

What is AMBA :- Arm Advanced Microcontroller Bus Architecture (AMBA).

on-chip interconnect spec. which defines the connection and management of functional blocks in System on chip (soc).

AMBA used in application processors, used in IoT subsystems, smartphones and networking soc. there are no. of benefits using AMBA -

IP reuse, flexibility, compatibility, support.

- IP reuse requires a common standard while supporting a wide variety of SoC's with different power and area requirements.

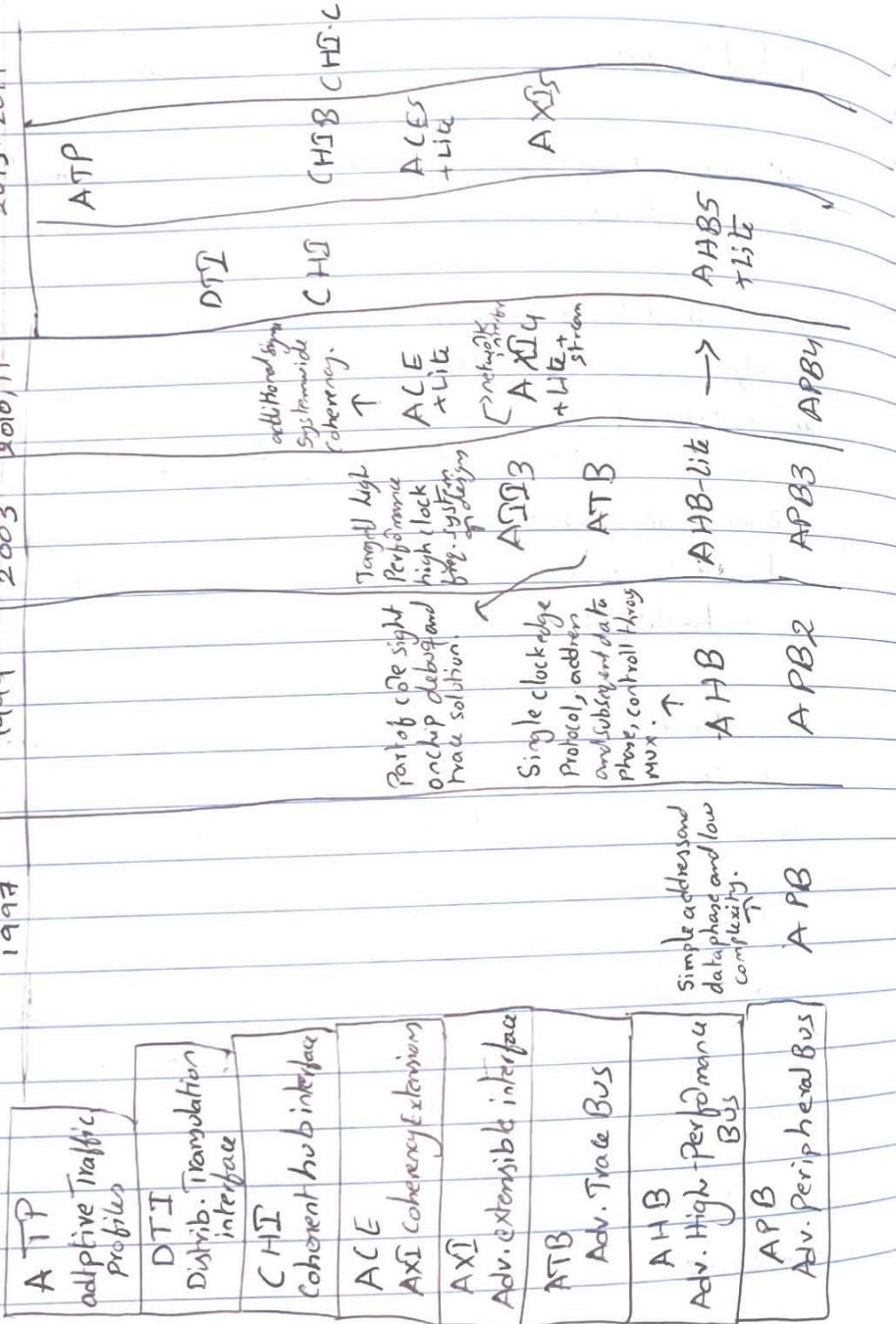
- Compatibility b/w different design teams and vendors.

Bus interface Performance :- Bandwidth and Latency.

Bandwidth is a ratio of data across interface in a Sync. system, the max bandwidth is limited by the product of the clock speed and width of data bus.

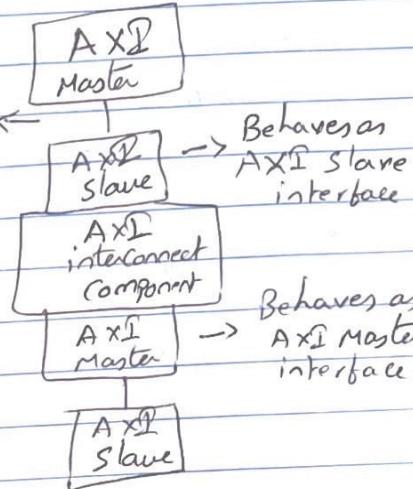
Latency is the delay b/w the initiation and completion of a transaction, although in a burst based system, the latency figure can often refer to the completion of the first transfer rather than the burst, depending on the importance to the system. The efficiency of your interface will depend on the extent to which it achieves the maximum bandwidth with zero latency.

Evolution of the ARR, AMBA Specifications:-



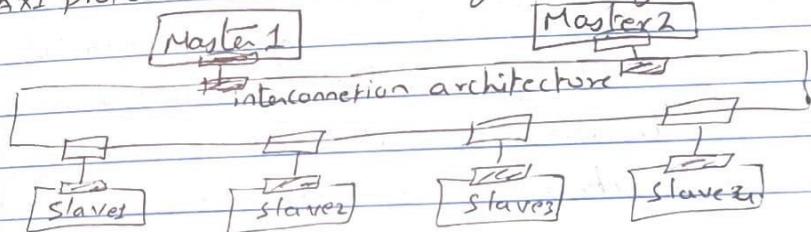
AXI is an interface Specification:- defines the interface of IP blocks, not the interconnect.

All Ax₁ connections
are blu masters
interfaces and
slave interfaces.

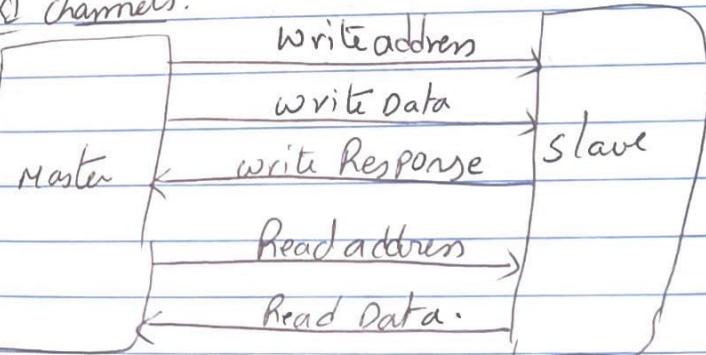


- AXI protocol defines the Signals and timing of the point-to-point connects b/w masters.

AXI protocol in multi master system design.



AXI channels:

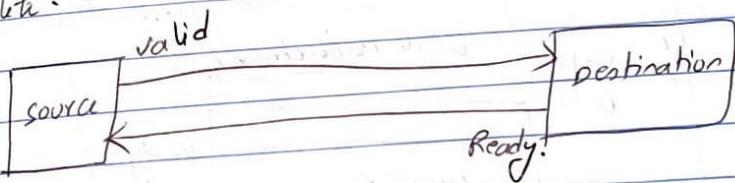


Main features of AXI

- Designed to improve bandwidth and latency of data transfers and transactions.
- Separate write/Read channels improve bandwidth.
- Multiple outstanding address - master can issue transactions without waiting for earlier transactions to complete. improves system performance, because it enables parallel processing of transactions.
- No strict timing relationship b/w the address and data phases. means can issue write address and no restriction on write data can be any time in that location address corresponding data.
- Support for unaligned data transfers.
- Out of order transaction completion :- includes transaction identifiers and there is no restriction on the ordering of transactions with different ID values. single port
- Burst transactions based on start address issue only:- starting address for the first transfer, and implies that for every the following transfer the address is incremented by the slave from the previous transfer address.

Channel handshake :-

Valid goes from the source to the destination and Ready goes in the inverse direction. whether the source or destination is a master or slave depends on which channel is being used. The source indicates when valid info. is available. This signal must remain asserted high until the information is accepted by the destination. Known as sticky signal. destination indicates when it can accept info using the Ready signal. Again Read goes from channel destination to channel source. it is not Async but does require a rising clock for the handshake to complete.

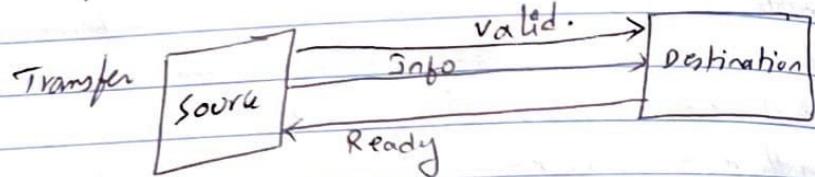


Source uses a VALID signal to indicate info is available:

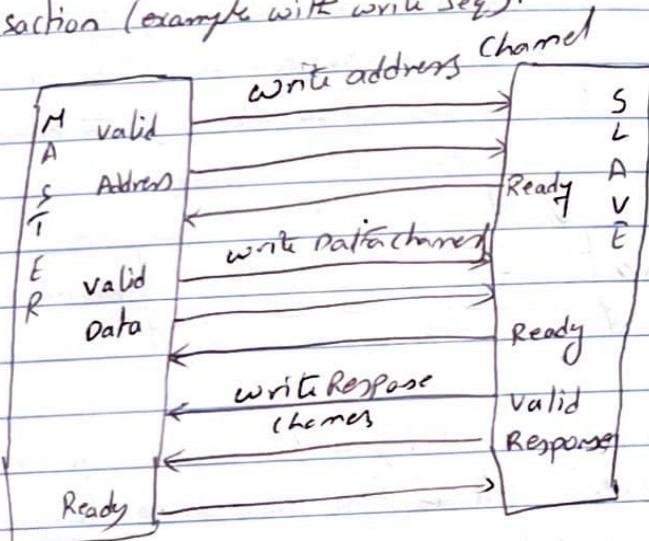
- Once valid is high there is no change until Ready is asserted.
- Destination uses the Ready signal to indicate ability to accept more information.

Axi terminology:

In interconnect fabric you need to know the capabilities of the masters and slaves.



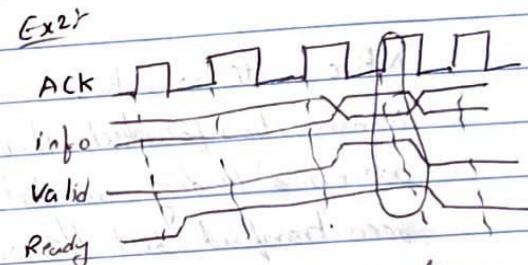
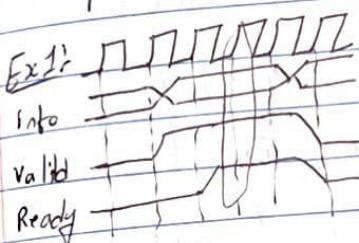
Transaction (example with write seg.).



channel transfer:

- ✓ valid / x ready → A source cannot wait for READY to be asserted before asserting VALID.
- A destination can wait for VALID to be asserted before asserting READY.

Ex2:



* write transaction - single data item.
ACLK, AWADDR, AwVALID, AWREADY.

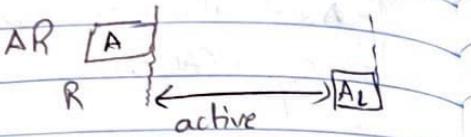
WData, WVALID, WREADY,
WLAST

BRESP, BVALID, BREADY

In multidata transfer
if there is an error in bus
the slave must wait till the complete burst done by master.

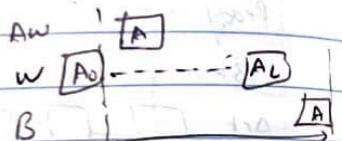
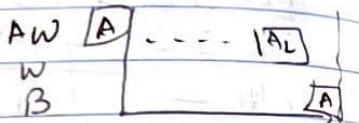
Active transactions:-

active read transaction → A transaction for which the address has been transferred, but the last read data has not yet been transferred.



Active write transaction:-

A transaction for which the write address or leading write data has been transferred but the write response has not yet been transferred.



DataSize, length and burst type:-

AxLEN[3:0] → From 1-16 data transfers in each burst (AXI3).

AxLEN[3:0] → From 1-256 data transfers in each burst (AXI4).

AxSSIZE[2:0] → 1, 2, 4, 8, 16, 32, 64 or 128 bytes per transfer.

AxBURST[1:0] → 00, 01, 10, 11.

00 → Fixed

Same address - ideal for FIFO's - Length from 1 to 16 transfers - fixed byte lanes only defined by start address and size.

01-SNCR

Block transfer - slave increments address - length from 1 to 16 transfers (AXI3). Length from 1 to 256 transfers (AXI4). Unaligned transfers supported.

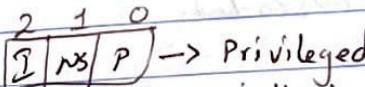
10-WRAP

Suitable for cache line-critical word first - length restricted to 2, 4, 8 or 16. Transfers must be aligned.

11- Reserved.

Protection level Support:-

AxPROT[2:0] defines 3 levels of access protection.



→ Privileged

indicates the processing mode of the master to protection units.

→ Non-Secure

Determines whether the access is secure or non-secure.

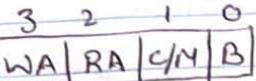
→ Instruction

This can indicate if an access is operating on instructions or data.

This may not be useful sometimes to understand we change 'NS' as '0' as a data.

Cache Support :-

Ax CACHE [3:0] -



Read allocate Cacheable (A \times 3)
if there is a cache miss Modifiable (A \times 4)
The burst and transfer
read then the characteristics can
transfer should change b/w source and destination
be allocated.

Write allocate (WA) :- if there is a cache miss on write
then the transfer should be allocated.

Cache can be placed anywhere in the system for example:

- Next to the processor (L1 cache).
- Within the interconnect.
- Next to the memory controller (L3 cache).

Response Signaling:-

RRESP[1:0] - Response to each transfer of READ burst.

BRESP[1:0] - Single response at completion of WRITE burst.

00 - OKAY

Normal access failure (Exclusive access failure).

01 - EXOKAY

Exclusive OKAY access success.

10 - SLVERR

Sub-4K register decode, slave specific error or security violation.

11 - DECERR

Address decode error (>4K) or security violation.

Write data strobes:-

WSTRB[Ex:0] - one bit per byte of WDATA bus to indicate valid data.

for 64 bit WDATA bus.

This signal is used by a master to tell a slave which bits of the data bus are required.

The write channel has strobe bits, one per byte on the data bus.

These bits make up the WSTRB signal. A master must ensure that the write strobes are HIGH only for byte lanes that contain valid data.

For 64bit wDATA bus

63	56 55	48 47	40 39	32 31	24 23	16 15	8 7 0
7	6	5	4	3	2	1	0

~~EF GTS 34 F3 21 10~~ WSTRB = 0xFC

~~7 6 EF F4 F3 21 10~~ WSTRB = 0x3C

~~EF GTS 34 33 21 10~~ WSTRB = 0x31

~~EF GTS 34 33 21 10~~ WSTRB = 0xE8

Byte lane strobes offer efficient movement of sparse data arrays. write transactions can be 'early terminated' by setting the remaining transfer byte lane strobes to '0'. although the remaining transfer still need to be completed.

Atomic access:-

AxLOCK indicates when Atomic Accesses are being performed.
AxLOCK[1:0] - AXI3

L > 0 Normal

L > 1 Exclusive

The exclusive access mechanism enables the implementation of semaphore type operations without requiring the bus to remain locked to a particular master for the duration of the operation.

L > 10 Locked

Locked accesses are similar to the mechanism supported with AHB - a locked transfer locks the channel which remains locked until an unlocked transfer is generated.

L > 11 Reserved

AxLOCK - AXI4

L > 0 Normal

L > 1 Exclusive

The exclusive access mechanism enables the implementation of semaphore type operations without requiring the bus to remain locked to a particular master for the duration of the operation.

Quality of Service:-

AxQOS - Defines the priority on a per transaction basis.

- AWQOS, ARQOS - 4 bits wide.

Values can be used by:

- Arbiters within an interconnect to resolve contentions
- slaves to reorder and prioritize transactions.

Encoding of 0x0 is
lowest priority

Encoding of 0xF is ↑
highest priority

Region Signaling and user Signals:

Region Signaling → AxRegion - Defines region identifiers.

- AWREGION, ARREGION 4 bits wide.
- Can be used to uniquely identify up to sixteen different regions.
- Single physical interface on a slave can provide multiple logical interfaces.

User Signals → AxUSER - The AXI protocol does not define functions for these signals.

- Can be useful to transfer additional control information.
- It is not required that user signals are supported on all channels.

Region Signaling and user Signals:-

AxREGION - Defines region identifiers.

- AWREGION, ARREGION 4 bits wide.

- Can be used to uniquely identify up to sixteen different regions.
- Single physical interface on a slave can provide multiple logical interfaces.

User Signals :-

AxUSER - The AXI protocol does not define functions for these signals.

- Can be useful to transfer additional control information.
- It is not required that user signals are supported on all channels.

AXI channel dependencies:-

WVALID can be asserted before AWVALID → write data can be seen before the address.

WLAST transfer must complete before BWVALID is asserted → All of the write data must have been sent before a write response can be seen.

→ AXI3 - the address does not have to be seen before a write response is sent.

→ AXI4 - all of the data and the address must have been transferred before a write response can be seen.

RVALID cannot be asserted until ARADDR has been transferred → no read data can be transferred without seeing the address first.

Read channel signals:-

Read Address | AR

ARVALID

ARREADY

ARADDR[31:0]

ARSIZE[2:0]

ARBURST[1:0]

ARCACHE[3:0]

ARPROT[2:0]

ARID[x:0] $\xrightarrow{x \in 3}$

• ARLEN[3:0]/ARLEN[7:0] $\xrightarrow{x \in 4}$ → AXI4 longer burst

• ARLOCK[1:0]/ARLOCK $\xrightarrow{x \in 2}$ → reduced to accommodate exclusive

• ARQOS[3:0] \rightarrow quality of transfers.

• ARREGION[3:0] service and slave region.

• ARUSER[x:0]

Read data | R

RVALID

RREADY

RLAST

RDATA[x:0]

RRRESP[1:0]

RID[x:0]

• RUSER[x:0]

Write Channel Signals:-

Write address | AW

AWVALID

AWREADY

AWADDR[31:0]

AWSIZE[2:0]

AWBURST[1:0]

AWCACHE[3:0]

AWPROT[2:0]

AWID[x:0]

AWLEN[3:0] $\xrightarrow{A_x \in 3}$

• AWLEN[3:0]/AWLEN[7:0] → generate longer burst

• AWLOCK[1:0]/AWLOCK → exclusive transfer.

• AWQOS[3:0] → Quality of Service for priority of transactions.

• AWREGION[3:0] → slave region.

AWUSER[x:0]

Write data | W

WVALID

WREADY

WLAST

WDATA[x:0]

WSTRB[x:0]

WUSER[x:0]

WID[x:0]

WLEN[3:0] $\xrightarrow{A_x \in 3}$

• WLEN[3:0]/WLEN[7:0] → generate longer burst

• WLOCK[1:0]/WLOCK → exclusive transfer.

• WQOS[3:0] → Quality of Service for priority of transactions.

• WREGION[3:0] → slave region.

WUSER[x:0]

Write response | B

BVALID

BREADY

BRSP[1:0]

BID[x:0]

BUSER[x:0]

Locked access overview - AXI3:-

Locked transaction → Master ensures that there are no other outstanding transactions waiting to complete before starting locked transactions.

Initializes with locked transfer and interconnect ensures only that master is allowed access to the slave region during the locked transaction. → ARM supports locked access b68 completes with an unlocked transfer. only support legacy devices.

Locked access operation - AXI3:- we have two masters Mo and M₁. Before Master can start a locked sequence of transactions it must ensure it has no other when Mo uses a lock signal for a transaction to indicate its locked transaction.

Then interconnect must use an arbiter to ensure that only Mo can access the targeted slave region by blocking M₁ access until unblock from Mo completes.

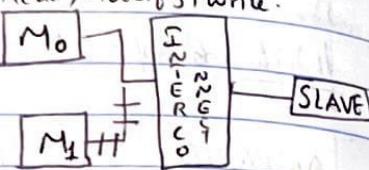
Master should wait till Mo completes transaction which takes atleast 2, if unlock is not given after write it removes and allows other Masters.

Exclusive access - AXI4.

- Exclusive access performs atomic operations more effectively than locked access by making better use of interconnect bandwidth.

Semaphore style Read then WRITE pair of transactions -

- Transactions can be made than one data transfer
- Transactions must have identical Address channel attributes
- Address must be aligned according to total no. of bytes in transfer.



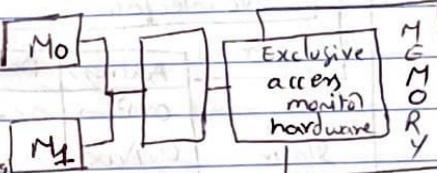
The bytes that are to be transferred are limited to 1, 2, 4, 8, 16, 32, 64 or 128 bytes.

Requires slave hardware support → Exclusive Access Monitor.

- Allows other masters access to the peripheral → use exclusive access for all new designs.
- Only use LOCK for legacy designs.

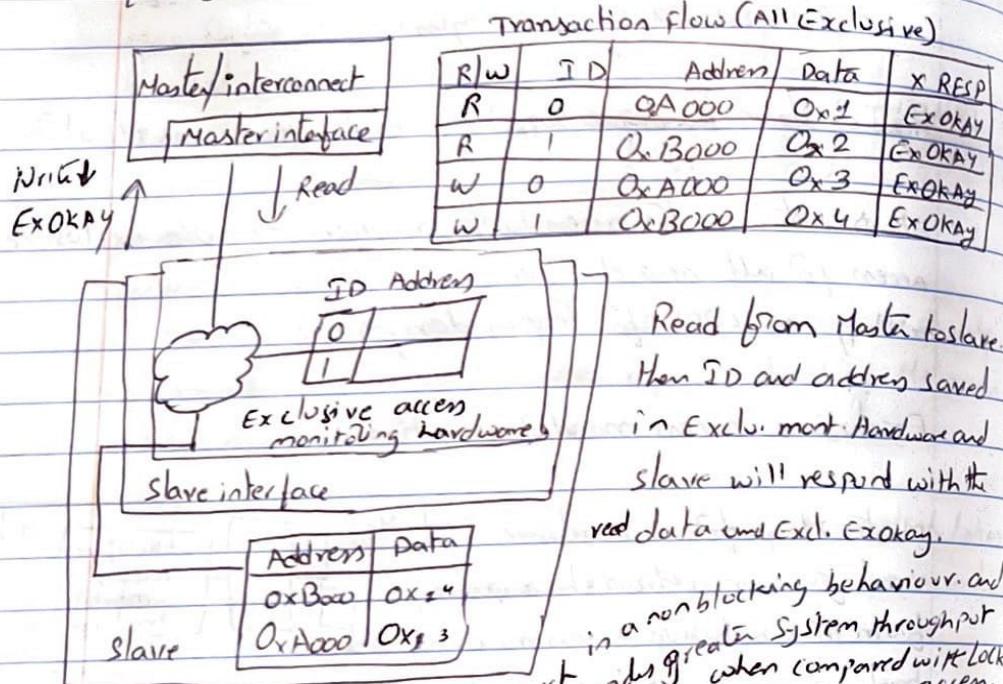
Exclusive access monitor operation:-

Master Mo performs an exclusive read from an address the response from the Exclusive Access monitoring



hardware can be EXOKAY, which means that the value is read and the ID of the transaction is stored in Exclusive access monitor hardware. Or the response can be OKAY, which means that the value is read, but there is no support for Exclusive access, and the master should treat this response as an error for the exclusive operation. At some point later, if EXOKAY was received during excl. read, Mo attempts to complete the ex. operation by performing an excl. write to the same address, using same transaction ID used for excl. read.

Exclusive transaction pairs - Par/Par -



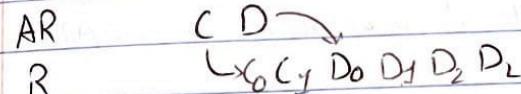
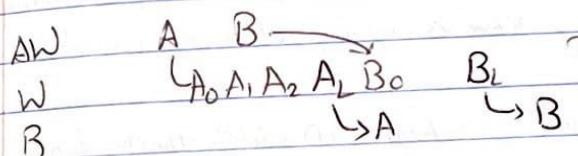
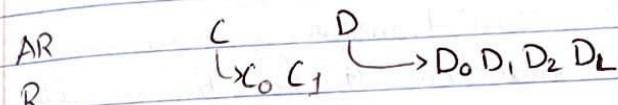
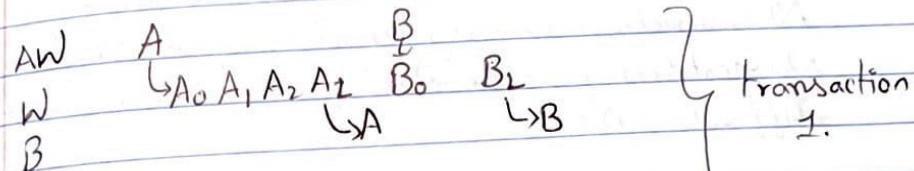
Par/fail:- → This OK → means write operation

has been carried on this memory

R/W	ID	Address	Data	XRESP
R	0	0xA000	0x1	EXOK
R	1	0xA000	0x2	EXOK
W	0	0xA000	0x3	EXOK
W	1	0xA000	0x4	OK

Master informed via OKAY
(Exclusive failure). response
and memory not updated.

Simple transaction:-



Transfer IDs:-

AWID
WID
BID
ARID
RID

These give the possibility to complete transaction
out of order meaning the transactions to faster memory
regions can complete without waiting for earlier transaction.
Improves throughput, system efficiency, system performance
because reduces transaction latencies.

AXI supports out-of-order transactions by enabling each interface to act as multiple ordered interfaces.

All transactions with a given ID must be ordered.

No restriction on the ordering of transactions with different ID's.

General Rules:- → All transfers have an ID.

- All transfers in a transaction have the same ID.

- Masters can support multiple IDs for multiple threads.

- Slaves generally need a configurable ID width.

AXI ID parameters → Write ID width, the no. of bits used for the AWID, WID and BID buses.

- Read ID width the no. of bits used for the ARID and RID buses.

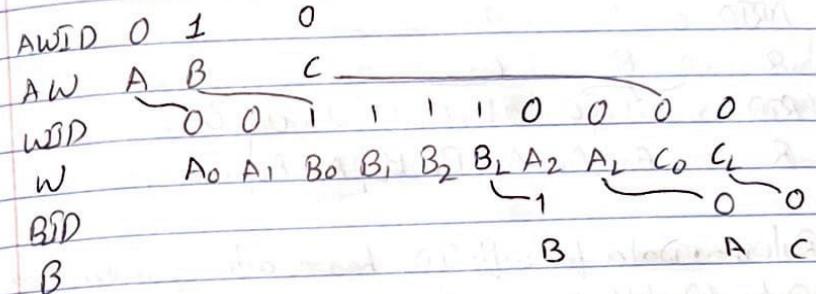
Write ordering rules:-

3 rules → i. write data on the wDATA channel must follow the same order as the address transfer on the AW channel.

ii) Transactions with diff. ID's can complete in any order, so here we can see that B completes before A.

→ - write data interleaving is not supported

iii) Master can have multiple outstanding transactions, but they must be performed in order and complete in order



Rules:- Data must follow the same order as address transfer.

• Data for diff. transaction ID's can be interleaved.

• Data with the same ID must follow in the order as issued.

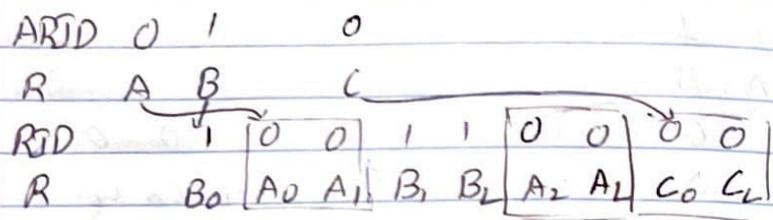
• Important issue for interconnect when routing different slave data to master ID.

Read ordering rules:- 3 rules

i) The read data for diff. IDs on the RDATA channel has no ordering restrictions. This means that it can be sent by the slave in any order.

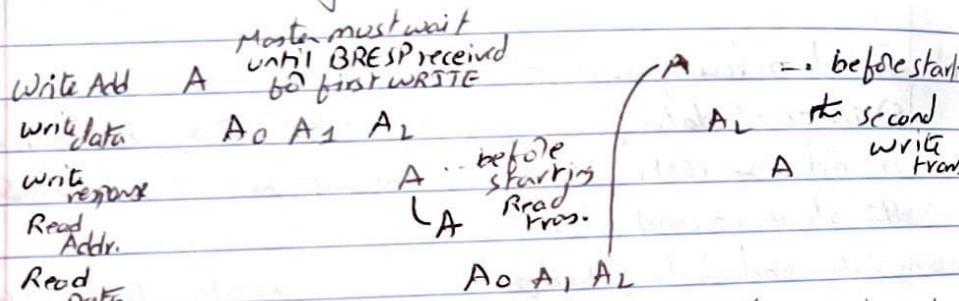
ii) The read data for diff. ID's on the RDATA channel can be interleaved, which is same for write transfers. And the RID values differs which transaction the data relates to.

iii) The read data on the RDATA channel, for transactions with the same ID, must be returned in the order they were requested.



- Rules:- Data for diff. ID's has no ordering restrictions.
- Data for diff. trans. ID's can be interleaved.
- Data with the same ID must follow in the order as issued.
- Important issue for the interconnect when routing diff slave data to master ID.

Read/write ordering:-



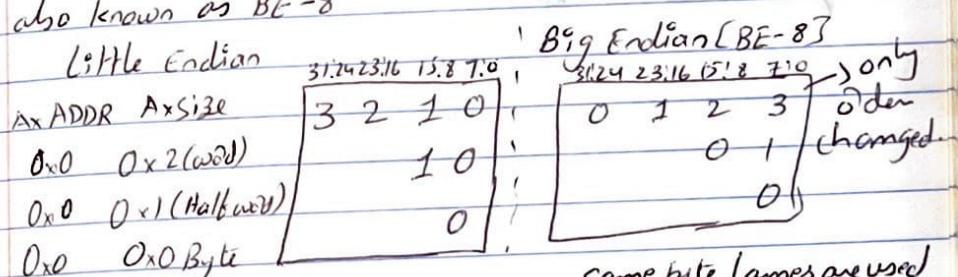
Master must ensure READ/WRITE transfer order req. is met.
Ex:- master reqs w-Rw seq completes in order.

Unaligned transfer start address :- ?

An unaligned transfer is where AXI ADDR Values do not have to be aligned to the width of the trans. AXISIZE.

Endianness Support:-

Access to mixed-endian data structures in the same memory space uses a Byte Invariance Endianness scheme also known as BE-8



same byte lanes are used in BE-8

Write interface attributes:-

→ Write issuing capability : The max no. of active write trans the master interface can generate.

→ Write interleave capability (AXI3 only) : The no. of active write trans. for which the master interface is capable of transmitting data.

→ Write acceptance capability (AXI3 only) : The max no. of active write trans. the slave interface can accept.

→ Write interleave depth : The no. of active write trans for which the slave interface can receive data.

Read interface attributes :-

- Read issuing capability - The max no. of active read trans. that a master interface can generate.
- Read acceptance capability - The max no. of active read trans. that a slave interface can accept.
- Read data reordering depth - The no. of active read trans. for which a slave interface can transmit data - this is counted from the earliest transaction.