Digital Interfaces

Lecture 5

System Design at Interface

- Important for system architect to plan how a chip will communicate externally
  - Also important to determine how system components communicate internally
- Depending on the kind of interface, different protocol options exist
- We will examine a simple, generic options for custom ASICs
  - SPI, often used for simple chip-to-chip communication
- We will also look at a more complex standard for on-chip communications
  - ARM Advanced Microcontroller Bus Architecture (AMBA)
  - Defines the AXI stream interfaces used for to communicate with Xilinx IP
    - tdata, tvalid, tready

Why Not Custom Protocol?

- Many advantages to using standard communication protocols
- It saves time
  - Re-use IP to define interface from one design to the next
- It reduces integration errors due to miscommunication
  - Dividing chip design between multiple designers is simpler when interfaces are well defined
- It makes your chip easy to integrate into customers' systems
  - No need to design controller for a bespoke communication protocol

- Originally a proprietary protocol, created by Motorola in 1979
- Now the de-facto standard for off-chip communication
  - Typically see the same set of signals, parameters
  - Not actually a controlled standard

Digital Interfaces: Off-Chip Communication

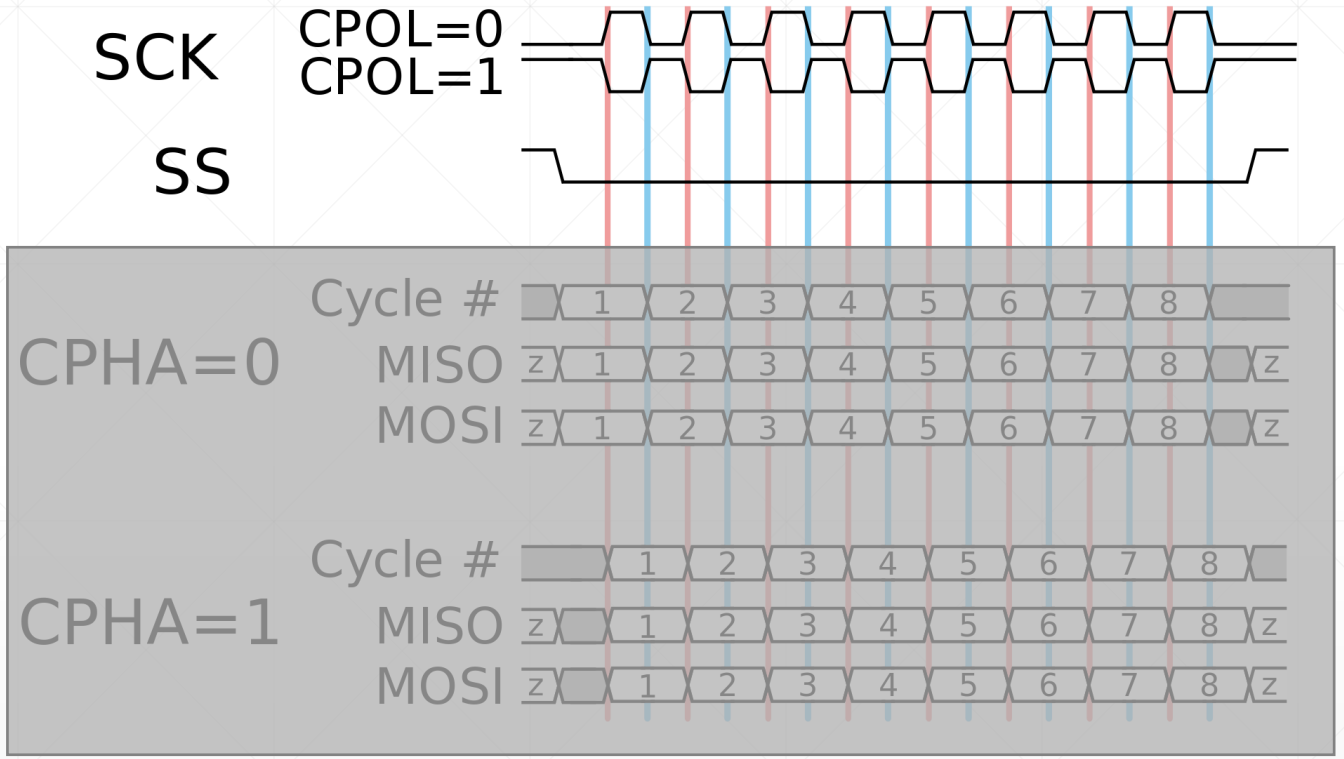
Serial Peripheral Interface

Serial Peripheral Interface: SPI

- Synchronous serial communication interface
- Capable of full duplex data transmission between master and slave
- Interface made up of 4 signals:
  - SCLK: Clock signal output from master
  - MOSI: Master Out Slave In (1 bit data)
  - MISO: Master In Slave Out (1 bit data)
  - SS_n: Slave select (enable for the slave)
- One master, many slaves
  - Signals shared except for SS_n

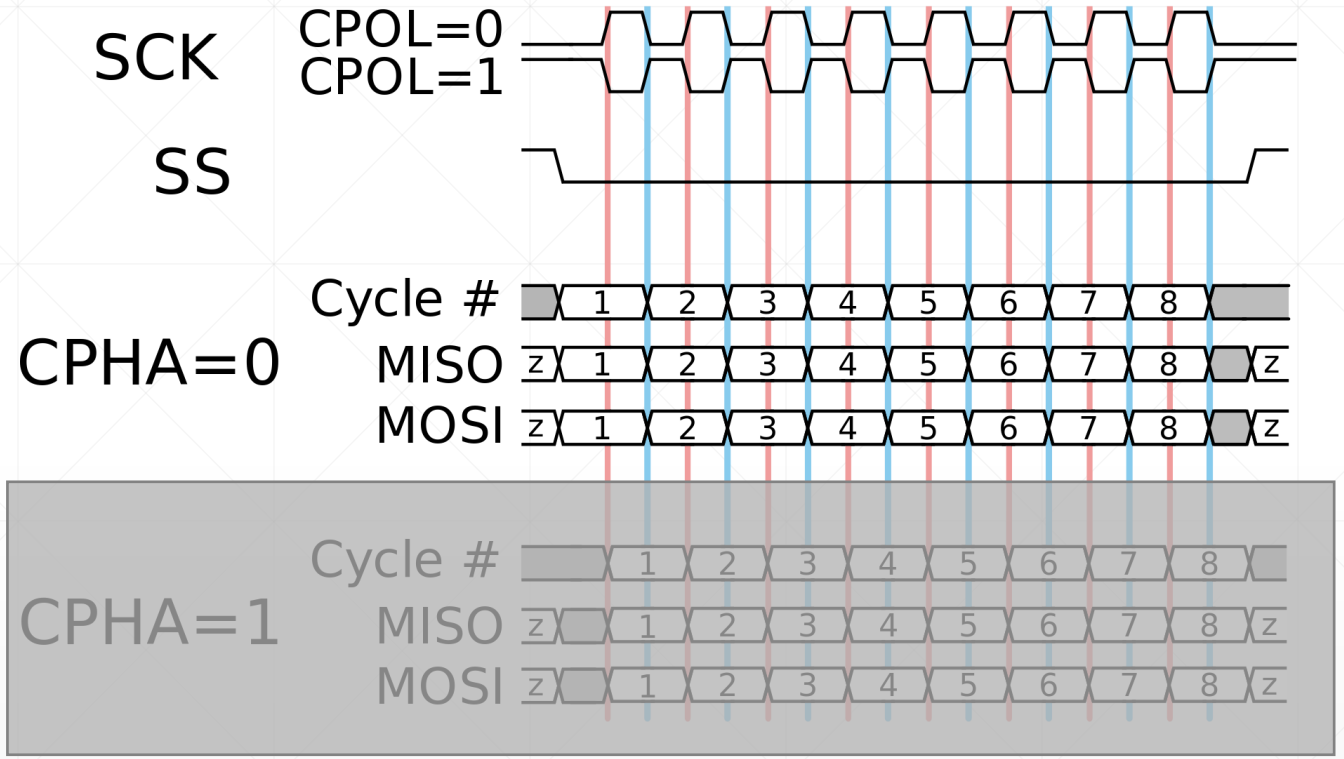
SPI Parameterisation

- Two properties of interface must be defined…
- CPOL: Clock polarity in idle state
- CPHA:
  - Determines the phase of the data change relative to the clock edge
  - CPHA=0: Change data on even clock edges, sample on odd
  - CPHA=1: Change data on odd clock edges, sample on even



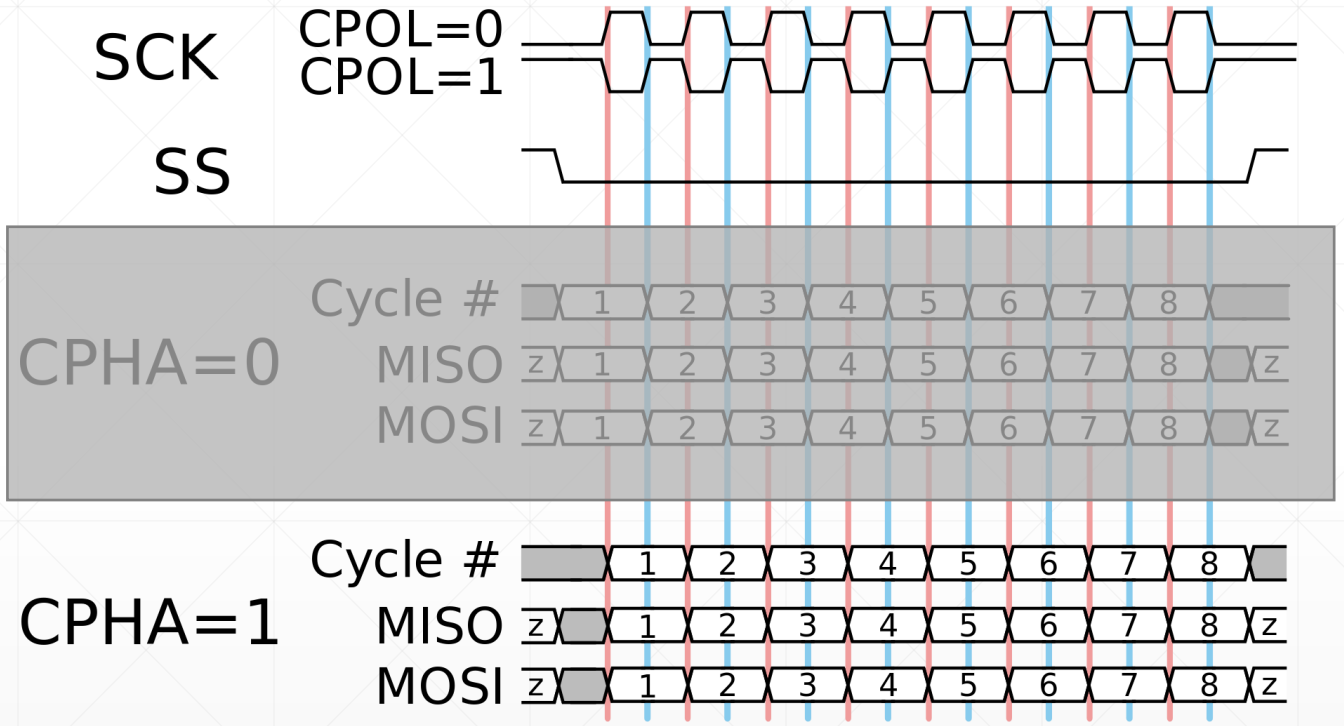
SPI Parameterisation

- Two properties of interface must be defined…
- CPOL: Clock polarity in idle state
- CPHA:
  - Determines the phase of the data change relative to the clock edge
  - CPHA=0: Change data on even clock edges, sample on odd
  - CPHA=1: Change data on odd clock edges, sample on even



SPI Parameterisation

- Two properties of interface must be defined…
- CPOL: Clock polarity in idle state
- CPHA:
  - Determines the phase of the data change relative to the clock edge
  - CPHA=0: Change data on even clock edges, sample on odd
  - CPHA=1: Change data on odd clock edges, sample on even
- Other variations exist as it is not an official standard…



- AMBA is an **open** standard for on-chip interconnects
  - Defined and licensed by ARM
  - Provides standard set of signals for communicating between different parts of an SoC
  - No fee for licensing, though does still involve legal agreement
- AXI is one set of bus standards defined within AMBA

Digital Interfaces: On-Chip Communication

Advanced eXtensible Interface

Advanced Microcontroller Bus Architecture (AMBA): Advanced eXtensible Interface (AXI)

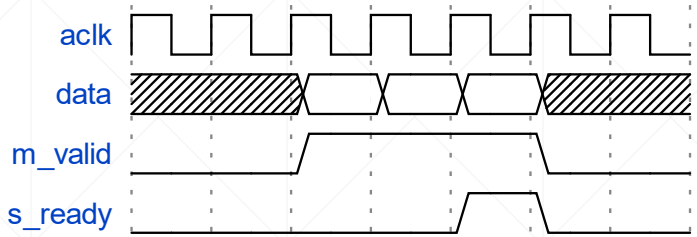
- Standard option for SoC bus architectures
  - Interface for many Xilinx IP blocks
- AMBA contains AXI, AXI-Lite, AXI-Stream
  - AXI provides address based control (read and write channels)
  - AXI-Lite is simpler version of this
  - AXI-Stream is for dedicated datapath connection, with single direction of data flow
- We will focus on AXI4-Stream
  - Not address based
  - Still uses the same handshake mechanism as AXI and AXI-Lite
  - This interface was used during the FIR implementation lab

AXI Handshake

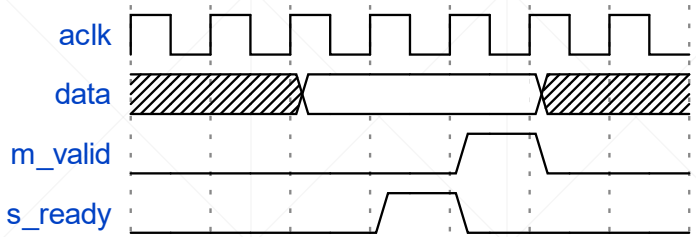
- AXI protocols use a handshake mechanism
- Uses “ready” and “valid” signals
- Master uses valid to say data is available
  - Must not wait for ready before asserting valid
  - Must hold valid high, and data steady, until it sees ready
- Slave uses ready to say it can capture the data
- When both ready and valid are high for a clock cycle, the data transfer will occur
- The handshake allows both Master and Slave to control flow of data
  - Ensures data is kept available until it has been captured

Poll: Which of these is an AXI compliant handshake?

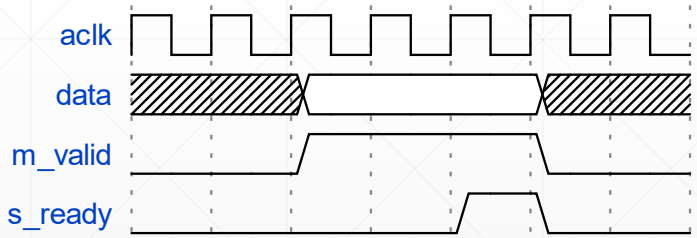
1



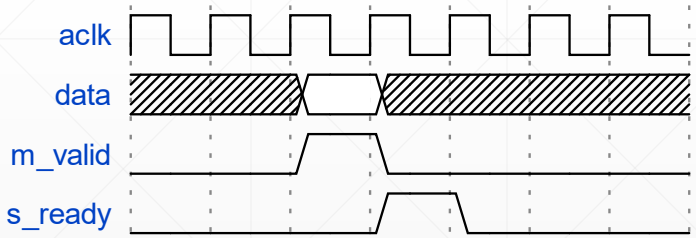
2



3



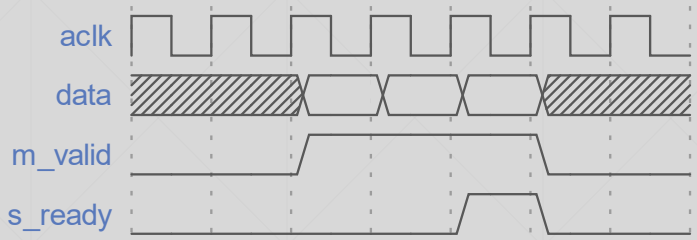
4



Drawn using https://wavedrom.com/editor.html

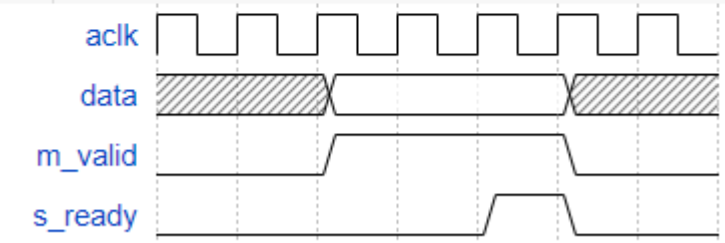
Poll: Which of these is an AXI compliant handshake?

1

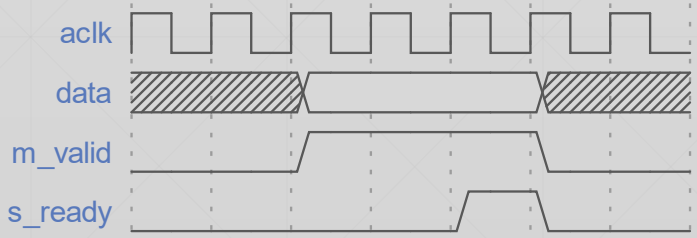


WaveDrom Editor

```
{signal: [
  {name: 'aclk',    wave: 'p.....'},
  {name: 'data',    wave: 'xx=..x.'},
  {name: 'm_valid', wave: '0.1..0.'},
  {name: 's_ready', wave: '0...10.'}
]}
```



3



Drawn using https://wavedrom.com/editor.html

AXI4 Stream

- Streaming data interface
  - Data sent **from** Master, **to** slave
- Signals used can vary
  - Select subset of AXI stream signals
- Basic stream interface includes:
  - aclk, aresetn, tvalid, tready, tdata
  - Width of tdata is always integer number of bytes
- May also include
  - tkeep, tid, tlast…
  - tuser and several others

AXI4 Stream Example Question

- Draw a set of waveforms for an AXI4 Stream transaction.
- Assume the following:
  - The master has three 6-bit words to send
  - The slave will take two clock cycles to process any data it receives
- What would change if the master could send two words at a time?
  - (Assuming the slave will accept two words at once, and still take two clock cycles to process the data)
- What would change if the master had three 10-bit words to send?
  - If two words at a time, how many bits in tkeep?

AXI4 and AXI4-Lite

- AXI4 and AXI4-Lite are address based interfaces
  - AXI4-Lite is a simplified version of AXI4 suitable for control register interfaces
- Many more signals on the interface
  - Read address, read data, write address, write response channels
  - 5-10 different signals per channel…
- All channels follow the same handshake principle we have explored with AXI Stream

AXI-STREAM signals

2.1 Signal list

The interface signals are listed in Table 2-1. For additional information on these signals see further sections of this chapter.

Table 2-1 uses the following parameters to define the signal widths:

n	Data bus width in bytes.
i	**TID** width. Recommended maximum is 8-bits.
d	**TDEST** width. Recommended maximum is 4-bits.
u	**TUSER** width. Recommended number of bits is an integer multiple of the width of the interface in bytes.

Table 2-1 Interface signals list

Signal	Source	Description
ACLK	Clock source	The global clock signal. All signals are sampled on the rising edge of **ACLK**.
ARESETn	Reset source	The global reset signal. **ARESETn** is active-LOW.
TVALID	Master	**TVALID** indicates that the master is driving a valid transfer. A transfer takes place when both **TVALID** and **TREADY** are asserted.
TREADY	Slave	**TREADY** indicates that the slave can accept a transfer in the current cycle.
TDATA[(8n-1):0]	Master	**TDATA** is the primary payload that is used to provide the data that is passing across the interface. The width of the data payload is an integer number of bytes.
TSTRB[(n-1):0]	Master	**TSTRB** is the byte qualifier that indicates whether the content of the associated byte of **TDATA** is processed as a data byte or a position byte.
TKEEP[(n-1):0]	Master	**TKEEP** is the byte qualifier that indicates whether the content of the associated byte of **TDATA** is processed as part of the data stream. Associated bytes that have the **TKEEP** byte qualifier deasserted are null bytes and can be removed from the data stream.
TLAST	Master	**TLAST** indicates the boundary of a packet.
TID[(i-1):0]	Master	**TID** is the data stream identifier that indicates different streams of data.
TDEST[(d-1):0]	Master	**TDEST** provides routing information for the data stream.
TUSER[(u-1):0]	Master	**TUSER** is user defined sideband information that can be transmitted alongside the data stream.

https://developer.arm.com/documentation/ihi0051/a/

AXI4 Signal definitions

Write address channel signals

Table A2-2 shows the AXI write address channel signals. Unless the description indicates otherwise, a signal is used by AXI3 and AXI4.

Table A2-2 Write address channel signals

Signal	Source	Description
AWID	Master	Write address ID. This signal is the identification tag for the write address group of signals. See *Transaction ID* on page A5-77.
AWADDR	Master	Write address. The write address gives the address of the first transfer in a write burst transaction. See *Address structure* on page A3-44.
AWLEN	Master	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address. This changes between AXI3 and AXI4. See *Burst length* on page A3-44.
AWSIZE	Master	Burst size. This signal indicates the size of each transfer in the burst. See *Burst size* on page A3-45.
AWBURST	Master	Burst type. The burst type and the size information, determine how the address for each transfer within the burst is calculated. See *Burst type* on page A3-45.
AWLOCK	Master	Lock type. Provides additional information about the atomic characteristics of the transfer. This changes between AXI3 and AXI4. See *Locked accesses* on page A7-95.
AWCACHE	Master	Memory type. This signal indicates how transactions are required to progress through a system. See *Memory types* on page A4-65.
AWPROT	Master	Protection type. This signal indicates the privilege and security level of the transaction, and whether the transaction is a data access or an instruction access. See *Access permissions* on page A4-71.
AWQOS	Master	*Quality of Service*, QoS. The QoS identifier sent for each write transaction. Implemented only in AXI4. See *QoS signaling* on page A8-98.
AWREGION	Master	Region identifier. Permits a single physical interface on a slave to be used for multiple logical interfaces. Implemented only in AXI4. See *Multiple region signaling* on page A8-99.
AWUSER	Master	User signal. Optional User-defined signal in the write address channel. Supported only in AXI4. See *User-defined signaling* on page A8-100.
AWVALID	Master	Write address valid. This signal indicates that the channel is signaling valid write address and control information. See *Channel handshake signals* on page A3-38.
AWREADY	Slave	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals. See *Channel handshake signals* on page A3-38.

Write data channel signals

Table A2-3 shows the AXI write data channel signals. Unless the description indicates otherwise, a signal is used by AXI3 and AXI4.

Table A2-3 Write data channel signals

Signal	Source	Description
WID	Master	Write ID tag. This signal is the ID tag of the write data transfer. Supported only in AXI3. See *Transaction ID* on page A5-77.
WDATA	Master	Write data.
WSTRB	Master	Write strobes. This signal indicates which byte lanes hold valid data. There is one write strobe bit for each eight bits of the write data bus. See *Write strobes* on page A3-49.
WLAST	Master	Write last. This signal indicates the last transfer in a write burst. See *Write data channel* on page A3-39.
WUSER	Master	User signal. Optional User-defined signal in the write data channel. Supported only in AXI4. See *User-defined signaling* on page A8-100.
WVALID	Master	Write valid. This signal indicates that valid write data and strobes are available. See *Channel handshake signals* on page A3-38.
WREADY	Slave	Write ready. This signal indicates that the slave can accept the write data. See *Channel handshake signals* on page A3-38.

Write response channel signals

Table A2-4 shows the AXI write response channel signals. Unless the description indicates otherwise, a signal is used by AXI3 and AXI4.

Table A2-4 Write response channel signals

Signal	Source	Description
BID	Slave	Response ID tag. This signal is the ID tag of the write response. See *Transaction ID* on page A5-77.
BRESP	Slave	Write response. This signal indicates the status of the write transaction. See *Read and write response structure* on page A3-54.
BUSER	Slave	User signal. Optional User-defined signal in the write response channel. Supported only in AXI4. See *User-defined signaling* on page A8-100.
BVALID	Slave	Write response valid. This signal indicates that the channel is signaling a valid write response. See *Channel handshake signals* on page A3-38.
BREADY	Master	Response ready. This signal indicates that the master can accept a write response. See *Channel handshake signals* on page A3-38.

Read address channel signals

Table A2-5 shows the AXI read address channel signals. Unless the description indicates otherwise, a signal is used by AXI3 and AXI4.

Table A2-5 Read address channel signals

Signal	Source	Description
ARID	Master	Read address ID. This signal is the identification tag for the read address group of signals. See *Transaction ID* on page A5-77.
ARADDR	Master	Read address. The read address gives the address of the first transfer in a read burst transaction. See *Address structure* on page A3-44.
ARLEN	Master	Burst length. This signal indicates the exact number of transfers in a burst. This changes between AXI3 and AXI4. See *Burst length* on page A3-44.
ARSIZE	Master	Burst size. This signal indicates the size of each transfer in the burst. See *Burst size* on page A3-45.
ARBURST	Master	Burst type. The burst type and the size information determine how the address for each transfer within the burst is calculated. See *Burst type* on page A3-45.
ARLOCK	Master	Lock type. This signal provides additional information about the atomic characteristics of the transfer. This changes between AXI3 and AXI4. See *Locked accesses* on page A7-95.
ARCACHE	Master	Memory type. This signal indicates how transactions are required to progress through a system. See *Memory types* on page A4-65.
ARPROT	Master	Protection type. This signal indicates the privilege and security level of the transaction, and whether the transaction is a data access or an instruction access. See *Access permissions* on page A4-71.
ARQOS	Master	*Quality of Service*, QoS. QoS identifier sent for each read transaction. Implemented only in AXI4. See *QoS signaling* on page A8-98.
ARREGION	Master	Region identifier. Permits a single physical interface on a slave to be used for multiple logical interfaces. Implemented only in AXI4. See *Multiple region signaling* on page A8-99.
ARUSER	Master	User signal. Optional User-defined signal in the read address channel. Supported only in AXI4. See *User-defined signaling* on page A8-100.
ARVALID	Master	Read address valid. This signal indicates that the channel is signaling valid read address and control information. See *Channel handshake signals* on page A3-38.
ARREADY	Slave	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals. See *Channel handshake signals* on page A3-38.

- https://developer.arm.com/documentation/ihi0022/hc/?lang=en

AXI4 Signal definitions

Read data channel signals

Table A2-6 shows the AXI read data channel signals. Unless the description indicates otherwise, a signal is used by AXI3 and AXI4.

Table A2-6 Read data channel signals

Signal	Source	Description
RID	Slave	Read ID tag. This signal is the identification tag for the read data group of signals generated by the slave. See *Transaction ID* on page A5-77.
RDATA	Slave	Read data.
RRESP	Slave	Read response. This signal indicates the status of the read transfer. See *Read and write response structure* on page A3-54.
RLAST	Slave	Read last. This signal indicates the last transfer in a read burst. See *Read data channel* on page A3-39.
RUSER	Slave	User signal. Optional User-defined signal in the read data channel. Supported only in AXI4. See *User-defined signaling* on page A8-100.
RVALID	Slave	Read valid. This signal indicates that the channel is signaling the required read data. See *Channel handshake signals* on page A3-38.
RREADY	Master	Read ready. This signal indicates that the master can accept the read data and response information. See *Channel handshake signals* on page A3-38.

Low-power interface signals

Table A2-7 shows the signals of the optional low-power interface. These signals are used by the AXI3 and AXI4 protocols.

Table A2-7 Low-power interface signals

Signal	Source	Description
CSYSREQ	Clock controller	System exit low-power state request. This signal is a request from the system clock controller for the peripheral to exit from a low-power state. See *Power-down or power-up handshake* on page A9-103.
CSYSACK	Peripheral device	Exit low-power state acknowledgement. This signal is the acknowledgement from a peripheral to a system exit low-power state request. See *Power-down or power-up handshake* on page A9-103.
CACTIVE	Peripheral device	Clock active. This signal indicates that the peripheral requires its clock signal. See *Peripheral clock required* on page A9-103.

- https://developer.arm.com/documentation/ihi0022/hc/?lang=en

AXI4-Lite

Table B1-1 AXI4-Lite interface signals

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

- https://developer.arm.com/documentation/ihi0022/hc/?lang=en

Top-Down Design vs Bottom-Up Design

- Two different approaches to planning your system architecture
- Top down
  - Start with high level view of what you want
  - Identify what sub-blocks are needed
    - Define how they will interact with each other at their interfaces
  - Further divide sub-blocks until you have set of “leaf cells” (won’t be further divided)
- Bottom up
  - Identify what building blocks are available to you
  - Build bigger cells by combining these building blocks
- In your labs, beginning your design with high-level block diagram is example of top-down approach