AXI Protocol

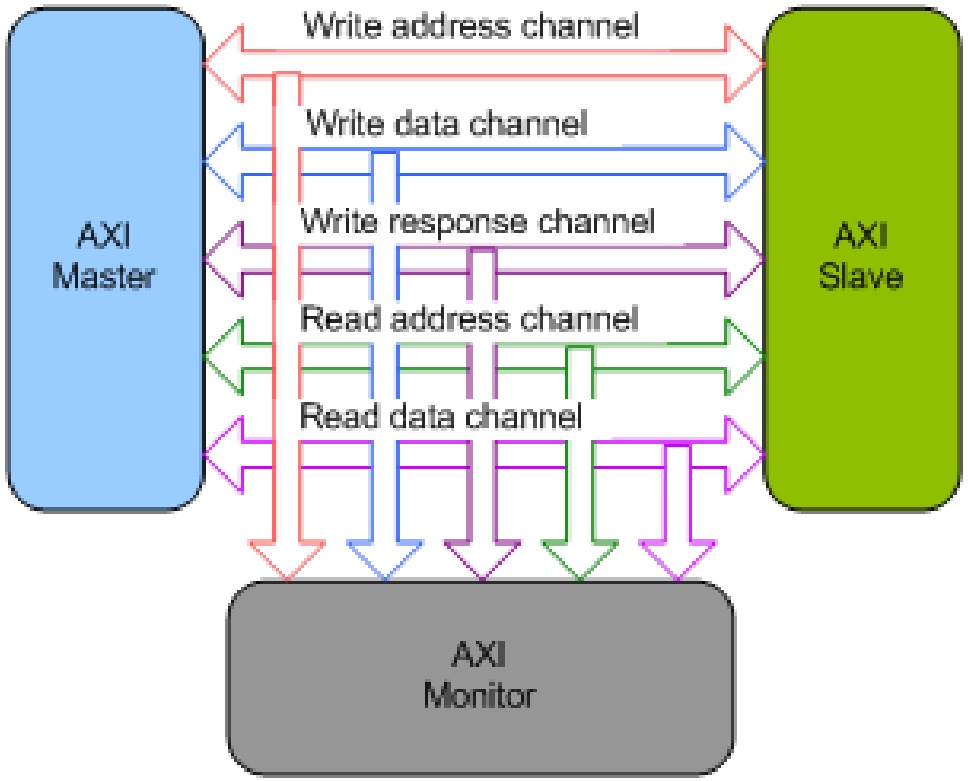
Mitesh Khadgi

For queries, contact: mitesh.khadgi2025@gmail.com

Introduction

- AXI, the third generation of AMBA interface AMBA 3 specification, is targeted at high performance, high clock frequency system designs and suitable for high speed sub-micrometer interconnect:
  separate address/control and data phases
- support for unaligned data transfers using byte strobes
- burst based transactions with only start address issued & issuing of multiple outstanding addresses
- easy addition of register stages to provide timing closure
- AXI consist of five different channels:
  Read Address Channel
- Write Address Channel
- Read Data Channel
- Write Data Channel
- Write Response Channel

5 Channels of AXI



AXI Read operation architecture:

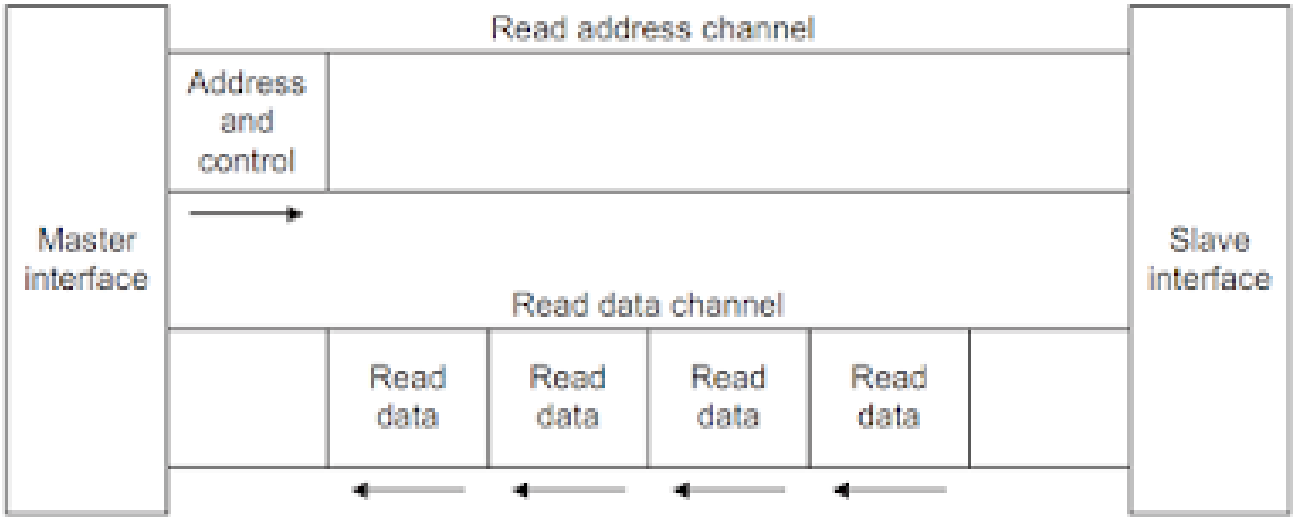


Figure 1-1 Channel architecture of reads

AXI Write operation architecture:

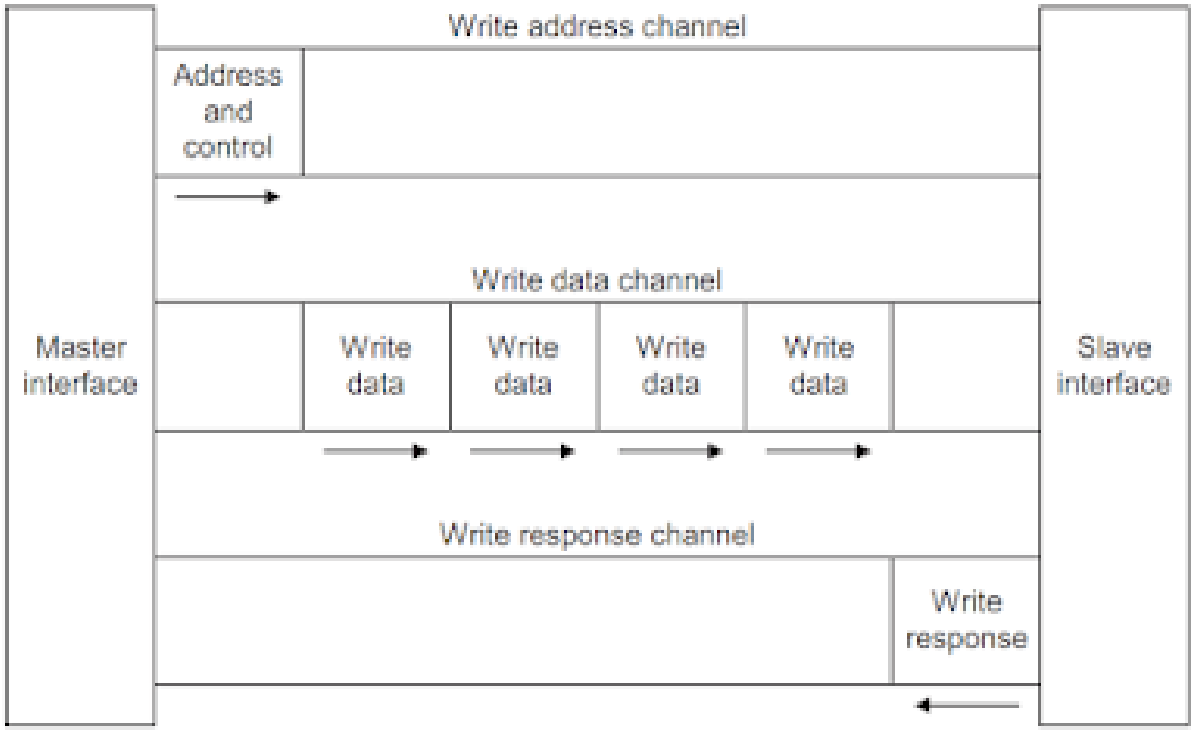


Figure 1-2 Channel architecture of writes

Transaction channel handshake pairs:

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

Handshake process:

- All five channels use the same VALID/READY handshake to transfer data and control information. This two-way flow control mechanism enables both the master and slave to control the rate at which the data and control information moves. The source generates the VALID signal to indicate when the data or control information is available. The destination generates the READY signal to indicate that it accepts the data or control information. Transfer occurs only when both the VALID and READY signals are HIGH.

  There are three scenarios when the transfer happens as shown below:

Handshaking control signals

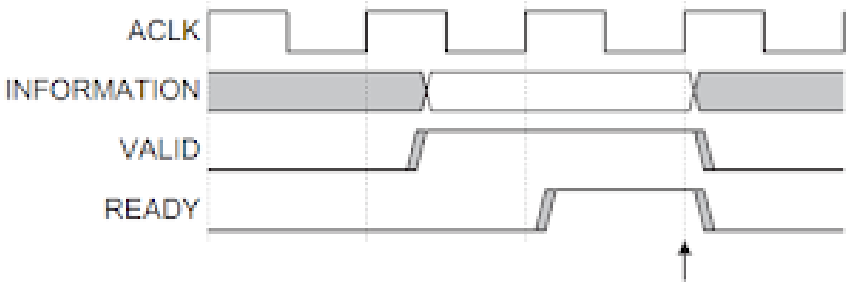


Figure 3-1 VALID before READY handshake

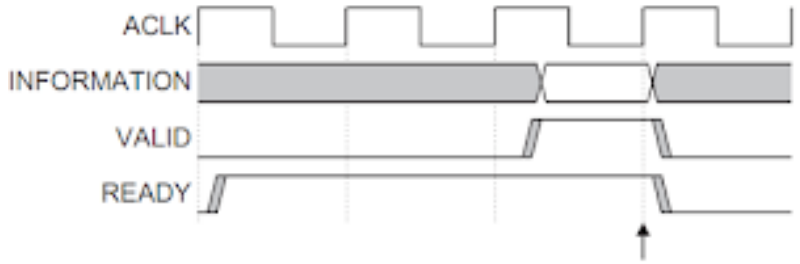


Figure 3-2 READY before VALID handshake

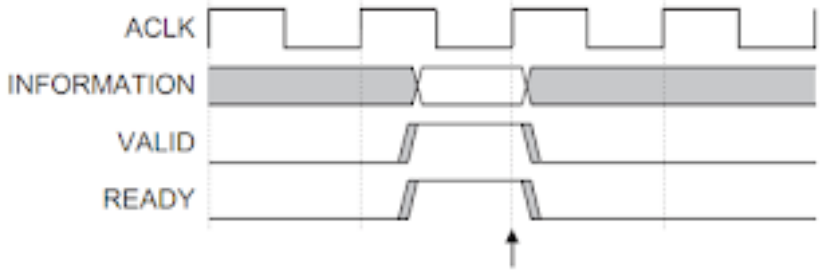


Figure 3-3 VALID with READY handshake

Signal Description:

Signal Name	Source	Description
Global Signals		
ACLK	Clock source	Global clock signal
ARESETn	Reset source	Global reset signal
Write address channel signals		
AWID[3:0]	Master	Write address ID
AWADDR[31:0]	Master	Write address
AWLEN[3:0]	Master	Burst length
AWSIZE[2:0]	Master	Burst size
AWBURST[1:0]	Master	Burst type
AWLOCK[1:0]	Master	Lock type
AWCACHE[3:0]	Master	Cache type
AWPROT[2:0]	Master	Protection type
AWVALID	Master	Write address valid
AWREADY	Slave	Write address ready
Write data channel signals		
WID[3:0]	Master	Write ID tag
WDATA[31:0]	Master	Write data
WSTRB[3:0]	Master	Write strobes
WLAST	Master	Write last
WVALID	Master	Write valid
WREADY	Slave	Write ready
Write response channel signals		
BID[3:0]	Slave	Response ID
BRESP[1:0]	Slave	Write response
BVALID	Slave	Write response valid
BREADY	Master	Response ready
Read address channel signals		
ARID[3:0]	Master	Read address ID
ARADDE[31:0]	Master	Read address
ARLEN[3:0]	Master	Burst length
ARSIZE[2:0]	Master	Burst size
ARBURST[1:0]	Master	Burst type
ARLOCK[1:0]	Master	Lock type
ARCACHE[3:0]	Master	Cache type
ARPROT[2:0]	Master	Protection type
ARVALID	Master	Read address valid
ARREADY	Slave	Read address ready

Read data channel signals

Read data channel signals		
RID[3:0]	Slave	Read ID tag
RDATA[31:0]	Slave	Read data
RRESP[1:0]	Slave	Read response
RLAST	Slave	Read last
RVALID	Slave	Read valid
RREADY	Master	Read ready

Master Slave configuration connected via interface through interconnect

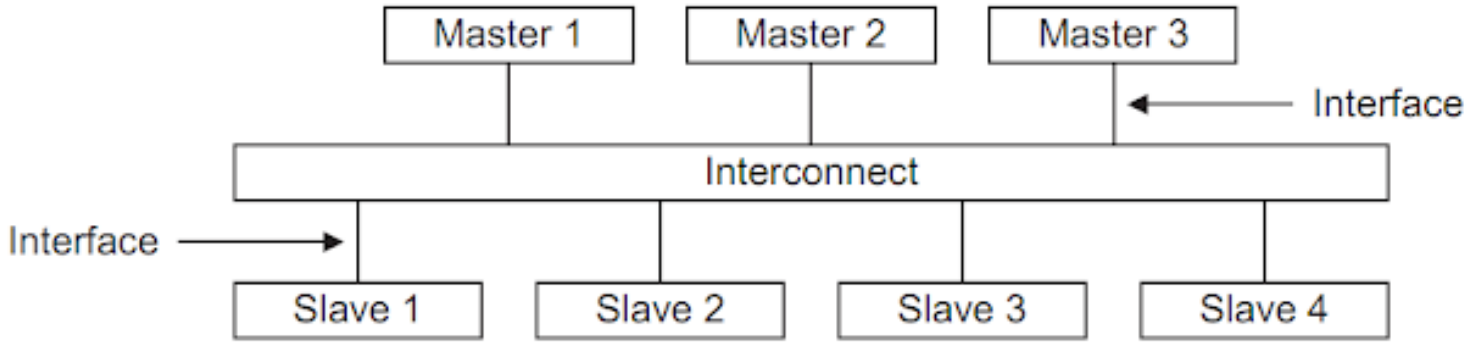


Figure 1-3 Interface and interconnect

Burst length and size

- There are different encoding techniques used in AXI for burst length, busrt size, burst type, cache encoding, protection, atomic access, response,

Table 4-1 Burst length encoding

ARLEN[3:0] AWLEN[3:0]	Number of data transfers
b0000	1
b0001	2
b0010	3
. . .	
b1101	14
b1110	15
b1111	16

Table 4-2 Burst size encoding

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

Burst Type

Table 4-3 Burst type encoding

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	-	-

Cache, Protection, and Lock

Table 5-1 Cache encoding

ARCACHE[3:0] AWCACHE[3:0]				
WA	**RA**	**C**	**B**	**Transaction attributes**
0	0	0	0	Noncacheable and nonbufferable
0	0	0	1	Bufferable only
0	0	1	0	Cacheable, but do not allocate
0	0	1	1	Cacheable and bufferable, but do not allocate
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	Cacheable write-through, allocate on reads only
0	1	1	1	Cacheable write-back, allocate on reads only
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Cacheable write-through, allocate on writes only
1	0	1	1	Cacheable write-back, allocate on writes only
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Cacheable write-through, allocate on both reads and writes
1	1	1	1	Cacheable write-back, allocate on both reads and writes

Table 5-2 Protection encoding

ARPROT[2:0] AWPROT[2:0]	Protection level
[0]	1 = privileged access 0 = normal access
[1]	1 = nonsecure access 0 = secure access
[2]	1 = instruction access 0 = data access

Table 6-1 Atomic access encoding

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

Response signals

Table 7-1 RRESP[1:0] and BRESP[1:0] encoding

RRESP[1:0] BRESP[1:0]	Response	Meaning
b00	OKAY	Normal access okay indicates if a normal access has been successful. Can also indicate an exclusive access failure.
b01	EXOKAY	Exclusive access okay indicates that either the read or write portion of an exclusive access has been successful.
b10	SLVERR	Slave error is used when the access has reached the slave successfully, but the slave wishes to return an error condition to the originating master.
b11	DECERR	Decode error is generated typically by an interconnect component to indicate that there is no slave at the transaction address.

Timing Diagrams for read burst, overlapping read burst and write burst

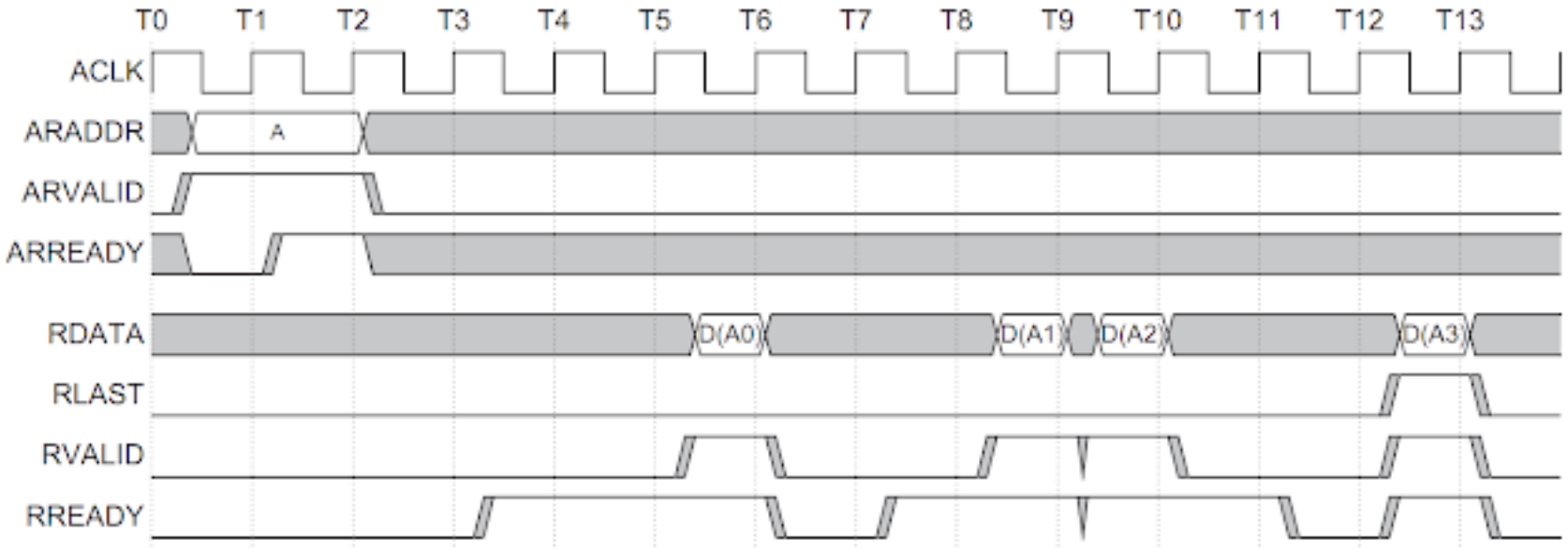


Figure 1-4 Read burst

Overlapping read bursts

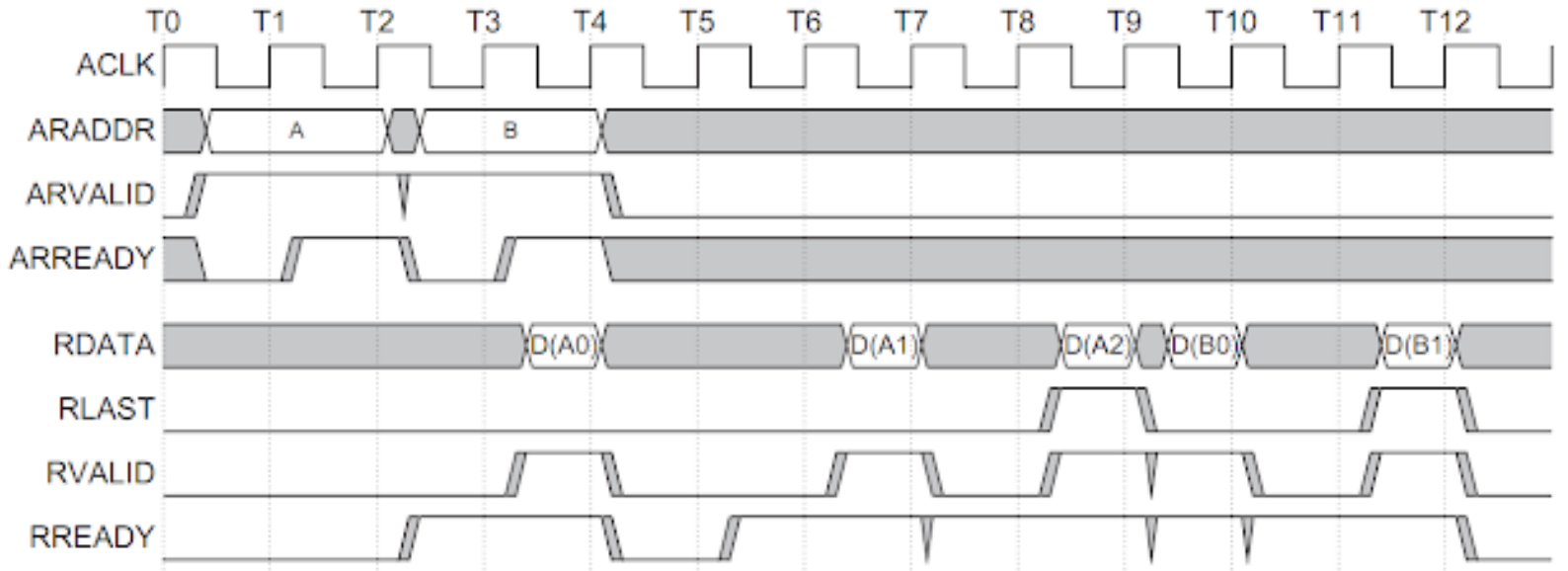


Figure 1-5 Overlapping read bursts

Write burst

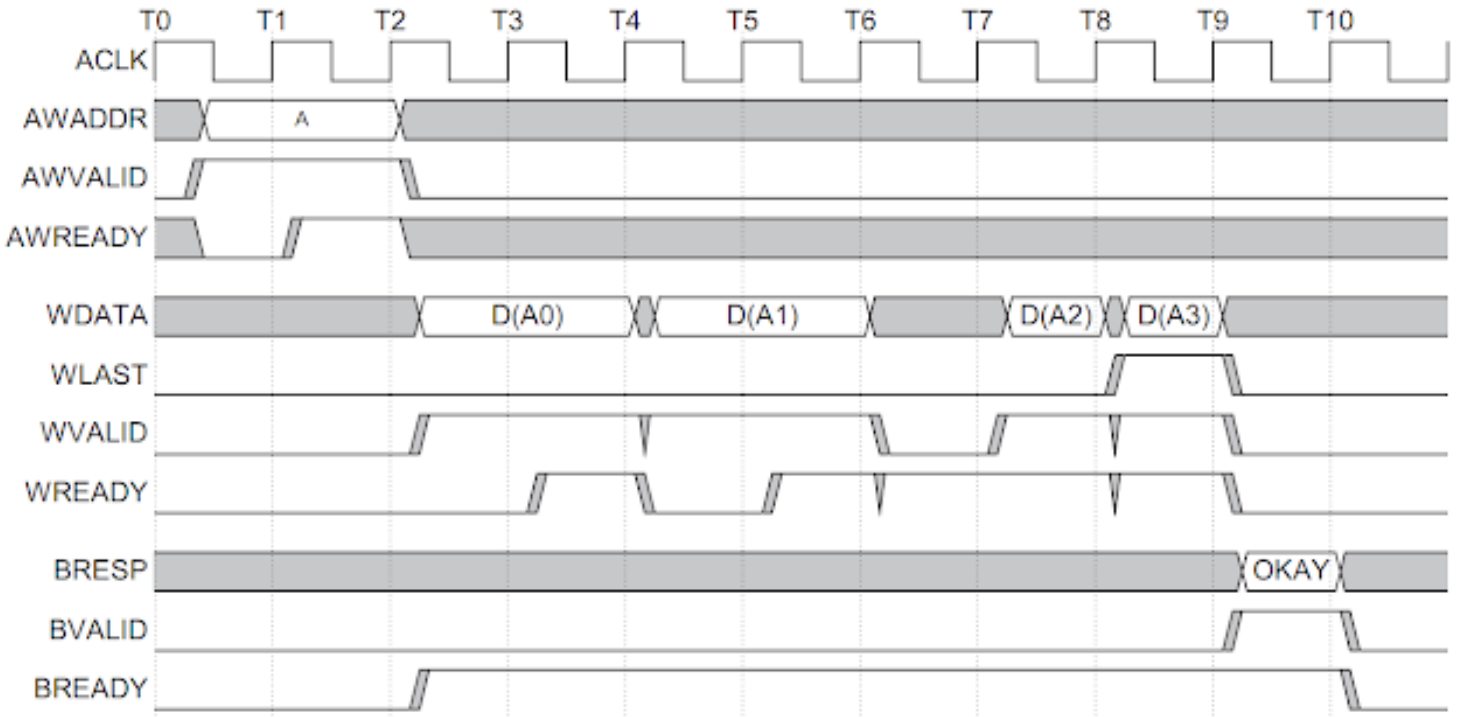


Figure 1-6 Write burst

Low-power interface signals:

Low-power interface signals		
CSYSREQ	Clock controller	System low-power request. This signal is a request from the system clock controller for the peripheral to enter a low-power state
CSYSACK	Peripheral device	Low-power request acknowledgement. This signal is the acknowledgement from a peripheral of a system low-power request
CACTIVE	Peripheral device	Clock active. This signal indicates that the peripheral requires its clock signal: 1 = peripheral clock required 0 = peripheral clock not required

Low power control signals

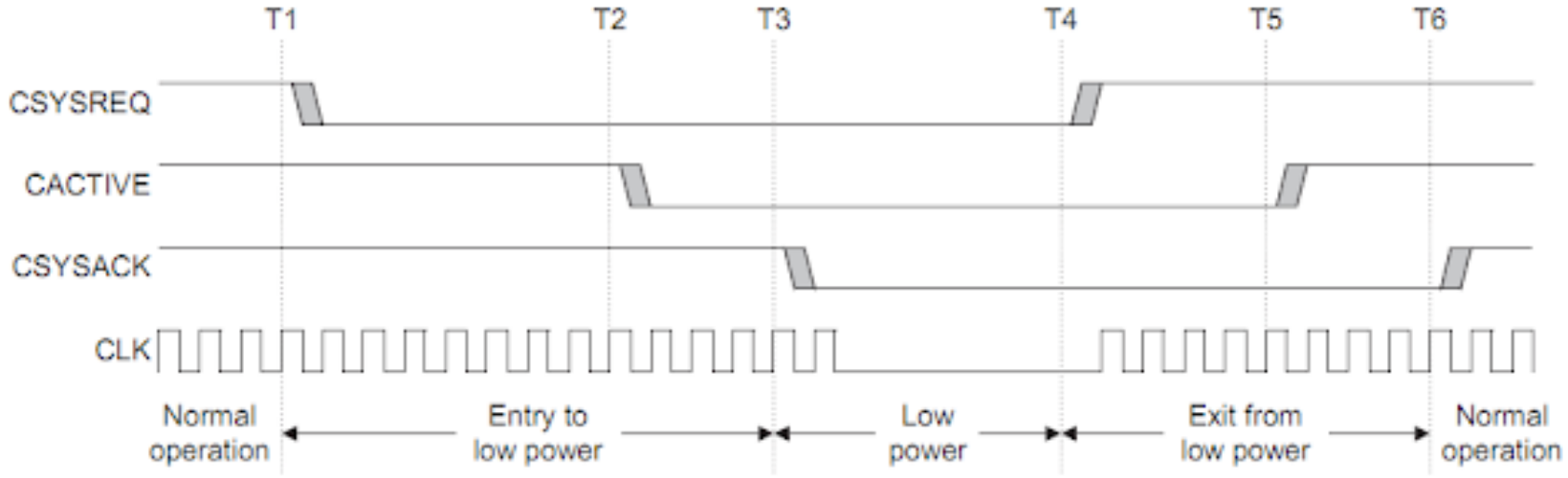


Figure 12-2 Acceptance of a low-power request

Denial of a low power request

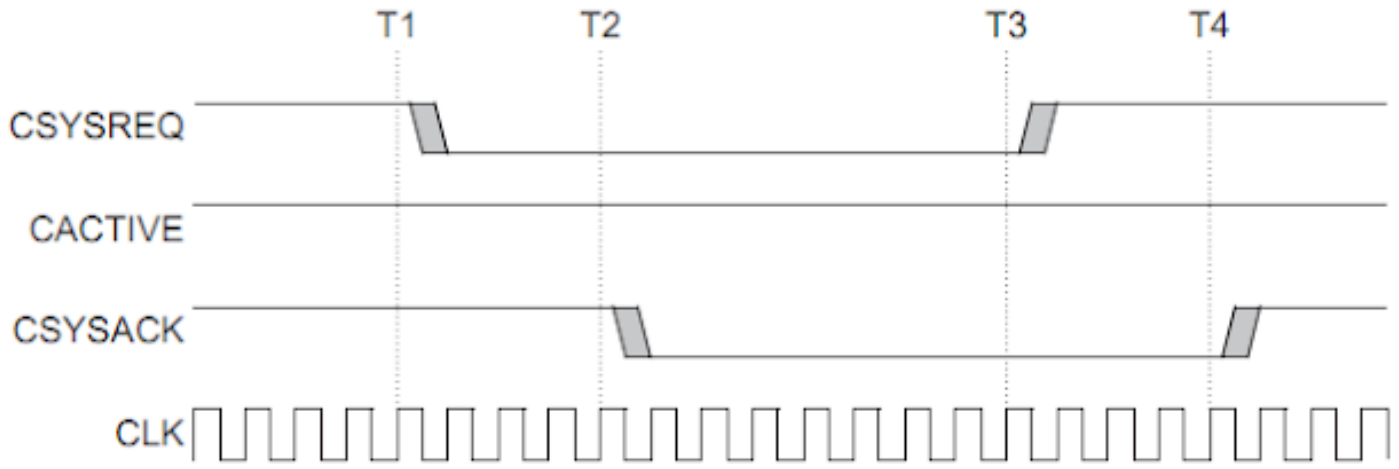


Figure 12-3 Denial of a low-power request

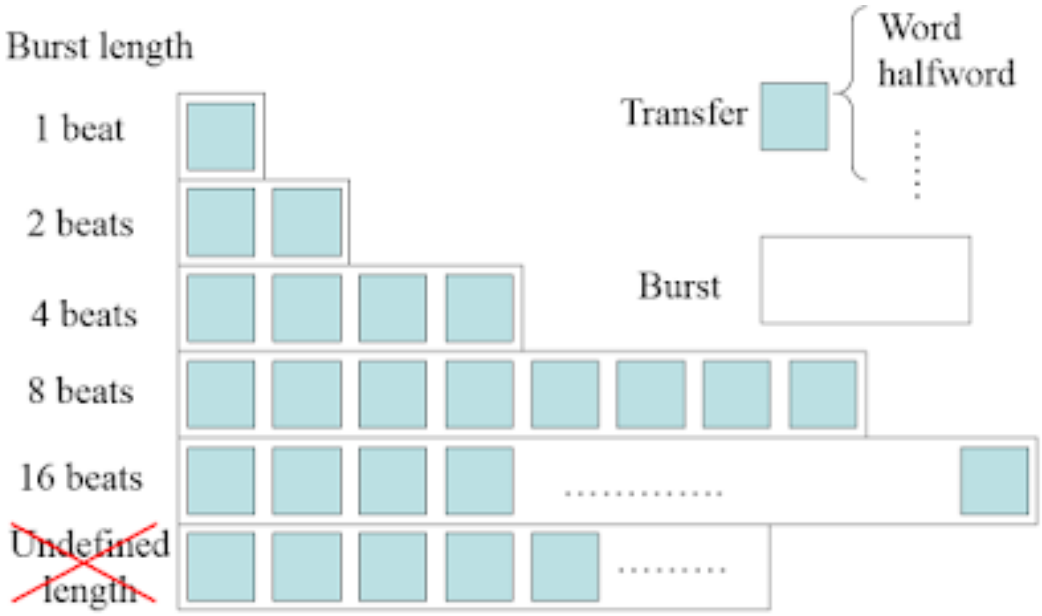
Difference between Exclusive access and Locked access

- The basic process for an exclusive access is:
  A master performs an exclusive read from an address location
- At some later time, the master attempts to complete the exclusive operation by performing an exclusive write to the same address location
- The exclusive write access of the master is signalled as:
  - Successful if no other master has written to that location between the read and write accesses
  - Failed if another master has written to that location between the read and write accesses. In this case the address location is not updated
- Locked access:

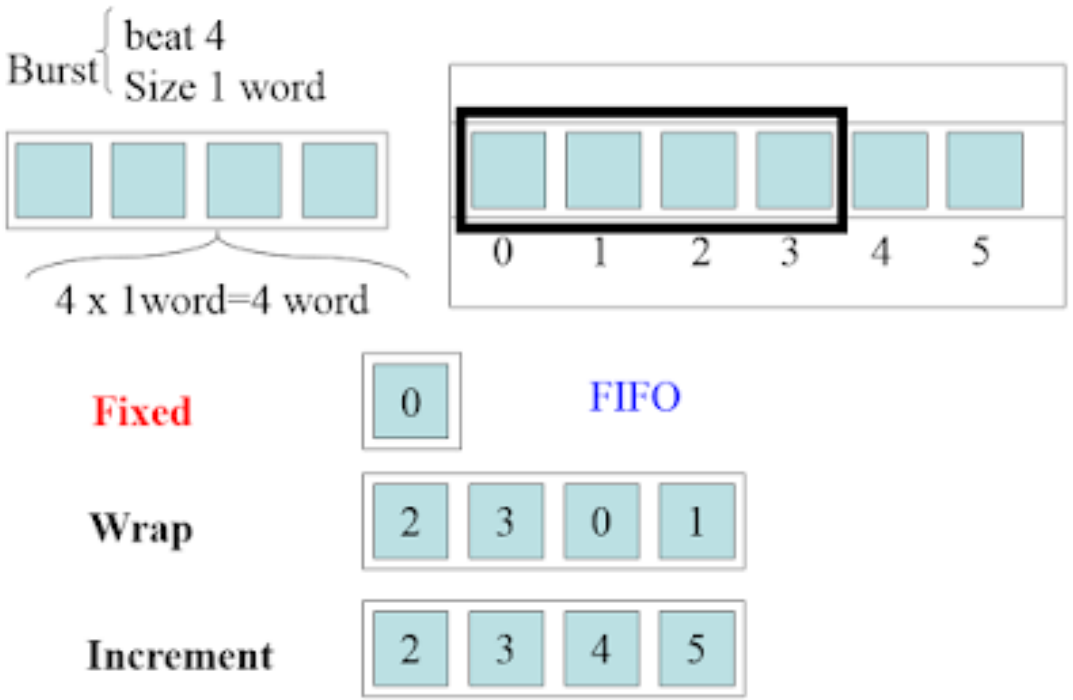
  When the ARLOCK[1:0] or AWLOCK[1:0] signals for a transaction show that it is a locked transfer then the interconnect must ensure that only that master is allowed access to the slave region until an unlocked transfer from the same master completes signalling the release of the locked trasnfer. The arbiter within the interconenct is used to enforce this restriction.

Burst operation

- In AXI channel, the number of data transfers in a single burst are called as beats.

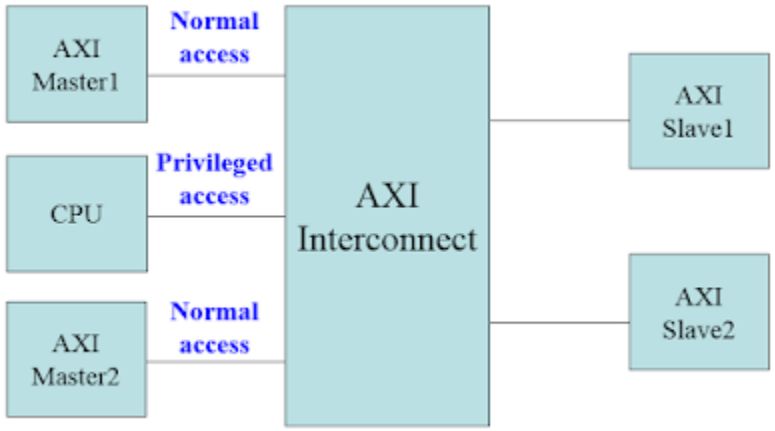


Burst Transfer Type



Normal/Privileged access

- Normal/Privileged is used by some masters to indicate their processing mode. A privileged processing mode typically has a greater lever of access within a system.



Different kinds of burst type

- **Fixed burst:** In a fixed burst, the address remains the same for every transfer in the burst. This burst type is for repeated accesses to the same location such as when loading or emptying a peripheral FIFO.

  Incrementing burst: In an incrementing burst, the address for each transfer in the burst is an increment of the previous transfer address. The increment value depends on the size of the transfer. For example, the address for each transfer in a burst with a size of four bytes is the previous address plus four.

  Wrapping burst: A weapping burst is similar to an incrementing burst, in that the address for each transfer in the burst is an increment of the previous transfer address. However, in a wrapping burst the address wraps around to a lower address when a wrap boundary is reached. The wrap boundary is the size of each transfer in the burst multiplied by the total number of transfers in the burst. There are two restrictions for wrapping bursts: the start address must be aligned to the size of the transfer and the length of the burst must be 2, 4, 8, or 16.

AXI Benefits

- Faster testbench development and more complete verification of AMBA AXI 3.0/4.0 designs
- Easy to use command interface simplifies testbench control and configuration of master and slave
- Simplifies results analysis
- Runs in every major simulation environment

Drawbacks of AXI

- The AMBA AXI4 has limitations with respect to the burst data and beats of information to be transferred
- The burst must not cross the 4k boundary. Bursts longer than 16 beats are only supported for the INCR burst type
- Both WRAP and FIXED burst types remain constrained to a maximum burst length of 16 beats. These are the drawbacks of AMBA AXI system which need to be overcome

AXI features

- AMBA AXI 3.0/4.0 Verification IP provides a smart way to verify the AMBA AXI 3.0/4.0 component of a SOC or a ASIC
- AMBA AXI 3.0/4.0 VIP is supported natively in SystemVerilog, VMM, RVM, AVM, OVM, UVM, Verilog, SystemC, VERA, Specman E and non-standard verification environment

Summary of AXI

- Productivity—By standardizing on the AXI interface, developers need to learn only single protocol for IP
- Flexibility
- AXI4 memory mapped interfaces and allows burst of up to 256 data transfer cycles with just a single address phase
- AXI4-Stream removes the requirement for an address phase altogether and allows
- unlimited data burst size
- Availability—By moving to an industry-standard, access to a worldwide community of ARM Partners.
- Many IP providers support the AXI protocol
- A robust collection of third-party AXI tool vendors is available that provide a variety of verification, system development, and performance characterization tools

Comparison betweenAMBA AHB and AMBA AXI Bus System Modeling

AMBA AHB	AMBA AXI
single-channel, shared bus.	Multi-channel, read/write optimized bus.
A 128 bit bus running at 400 MHz	A 64 bit bus running at 200 MHz
The AHB bus speed was assumed to be double the AXI Bus, and two times the width.	The primary throughput channels- R/W data channels, while the address, response channels are to improve pipelining of multiple requests.

Simulation study between AMBA AHB and AMBA AXI Results

Bus Type	*Latency*	*Throughput*	*Power*
AHB Bus	Excellent	Good	Excellent
AXI Bus	**Good**	**Excellent**	**Good**

Difference between AXI3 and AXI4

- AXI3 Vs AXI4 The difference between AXI3 and AXI4

  1. AXI3 supports burst lengths up to 16 beats only. While AXI4 supports burst lengths of up to 256 beats.
  2. AXI3 supports write interleaving. AXI4 does NOT support write interleaving
  3. AXI3 supports locked transfers, AXI4 does NOT support locked transfers
  4. AXI4 supports QoS, AXI3 does NOT suppor QoS.

  I have seen many IP providers e.g. Synopsys supporting burst lengths up to 256 beats in AXI3
  I have also seen many IP providers e.g. Synopsys NOT supporting write interleaving in AXI3.

  Looks like the industry norm is to use AXI3 with burst lenghts up to 256 beats without support for write Interleaving.

Thank You ☺