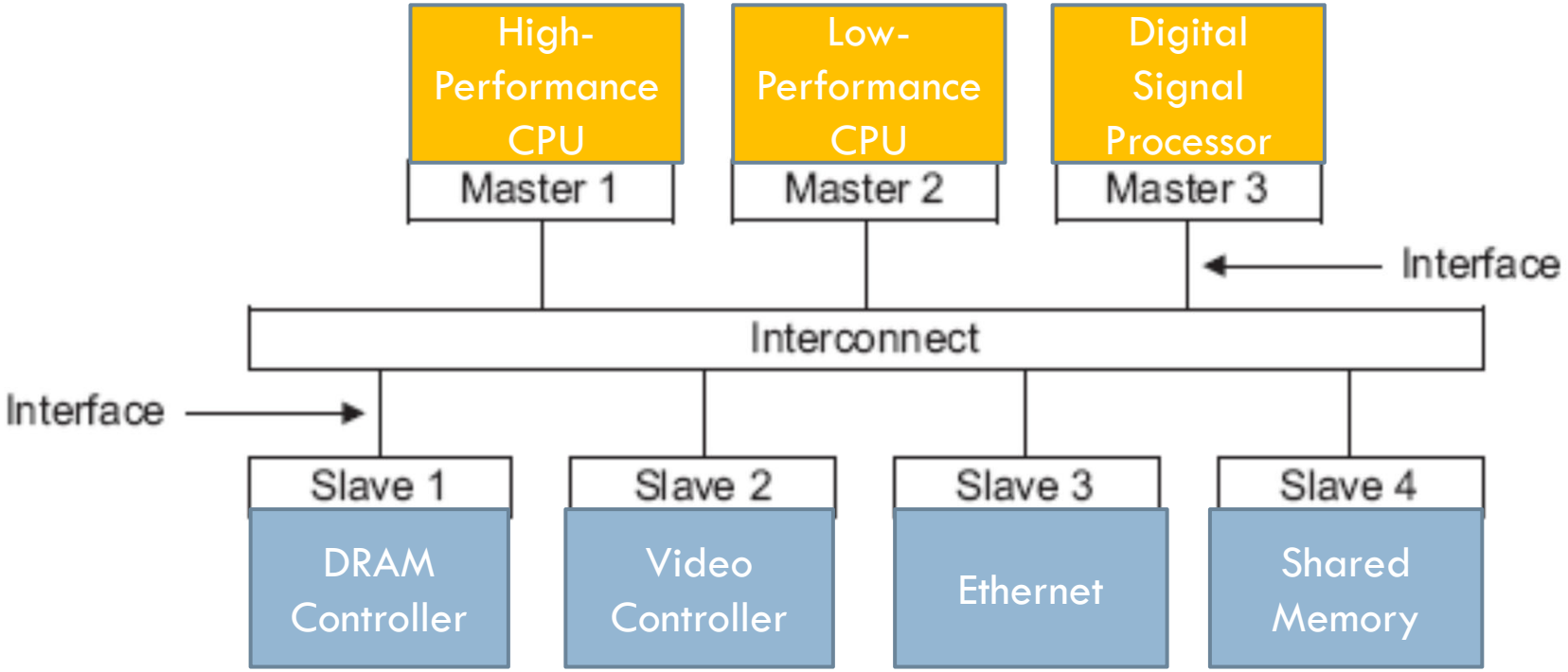
- AXI – Advanced eXtensible Interface
- Features
- architecture Introduction

Transaction: transfer of data from one point in the hardware to another point
Master: Initiates the transaction
Slave: Responds to the initiated transaction

http://www.gstitt.ece.ufl.edu/courses/fall15/eel4720_5721/labs/refs/AXI4_specification.pdf

AXI architecture overview

- AMBA AXI architecture overview

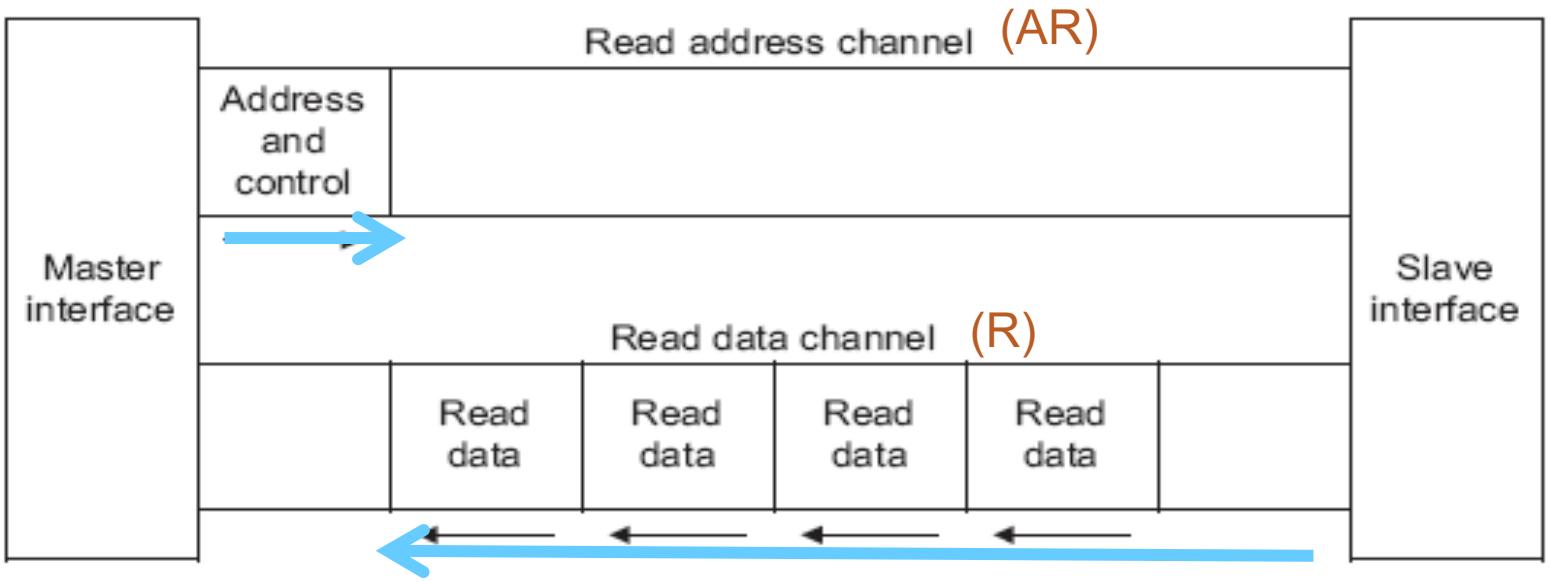


AXI Features

- Several revisions: AXI2, AXI3, AXI4
- Separate address/control and data phases
- Separate Read and Write data channels
- Support unaligned data transfers using byte strobes
  - Ex: access a 32-bit data that starts @ 0X80004002
- Ability to issue multiple outstanding addresses
- Out-of-order transaction completion
- Easy addition of register stages to provide timing closure
- Easy addition of register stages for timing closure
  - 5 independent channels, each in one direction

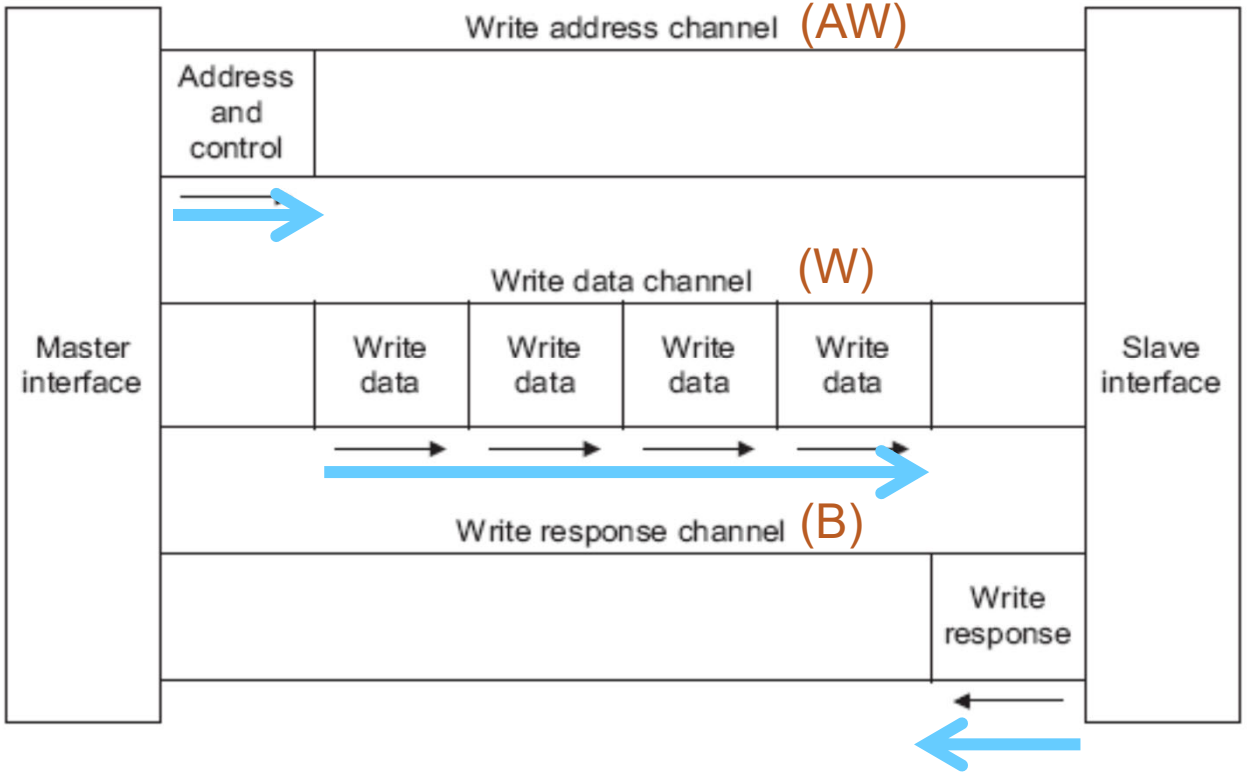
Channel architecture - reads

- Address/Control is issued before the actual data transfer
- 2 Channels for Read Transaction (out of 5 channels)
- Read address channel, Read data channel



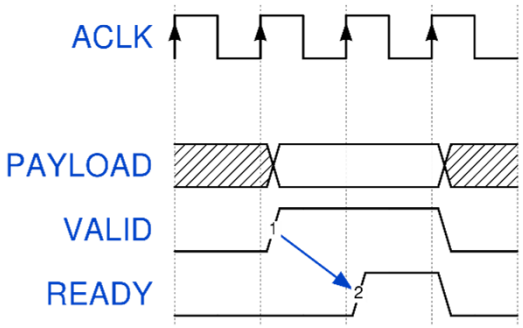
Channel architecture - write

- 3 Channels for Write Transaction (out of 5 channels)
- Write Address channel, Write data channel, Write response channel



Key Handshake Signals

- Two-way handshake mechanism composed by VALID and READY
  - VALID
    - Asserts when valid data or control information available on the channel
  - READY
    - Asserts when receive can accept the data
  - LAST
    - Asserts while the final data completes

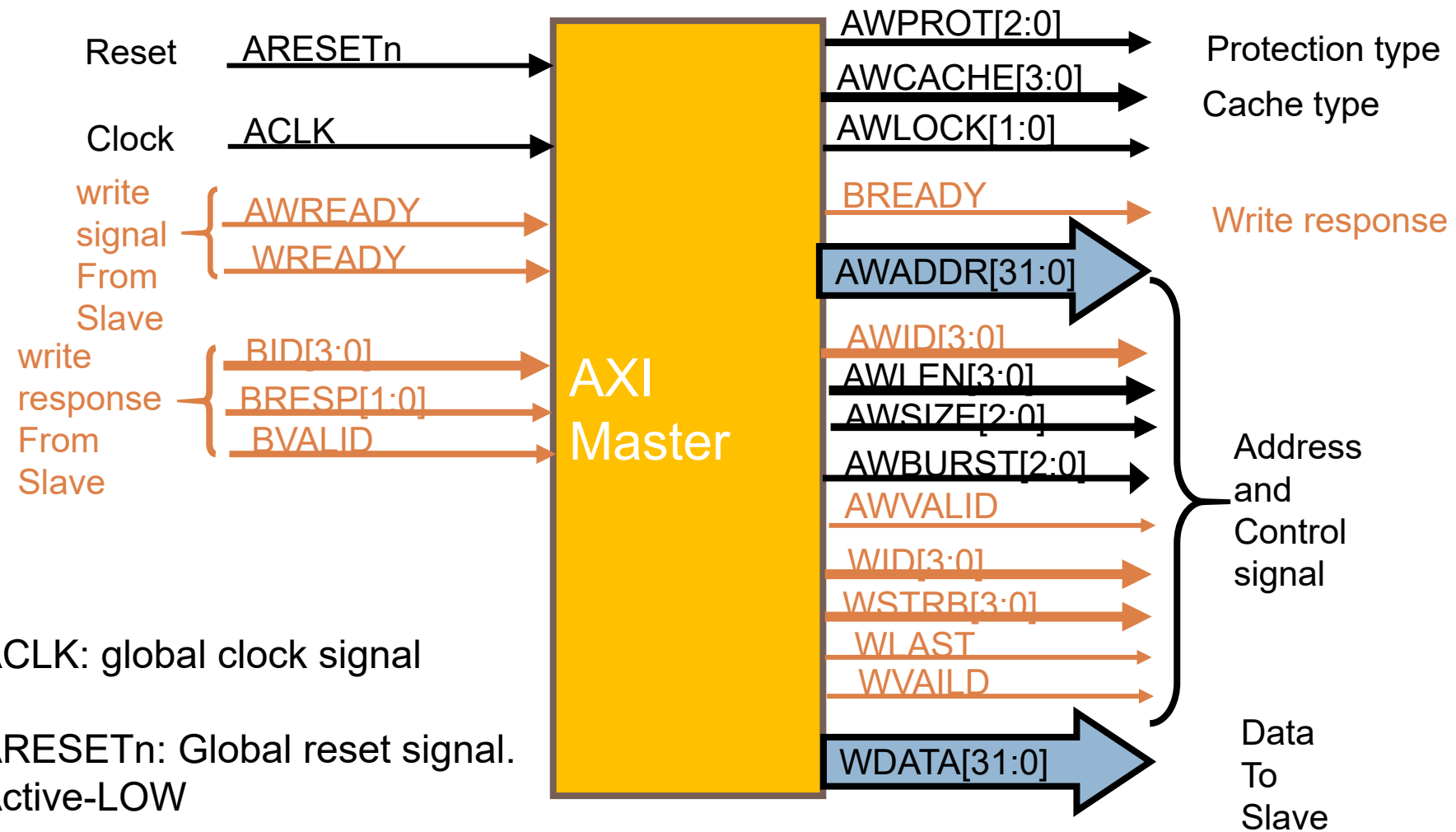


Source: wikipedia

payload: 承載量,在一堆資料中我們所關心的部分;

xVALID or xREADY, where x is one of {AR, R, AW, W, B}.

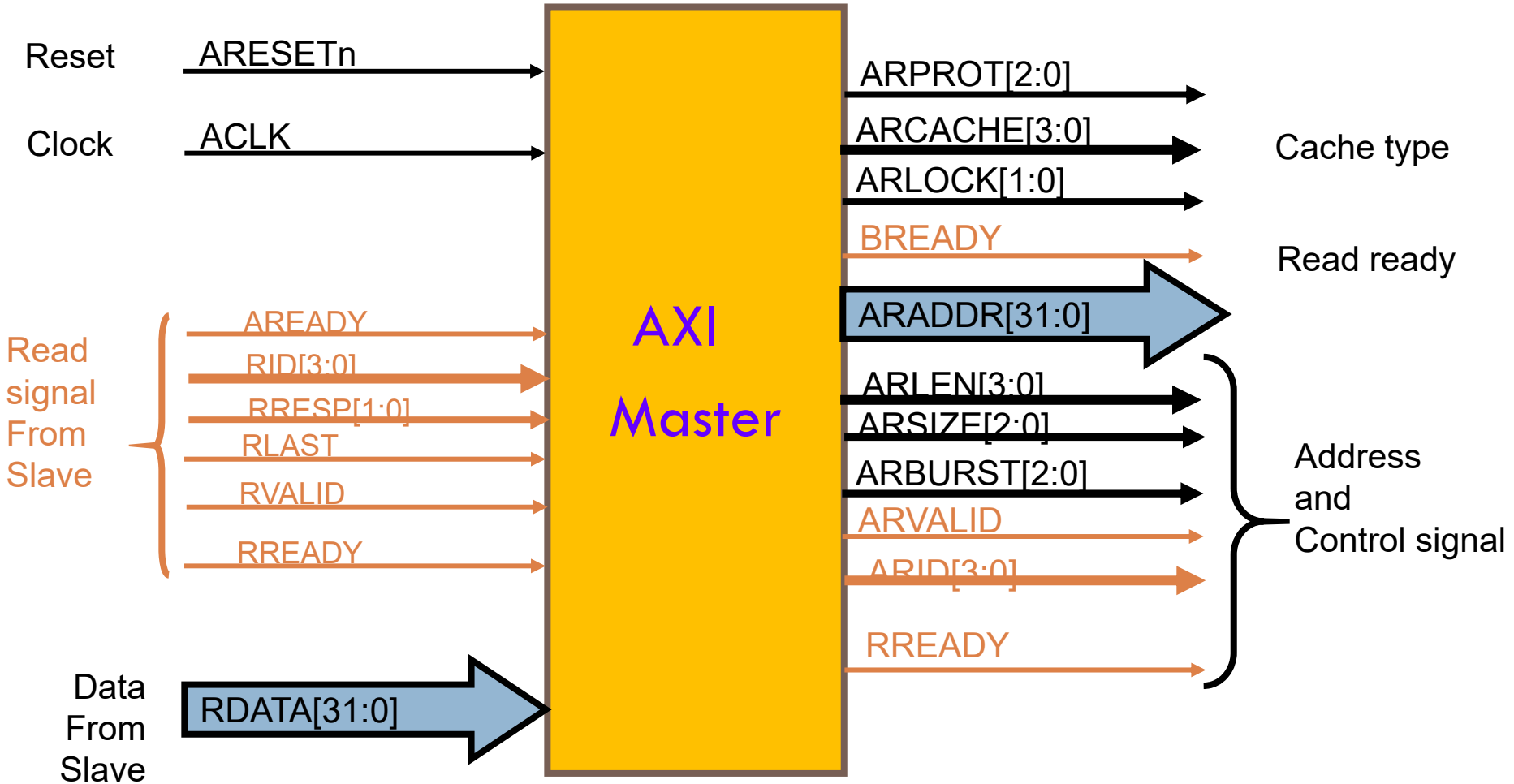
Master - WRITE



ACLK: global clock signal

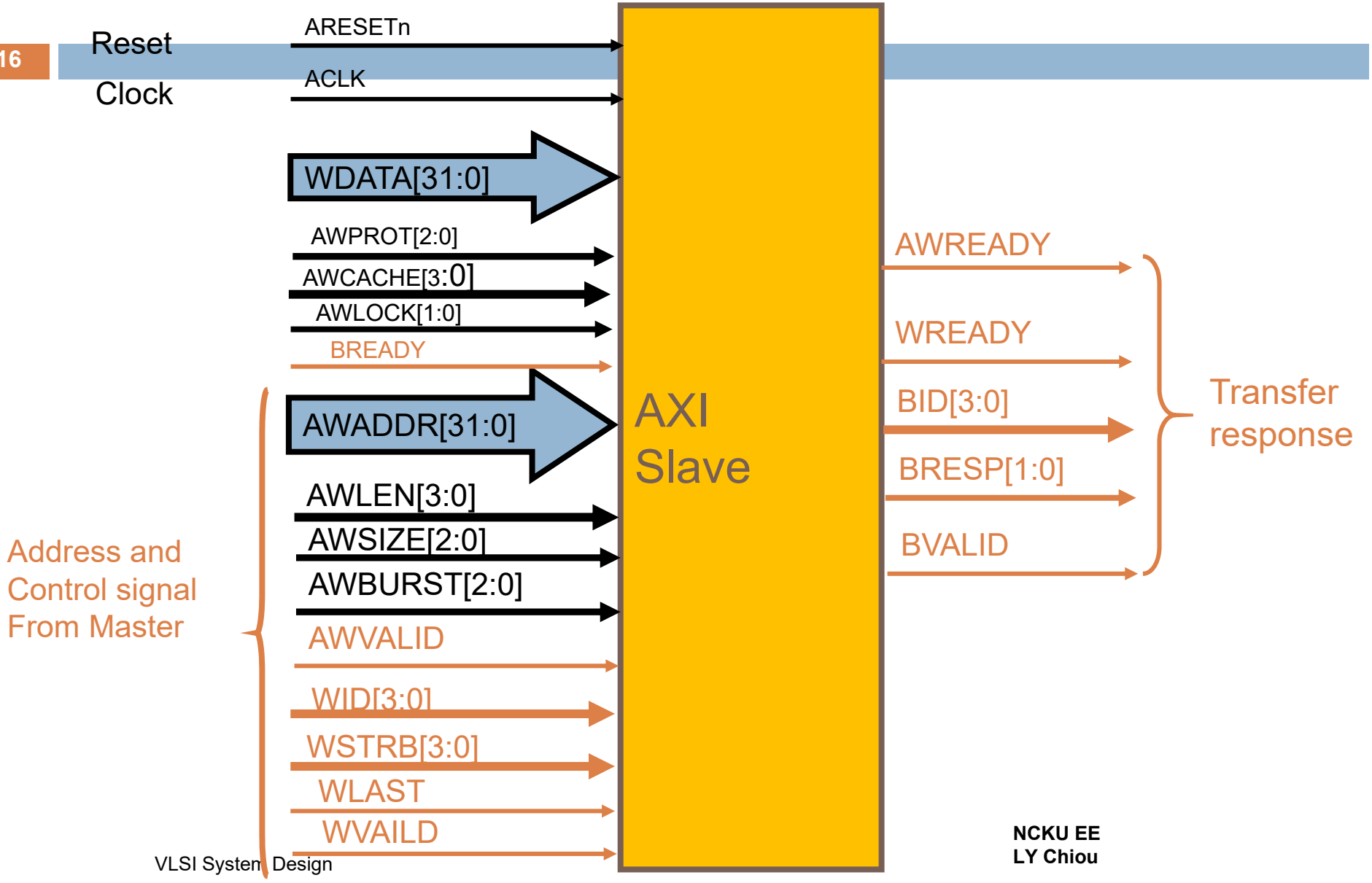
ARESETn: Global reset signal. Active-LOW

Master - READ



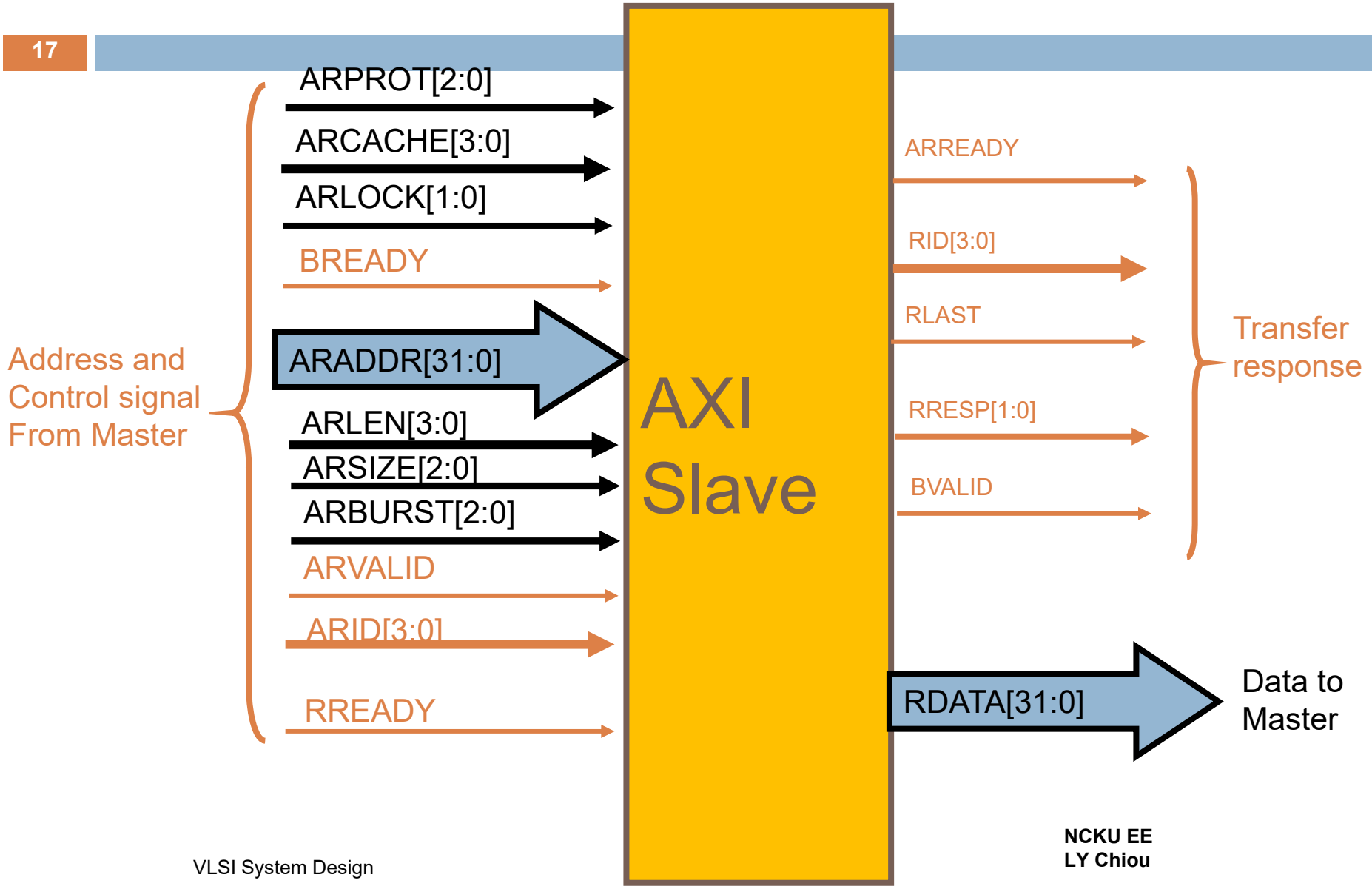
NCKU EE
LY Chiou

Slave - WRITE



NCKU EE
LY Chiou

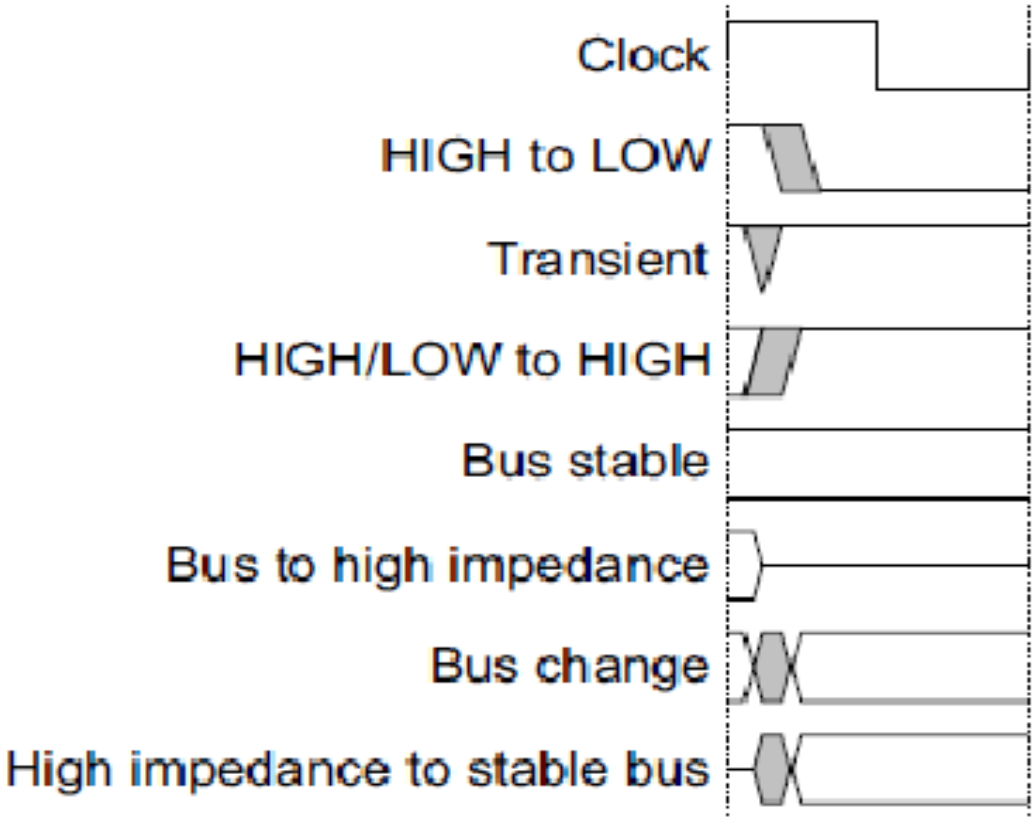
Slave - READ


ARPROT[2:0]
ARCACHE[3:0]
ARLOCK[1:0]
BREADY
ARADDR[31:0]
ARLEN[3:0]
ARSIZE[2:0]
ARBURST[2:0]
ARVALID
ARID[3:0]
RREADY
Address and
Control signal
From Master
AXI
Slave
ARREADY
RID[3:0]
RLAST
RRESP[1:0]
BVALID
Transfer
response
RDATA[31:0]
Data to
Master

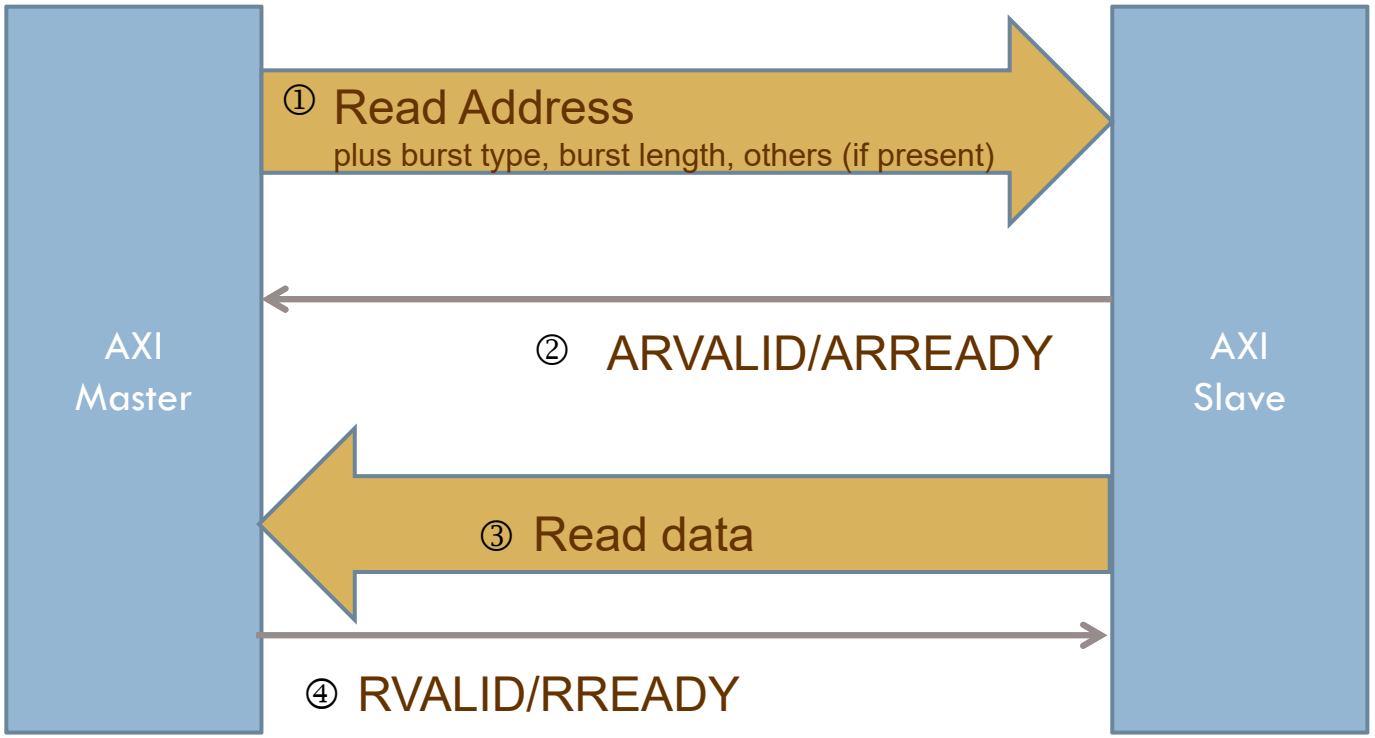
Outline

- Comparison between the AXI and the AHB
- AXI Introduction
- Basic transfer examples
- Channel handshake
- Additional features

Timing Diagram Conventions



Read Transaction

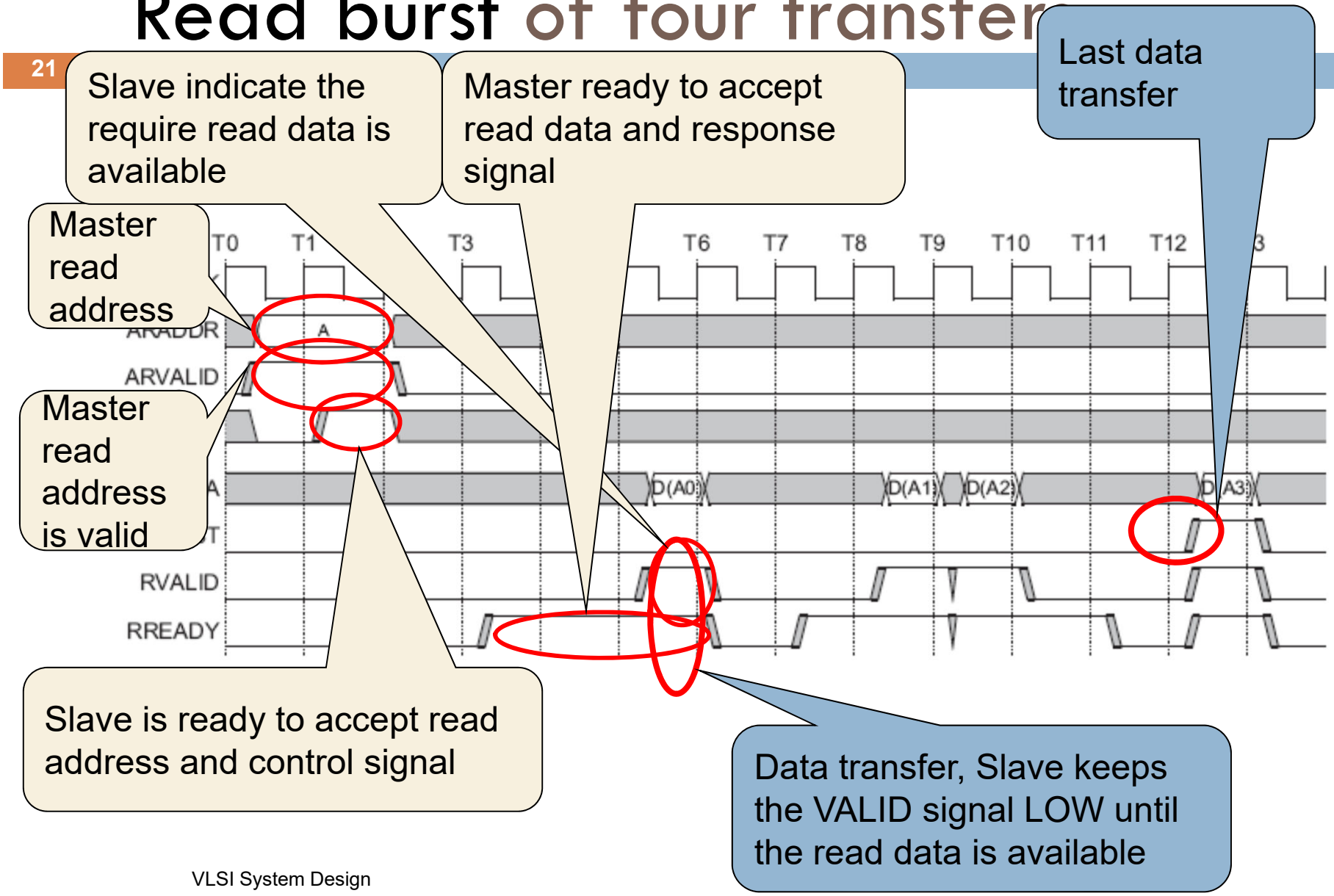


Two main rules:

- A source must not wait for a high xREAD to assert xVALID.
- Once asserted, a source must keep a high xVALID until a handshake occurs

A Simple Transfer Example –

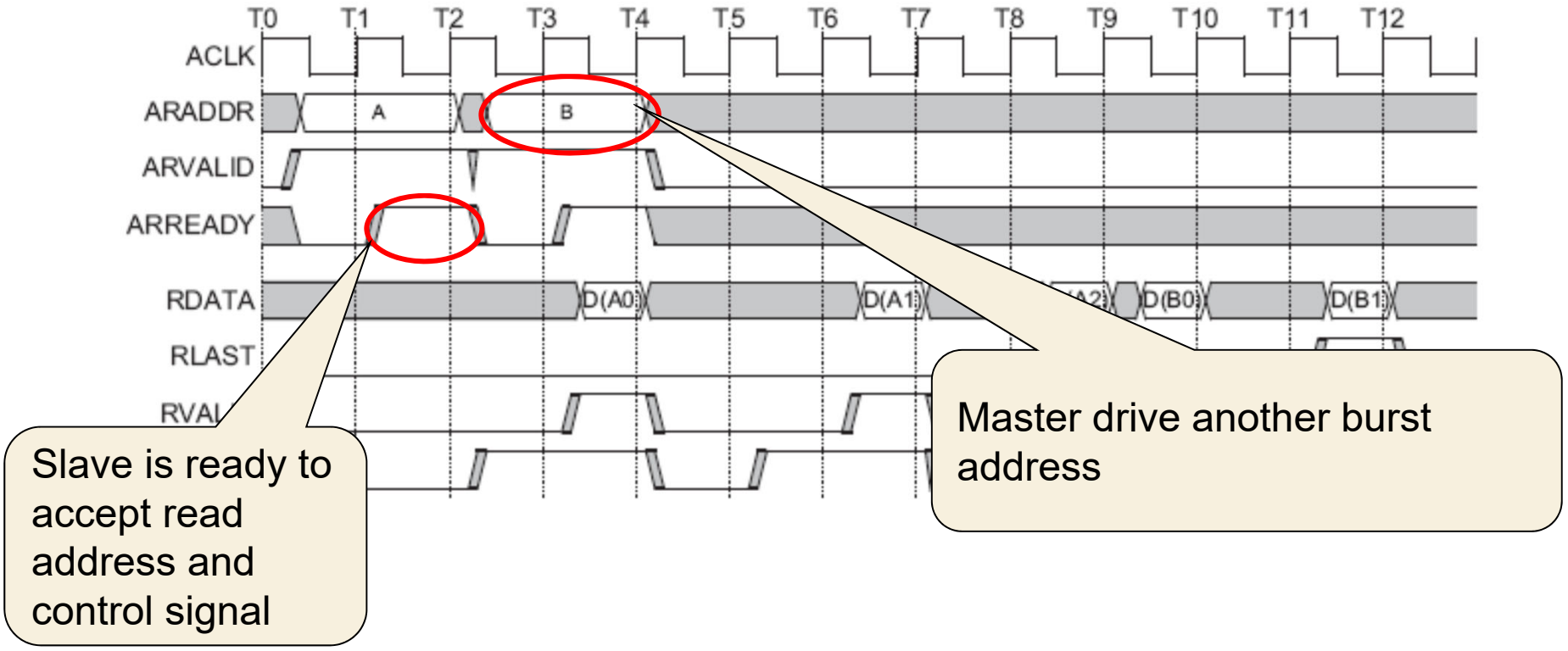
Read burst of four transfer



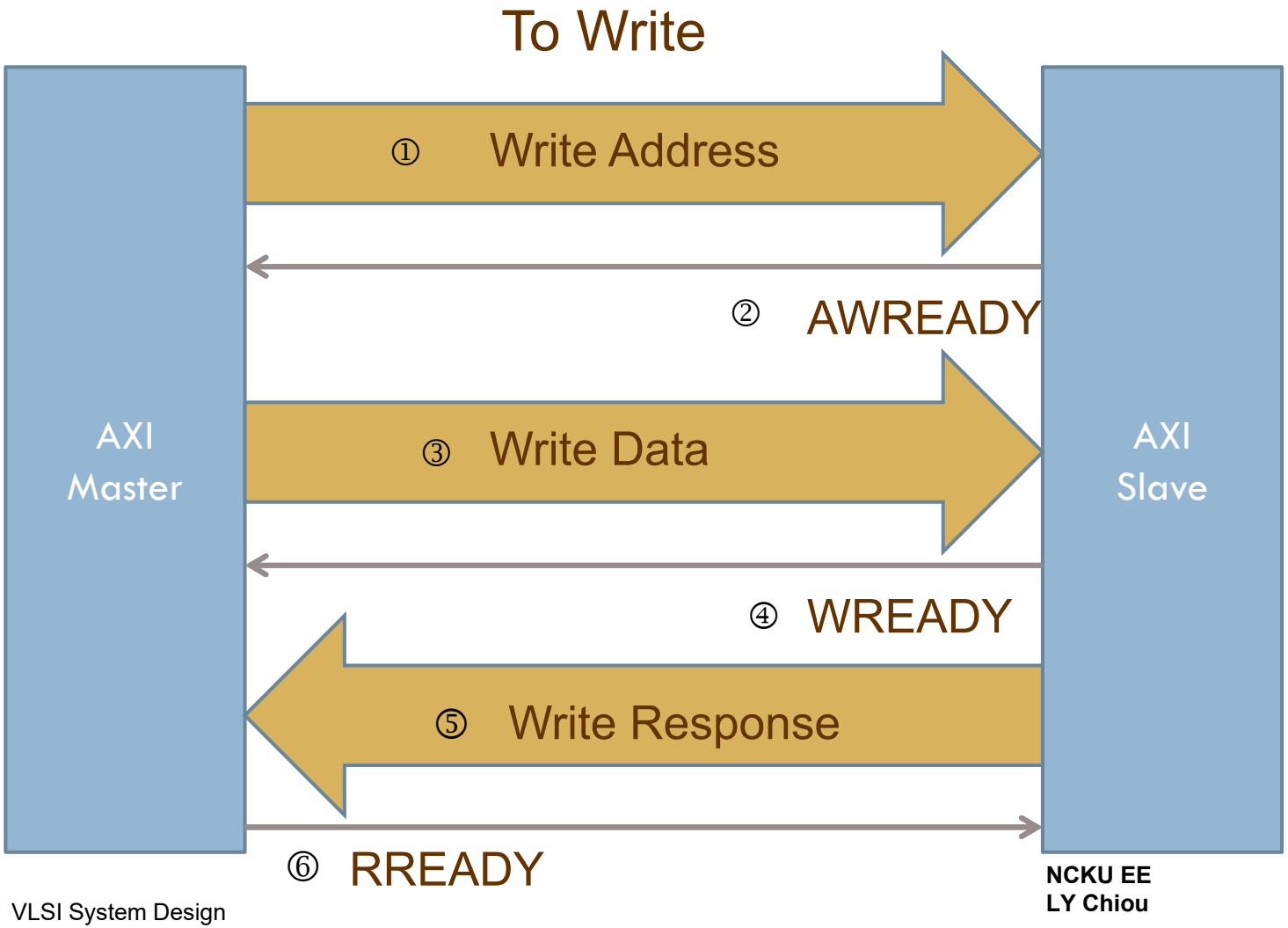
A Simple Transfer Example –

Overlapping read burst

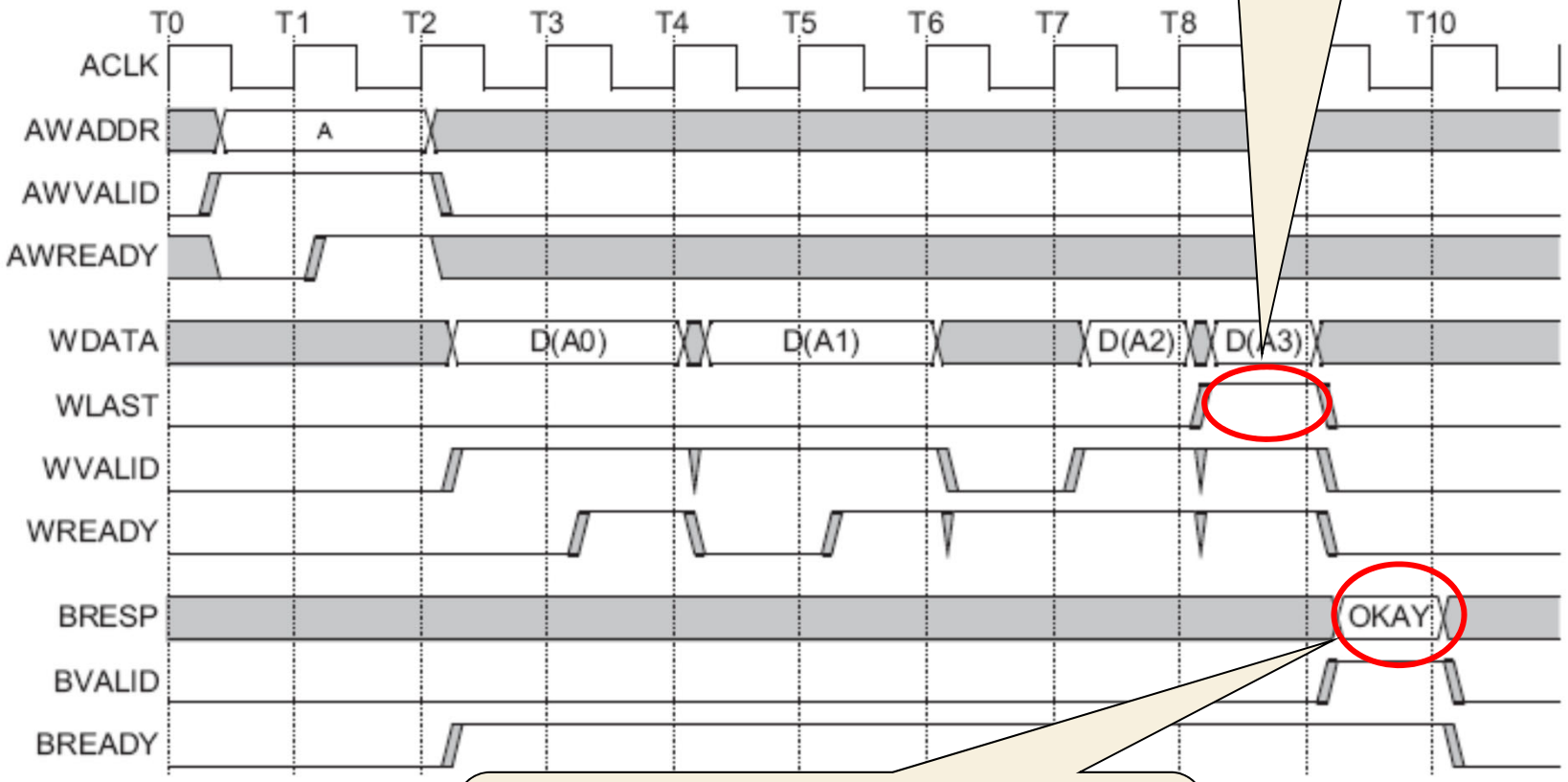
- Master can drive another burst address after the slave accepts the first address



Write Transaction



A Simple Transfer Example –
Write burst



Master indicate the last write transfer

Slave write response always follow the last write transfer

NCKU EE
LY Chiou

Outline

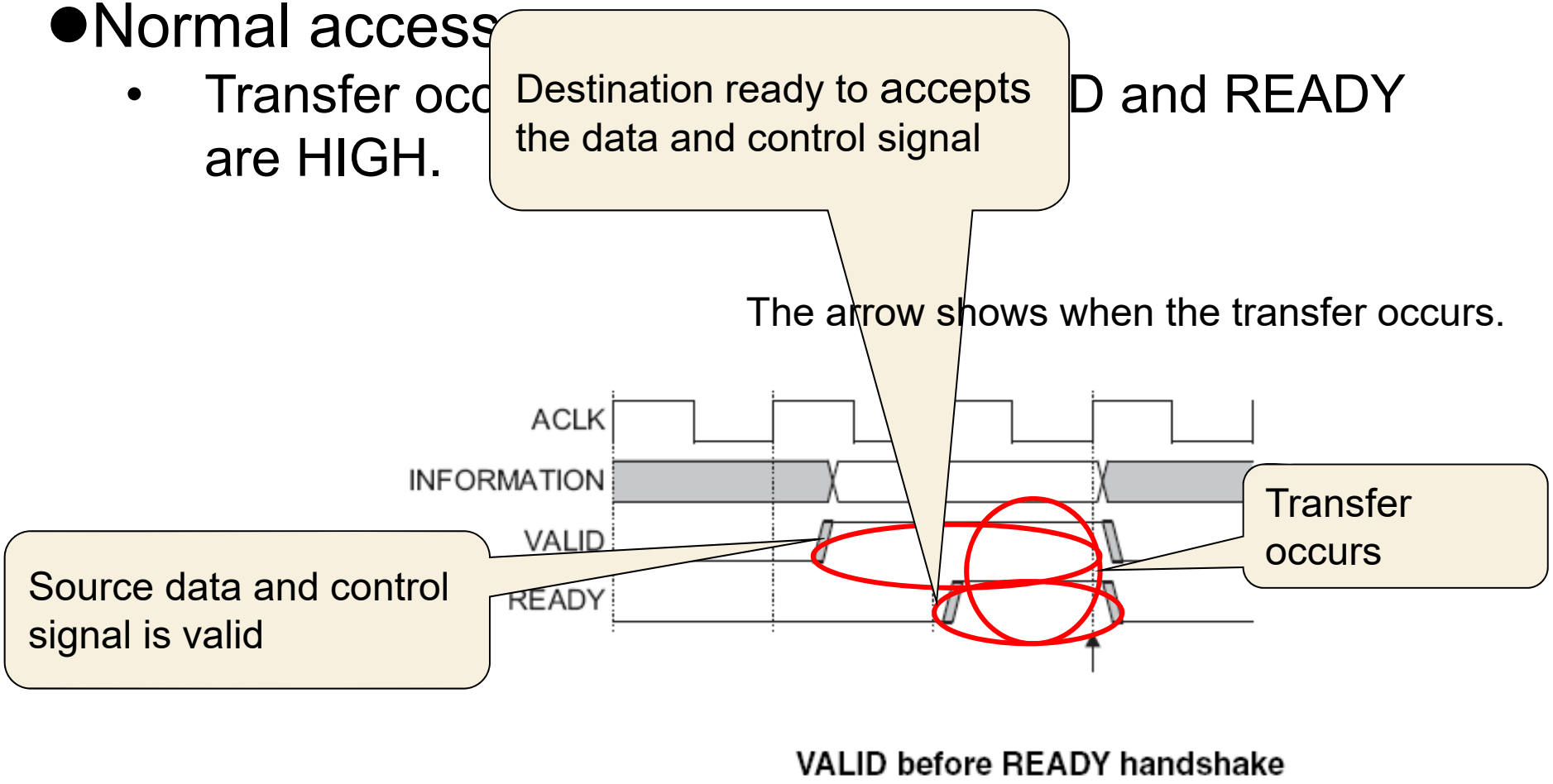
- Comparison between the AXI and the AHB
- AXI Introduction
- Basic transfer examples
- Channel handshake
- Additional features

Handshake process

- All five channels use the same VALID/READY handshake to transfer data and control information
- The source generates the VALID signal to indicate data or control information is available.
- The destination generates the READY signal to indicate that it can accepts the data or control information.
- Transfer occurs only when both the VALID and READY signals are HIGH.

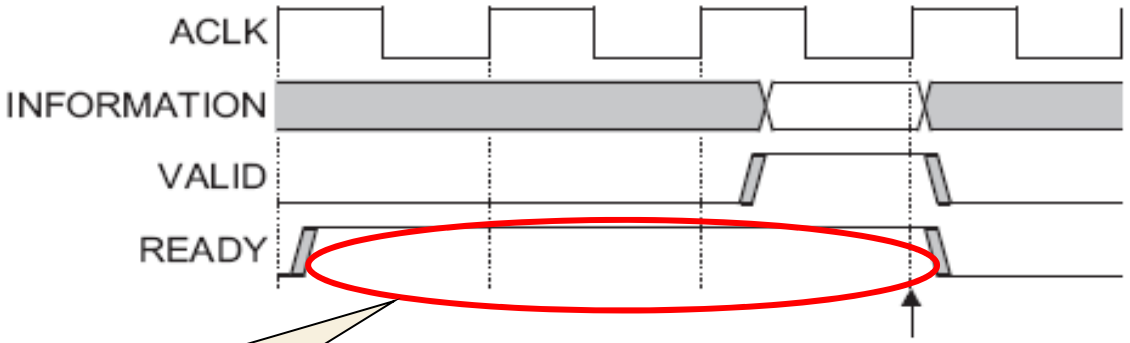
Handshake process - example

- Normal access
  - Transfer occ D and READY are HIGH.



Handshake process - example(con't)

- Fast access
  - **Recommend READY is HIGH to indicate the destination is ready to receive data**



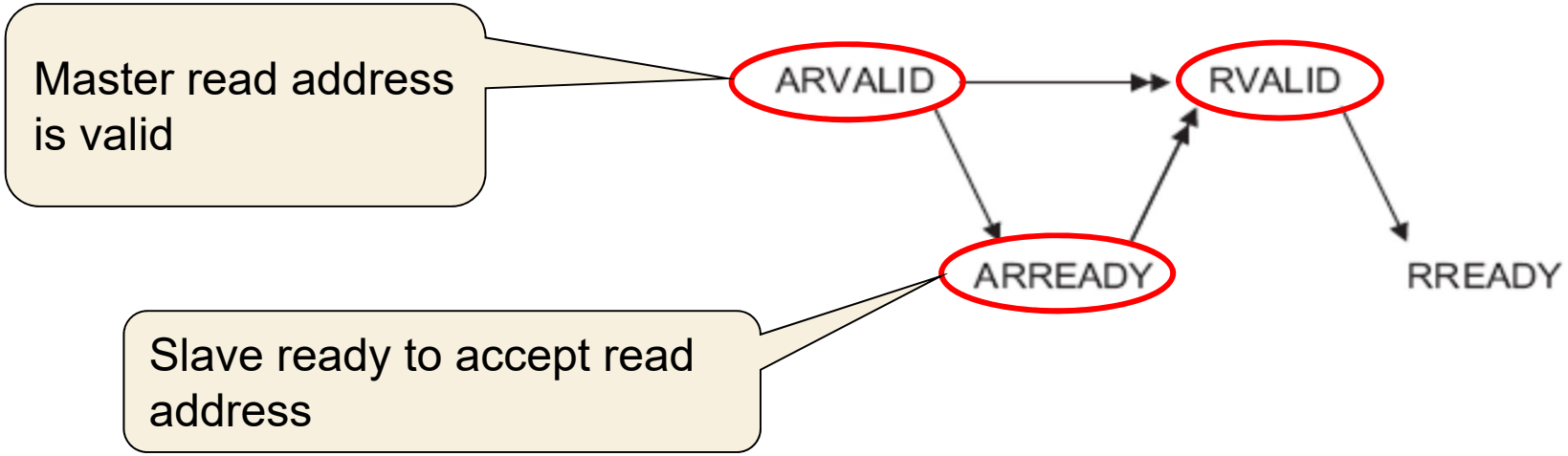
Destination can receive data early

READY before VALID handshake

The destination can accept the data or control information in a single cycle as soon as it becomes valid.

Read transaction handshake dependencies

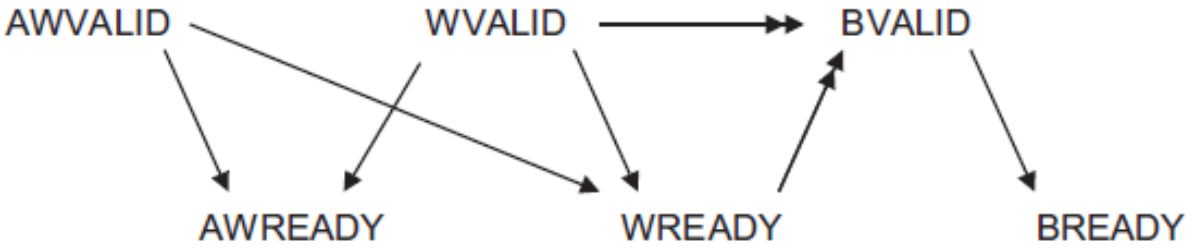
- The slave can wait for ARVALID to be asserted before it asserts ARREADY
- The slave must wait for both ARVALID and ARREAD to be asserted before it starts to return read data by asserting RVALID.



Read data always come after Read address

Write transaction handshake dependencies

- The master must not wait for the slave to assert AWREADY or WREADY before asserting AWVALID or WVALID
- The slave can wait for AWVALID or WVALID, or both, before it asserts AWREADY
- The slave can wait for AWVALID or WVALID, or both, before it asserts WREADY
- The slave must wait for AWVALID or WVALID, or both, before it asserts BVALID



Write transaction handshake dependencies (Cont'd)

- Write data can appear before Write address
- Write data appear in the same cycle as address
- Write response always come after Write data

Outline

- Comparison between the AXI and the AHB
- AXI Introduction
- Basic transfer examples
- Channel handshake
- Additional features

Additional features

- Response signal
- Ordering model
- Atomic accesses
- Low power interface

Response signal

- Source: Slave
- response signals for read and write transactions.

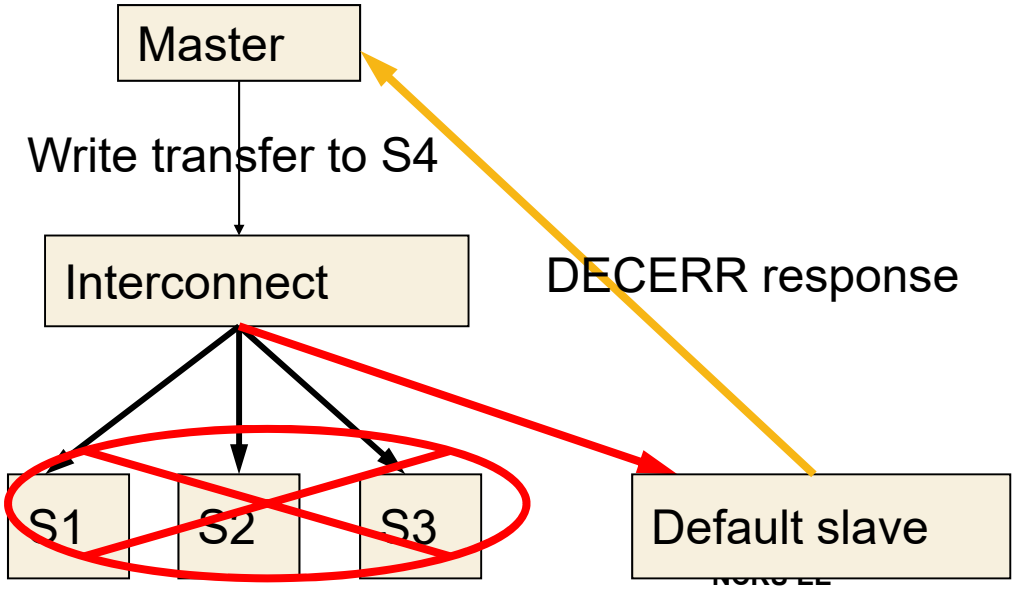
RRESP[1:0]

BRESP[1:0]

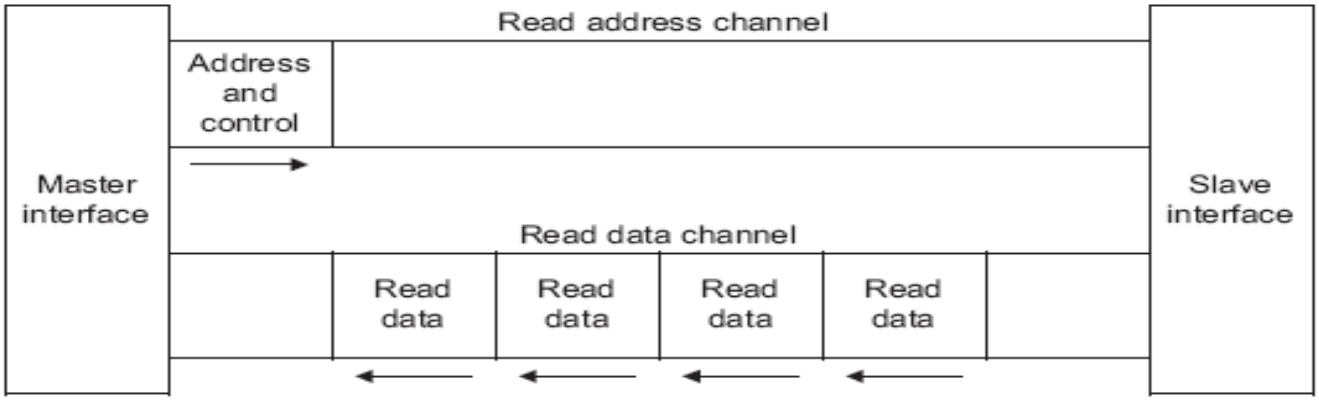
Response	Meaning
b00 OKAY	normal access has been successful. Or exclusive access failure.
b01 EXOKAY	exclusive access has been successful.
b10 SLVERR	indicates unsuccessful transaction.
b11 DECERR	indicate that there is no slave at the transaction address.

Response signal(con't)

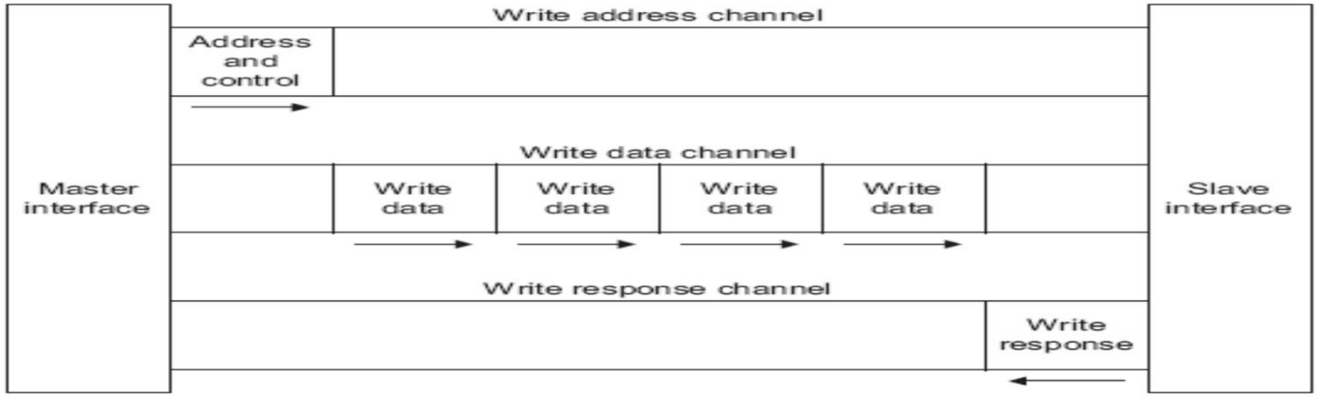
- The required number of data must be performed, even if an error is reported.
- Default slave
  - When the interconnect cannot successfully decode a slave access , It routes the access to default slave, and default slave return DECERR response



Difference between read and write response signal



- Can give different responses for different transfer within a burst

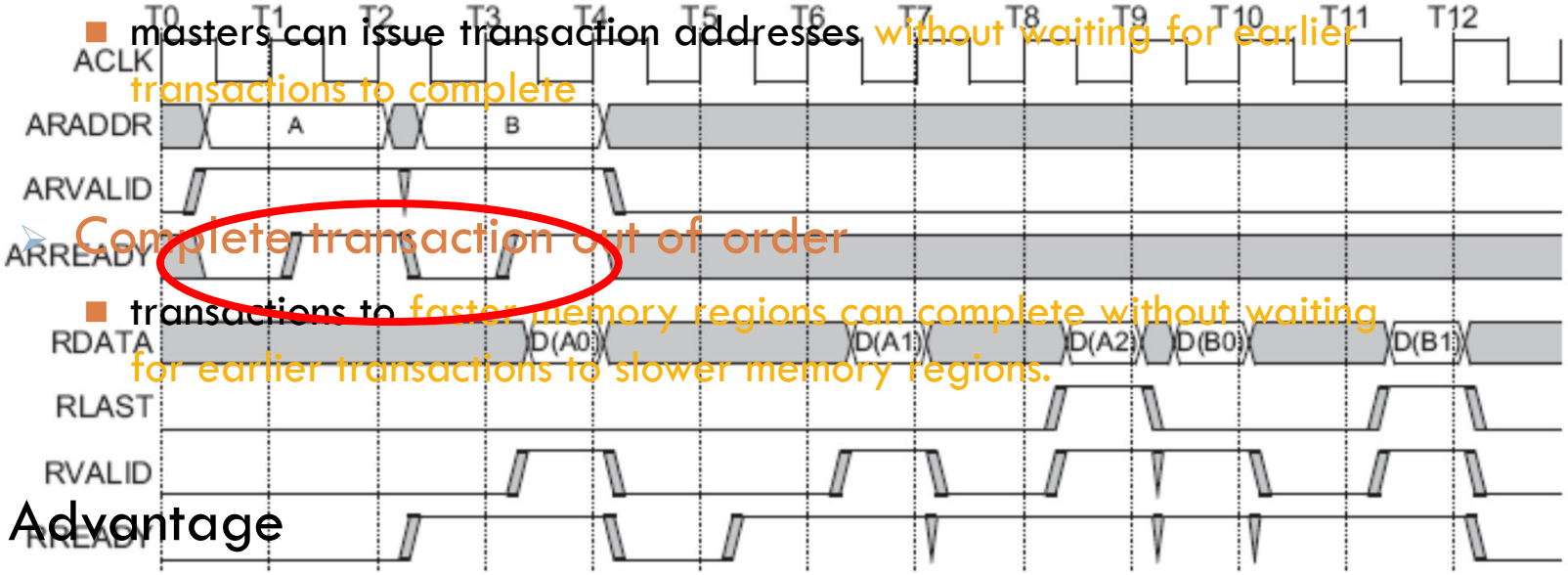


Additional features

- Response signal
- Ordering model
- Atomic accesses
- Low power interface

Ordering model

- AXI ability
  - Issue multiple outstanding addresses
    - masters can issue transaction addresses without waiting for earlier transactions to complete
  - Complete transaction out of order
    - transactions to faster memory regions can complete without waiting for earlier transactions to slower memory regions.



- Advantage
  - High performance interconnect
  - Maximizing data throughput

Ordering model – transfer ID fields

- The ARID or AWID field of a transaction
  - **provide additional information about the ordering requirements of the master.**
    - Write transactions with the same AWID value must complete in order.
    - Reads with the same ARID are from the same slave then the slave must ensure that the read data returns in order.
- Complete transactions out of order
  - **Transactions from the same master, but with different ID values, can complete in any order.**

Ordering model – Write data interleaving

- Write data interleaving enables a slave interface to accept interleaved write data with different AWID values.
- master interface can interleave write data with different WID values if the slave interface has a write data interleaving depth greater than one.
- Write data interleaving can prevent stalling(deadlock) and improve system performance.

Additional features

- Response signal
- Ordering model
- Atomic accesses
- Low power interface

Atomic accesses - signal

- Source: Master
  - Atomic access encoding

ARLOCK[1:0] AWLOCK[1:0]	Access type
b00	Normal access
b01	Exclusive access
b10	Locked access
b11	Reserved

- Source: slave
  - **RRESP[1:0]** or **BRESP[1:0]** signal indicates the success or failure of the exclusive access.
    - EXOKAY
    - OKAY

Atomic access

- Locked access
  - Only the master is allowed access to the slave until an locked transfer from the same master complete
  - Disadvantage
    - Impact on the interconnect performance
- Exclusive access
  - without requiring the bus to remain locked to a particular master for the duration of the operation.
  - Advantage
    - do not impact either the critical bus access latency or the maximum achievable bandwidth.

Atomic access – exclusive access flow chart