AXI SLAVE

Design Document

Contents

1. Introduction

Advanced eXtensible Interface (AXI) is a part of ARM Advanced Microcontroller Bus Architecture specifications. It is a parallel high-performance, synchronous, high-frequency, multi-initiator, multi-target communication interface, mainly designed for on-chip communication. The AXI-4 protocol is the fourth generation of the AMBA interface specifications. It is an update to AXI3 which is designed to enhance the performance. It includes the following enhancements: Support for burst lengths up to 256 beats.

The AXI protocol is burst-based and defines the following independent transaction channels:

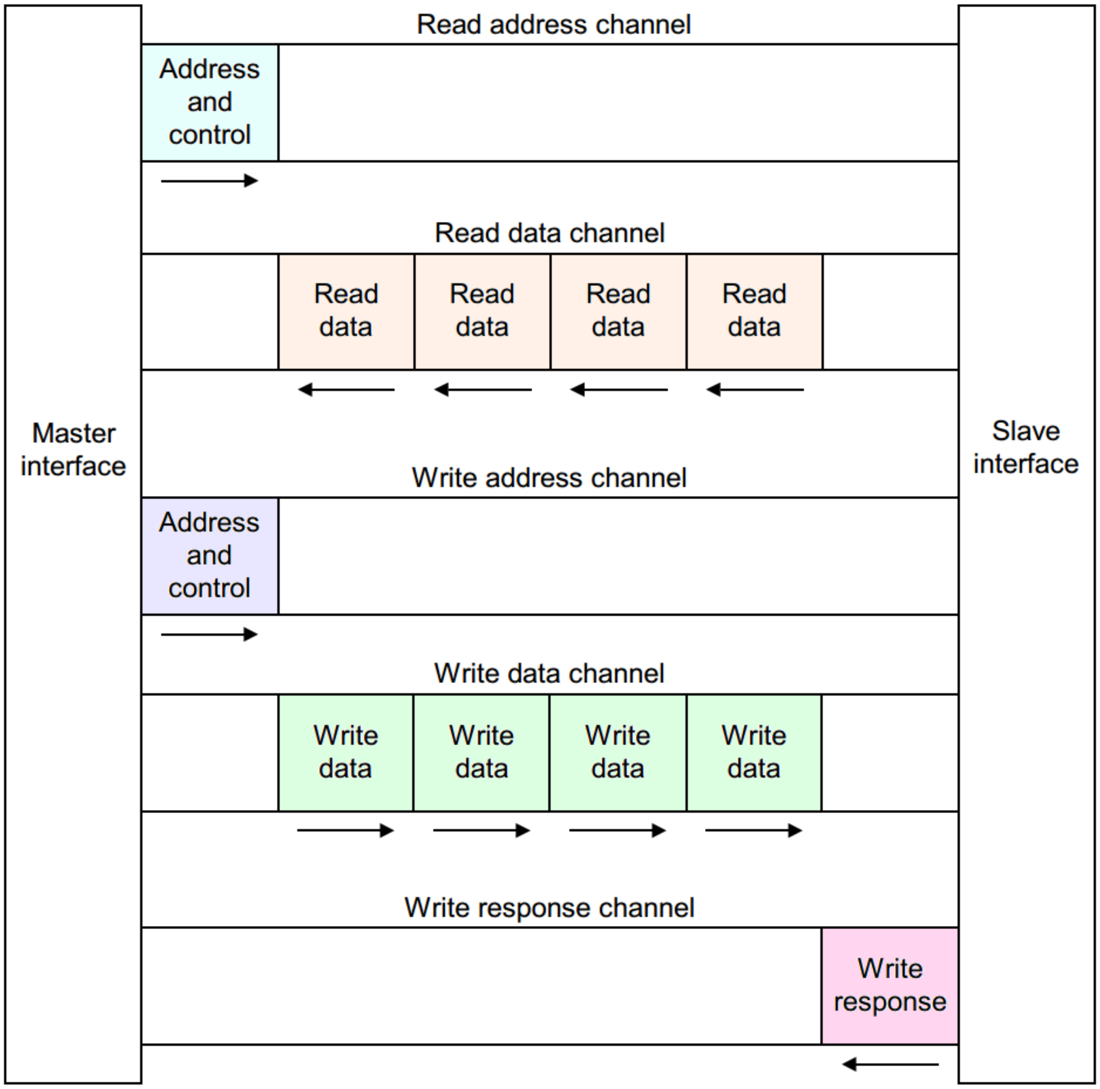
- read address
- read data
- write address
- write data
- Write response.

An address channel carries control information that describes the nature of the data to be transferred. The data is transferred between master and slave using either:

- A write data channel to transfer data from the master to the slave. In a write transaction, the slave uses the write response channel to signal the completion of the transfer to the master.
- A read data channel to transfer data from the slave to the master.

The Applications of AXI protocol:

- For on chip communications between CPU, memory and peripheral IPs.
- Fixed burst is used for data transfer to fixed in FIFO's and in UART transmitters and receivers.
- .WRAP Burst is used for accessing Instructions in Cache memory.
- Write Strobe is used for Atomic access in memories.



Channel Definition

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.

Read and write address channels

Read and write transactions each have their own address channel. The appropriate address channel carries all of the required address and control information for a transaction.

Read data channel

The read data channel carries both the read data and the read response information from the slave to the master, and includes:

- The data bus, that can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits wide
- A read response signal indicating the completion status of the read transaction.

Write data channel

The write data channel carries the write data from the master to the slave and includes:

- The data bus, that can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits wide
- A byte lane strobe signal for every eight data bits, indicating which bytes of the data are valid.

Write data channel information is always treated as buffered, so that the master can perform write transactions without slave acknowledgement of previous write transactions.

Write response channel

A slave uses the write response channel to respond to write transactions. All write transactions require completion signaling on the write response channel.

2. Signal Description

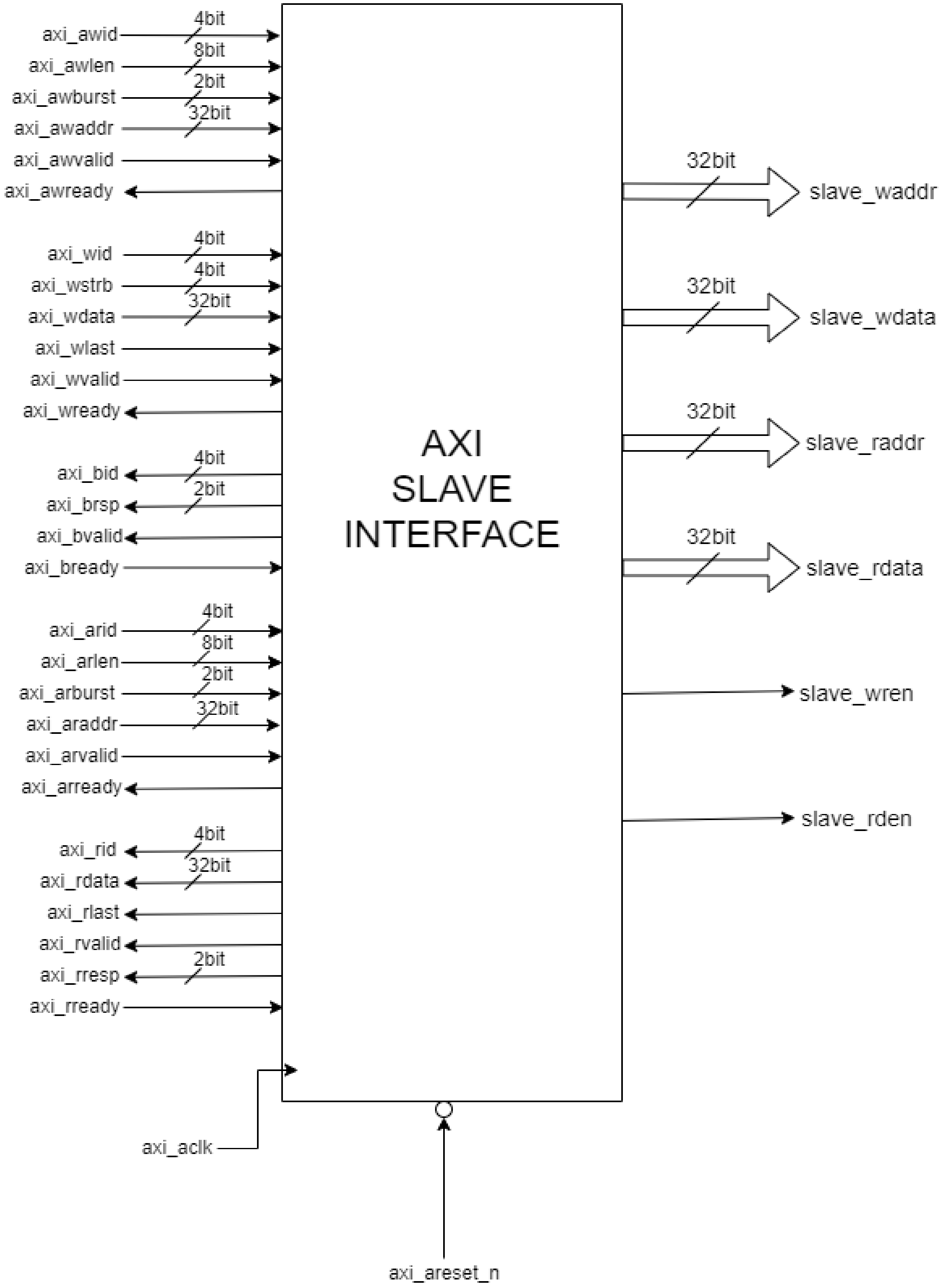
Pin Name	Default value	Direction	Width	Description
GLOBAL SIGNALS				
axi_areset_n	1’d1	Input	1-bit	Active low reset. Slave is in operating mode when the value of ARESETn is high. Pulling down the value to logic low will make the slave block to reset making all the outputs zero.
axi_aclk	1’d0	Input	1-bit	Global clock signal for AXI Slave. Default clock signal for the slave block
WRITE ADDRESS CHANNEL				
axi_awaddr	32’d0	Input	32-bit	Write Address. Write address bus carry the address required for write transactions
axi_awvalid	1’d0	Input	1-bit	Write Address Valid. Handshaking input from master system indicating that valid address is present on write address bus.
axi_awready	1’d0	Output	1-bit	Write Address Ready. Handshaking output produced from AXI slave indicating that slave is ready to accept the address present on adress bus.
axi_awid	4’d0	Input	4-bit	Write Address ID. Master should send the write address with valid slave ID on this signal to capture the write address at slave.
axi_awlen	8’d0	Input	8-bit	Write Burst Length. Master will specify the number of transfers(beats) in the write burst to the slave with this signal.
axi_awburst	2’d0	Input	2-bit	Write Burst Type. Master will specify type of burst will be used for transfer of write data with this signal.

WRITE DATA CHANNEL				
axi_wdata	32’0	Input	32-bit	Write Data. Write data bus carry the 32 bit data sent from master.
axi_wvalid	1’d0	Input	1-bit	Write Data Valid. Handshaking input from master system indicating that valid data is present on write data bus.
axi_wready	1’d0	Output	1-bit	Write Data Ready. Handshaking output produced from AXI slave indicating that slave is ready to accept the data present on adress bus.
axi_wid	4’d0	Input	4-bit	Write Data ID. Master should send the data to be written, with valid slave ID on this signal to capture the data at slave.
axi_wlast	1’d0	Input	1-bit	Write Last. Master produce write last signal along with the last beat of burst transaction to indicate end of the burst.
axi_wstrb	4’d0	Input	4-bit	Write Strobe. Master will specify which byte lines of data bus contain valid data from this signal.
WRITE RESPONSE CHANNEL				
axi_bresp	2’b00	output	2-bits	Write Response or B Response. B response bus carry responses for every transactions indicating status of transaction.
axi_bvalid	1’d0	Output	1-bit	Bresponse Valid. Handshaking output produced from AXI slave indicating that valid B response is present on the response bus.
axi_bready	1’d0	Input	1-bit	Bresponse Ready. Handshaking input from the master system indicating that master is ready to accept the response present on response bus.
axi_bid	4’d0	Output	4-bit	Write Response ID. Slave send the write response along with its valid ID on this signal.

READ ADDRESS CHANNEL				
axi_araddr	32’d0	Input	32-bits	Read Address. Read address bus carry the address required from which data has to be read.
axi_arvalid	1’d0	Input	1-bit	Read Address Valid. Handshaking input from master system indicating that valid address is present on read address bus.
axi_arready	1’d0	Output	1-bit	Read Address Ready. Handshaking output produced from AXI slave indicating that slave is ready to accept the read address present on adress bus.
axi_arid	4’d0	Input	4-bit	Read Address ID. Master should send the read address with valid slave ID on this signal to capture the read address at slave.
axi_arlen	8’d0	Input	8-bit	Read Burst Length. Master will specify the number of transfers(beats) should be there in the read burst from the slave with this signal.
axi_arburst	2’d0	Input	2-bit	Read Burst Type. Master will specify type of burst should slave use for transfer of read data with this signal.
READ DATA CHANNEL				
axi_rdata	32’d0	Output	32-bit	Read Data. Read data bus carry the 32 bit data sent to the master.
axi_rvalid	1’d0	Output	1-bit	Read Data Valid. Handshaking output produced from AXI slave indicating that valid read data is present on read data bus.
axi_rready	1’d0	Input	1-bit	Read Data Ready. Handshaking input from the master system indicating that master is ready to accept the data present on read data bus.
axi_rid	4’d0	Output	4-bit	Read ID. Slave send the read data along with its valid ID on this signal.

axi_rlast	1’d0	Output	1-bit	Read Last. Slave produce read last signal along with the last beat of burst transaction to indicate end of the burst.
axi_rresp	2’b00	output	2-bit	Read response. This two bit signal indicate the status of read transaction.
SLAVE TO MEMORY OUTPUTS				
slave_wren	1’d0	Output	1-bit	Write Enable. Write enable signal to memory.
slave_waddr	32’d0	Output	32-bit	Write Address. Carries the address of the location to which data has to be written.
slave_wdata	32’d0	Output	32-bit	Write data. It carries the data to be written in memory. The data is written into the memory pointed by the write address.
slave_rden	1’d0	Output	1-bit	Read Enable. Read enable signal to memory.
slave_raddr	32’d0	Output	32-bit	Read Address. Carries the address of the location to which data has to be written.
slave_rdata	32’d0	Input	32-bit	Read data. It carries the data to be read from memory. The data is readout from the memory pointed by the read address.

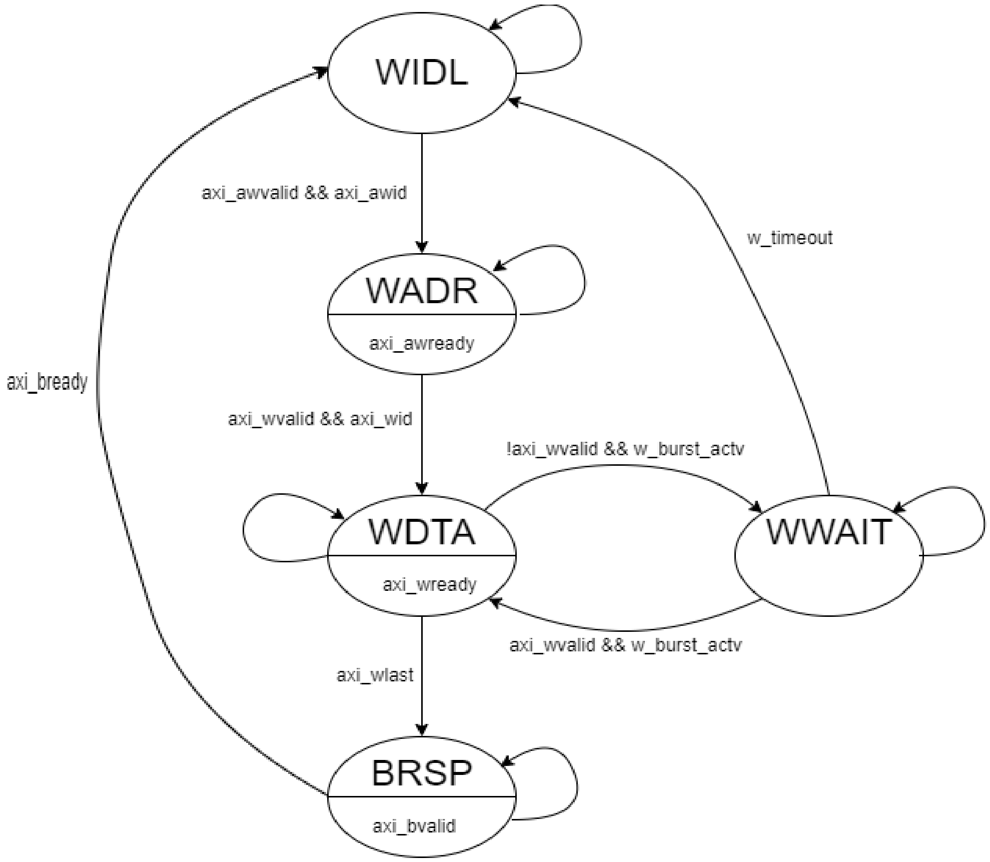
3. AXI Slave Block Diagram



4. AXI Slave FSMs

All the Handshaking signals are generated depending on write FSM and read FSM, write FSM produces handshaking signals for write address channel, write data channel and write response channel and read FSM produces handshaking signals for read address channel and read data channel.

WRITE FSM

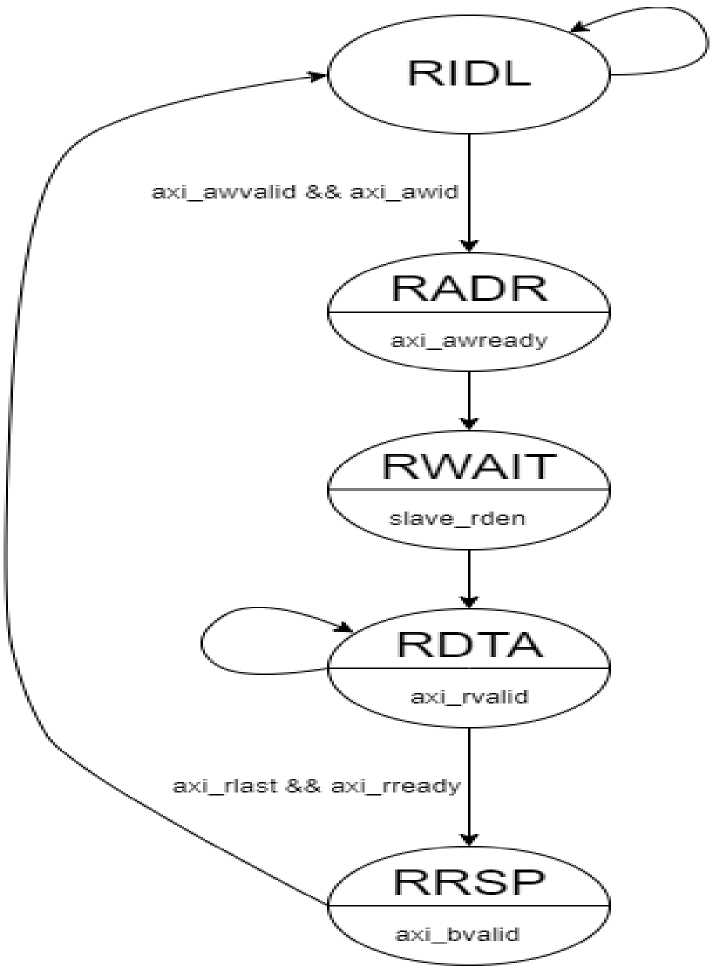


All the handshaking signals from the slave for the write operation to the MASTER system are generated by the Write FSM. Write handshaking signals from MASTER system are inputs to the write FSM and each state of the FSM is responsible for generating respective handshake signal to the MASTER system.

- For axi_awvalid request FSM produces axi_awready grant signal indicating that the slave is ready to accept the address.

- For axi_wvalid request, FSM will produce axi_wready grant signal only if the address is latched before. MASTER may request for write data before write address transfer but slave will will not generate ready signal until the write address is latched.
- Master can assert axi_wvalid continuously and burst full-length data for each clock cycles when there is ready signal or it can send interleaved data with different number of beats.
- For interleaved data transfer the time duration between two wvalid signals should not cross 2000ns. If the wvalid signal crosses the time limit MASTER system should restart the write process by sending write address.
- MASTER system will assert axi_wlast signal along with last beat (single transfer in a burst) of the burst, FSM produces axi_bvalid request to send write response to the MASTER system. axi_bvalid remain high for one clock cycle after MASTER system asserts axi_bready.

READ FSM



All the handshaking signals from the slave for the read operation to the MASTER system are generated by the read FSM. Read handshaking signals from the MASTER system are inputs to the read FSM. Each state of the FSM is responsible for generating respective handshake signal to the MASTER system.

- For axi_arvalid request FSM produces axi_arready grant signal indicating that the slave is ready to accept the address.
- Once the address is latched slave access the data from the latched address and asserts the axi_rvalid to indicate that the data is ready on the bus to be sent to the MASTER system. The MASTER should assert axi_rready signal to accept and latch the read data. axi_rvalid remains high for one clock cycle after the MASTER system asserts axi_rready.
- AXI Slave asserts axi_rlast signal along with the last beat of the burst.
- Read response (axi_rresp[1:0]) is sent when the last data is accepted by the MASTER system.

5. Transaction structure

Write address latching

Write address channel contains a 32-bit write address bus (axi_awaddr[31:0]) for transfer of write addresses to the slave. The write address is sent on the axi_awaddr by the MASTER system is captured and latched by the slave only after axi_awvalid and axi_awready are high for one clock cycle. Master system must keep axi_awvalid asserted for at least one clock cycle after slave asserts axi_awready. The address is transferred at the same clock cycle when both axi_awvalid and axi_awready are high and it is flopped to start_awaddr_reg in the next cycle. The write control signals are also registered on the same clock cycle of write address. The address is further calculated depending on the control signals.

Write data latching

Write data channel contains 32-bit wide write data bus (axi_wdata[31:0]).Write data is sent by the MASTER system on the axi_wdata bus of write channel along with axi_awvalid signal. The data is captured in the clock cycle when both axi_wvalid and axi_wready are high and registered to WR_reg with respect to the write strobes (axi_wstrb[3:0]) in the next clock cycle. Slave is capable of capturing both continuous and interleaved data bursts. Only the byte lanes specified by the axi_wstrb[3:0] captured and flopped to the WR_reg.

Write Response or Bresponse generation

Write response is used to confirm the MASTER system that the write data burst has arrived and captured completely by the slave. Write response channel contains 2-bit Bresponse bus (axi_brsp[1:0]) used to send the response to the MASTER system. The axi_brsp will be at high impedance(2'b00) by default and will have the value 2'b00 when axi_bvalid is high. Bresponse is sent along with the axi_bvalid. MASTER system should capture the Bresponse when both axi_bvalid and axi_bready are high.

Read address latching

Read address is sent by the MASTER system to fetch the read data by the slave. Read address channel consists of 32-bit wide read address bus (axi_raddr[31:0]). Read address is sent by the MASTER system to the slave on axi_raddr. The read address is captured when both axi_arvalid and axi_arready are high for one clock cycle and flopped in the next clock cycle to strt_raddr_reg. The read control signals are also registered at the same clock cycle of axi_raddr. The registered read address is used to calculate the next addresses depending on the control information.

Read data generation

Read data channel consists of 32-bit read data bus (axi_rdata[31:0]). Read data is transferred from slave to MASTER system with respect to the control signals sent by the MASTER system in read address channel. Slave will fetch the data from memory side pointed by the calculated read addresses in slave. Read data is asserted on axi_rdata bus along with the axi_rvalid signal. The MASTER system should assert axi_rready signal to capture the read data when both axi_rvalid and axi_rready are high for one clock cycle. Depending on the burst length slave asserts axi_rlast signal along with the last read data of the burst and produces read response(axi_rrsp[1:0]) in the next clock cycle of last beat capture. Read response bus is of 2-bits with default value as 2'b00. When the last read data is sent to MASTER system axi_rrsp will have the value of 2'b00, indicating all the requested data in the burst have been sent.

Address structure

For both write and read operation the MASTER system will send only the first address of the transaction. AXI Slave will calculate the address of subsequent transfer in the burst depending on the control information driven by the MASTER system. Depending on burst type either the address increments or remains same. For FIXED burst address is same throughout the burst and for INCR burst address increments by one for each transfer (beat) start from first address. The increment depends on the burst length. Increment stops once the number of transfers are equal to the burst length. For WRAP burst the address increments up to the burst length for the last beat it wraps around to the base address.

6. Verification goals

For AXI Slave block Test bench acts as a MASTER system as well as configurations register block. Since AXI Slave is an interface block testbench should cover the ports at both sides of the slave block.

- Slave should be capable of generating handshaking signals for all the five channels.
- Information should be registered at the next clock cycle of the handshaking process.
- Address should be calculated depending on the control information driven by the TB.
- Appropriate address and data should be generated at the memory side with respect to the driven inputs from the MASTER side.

Verification sequence

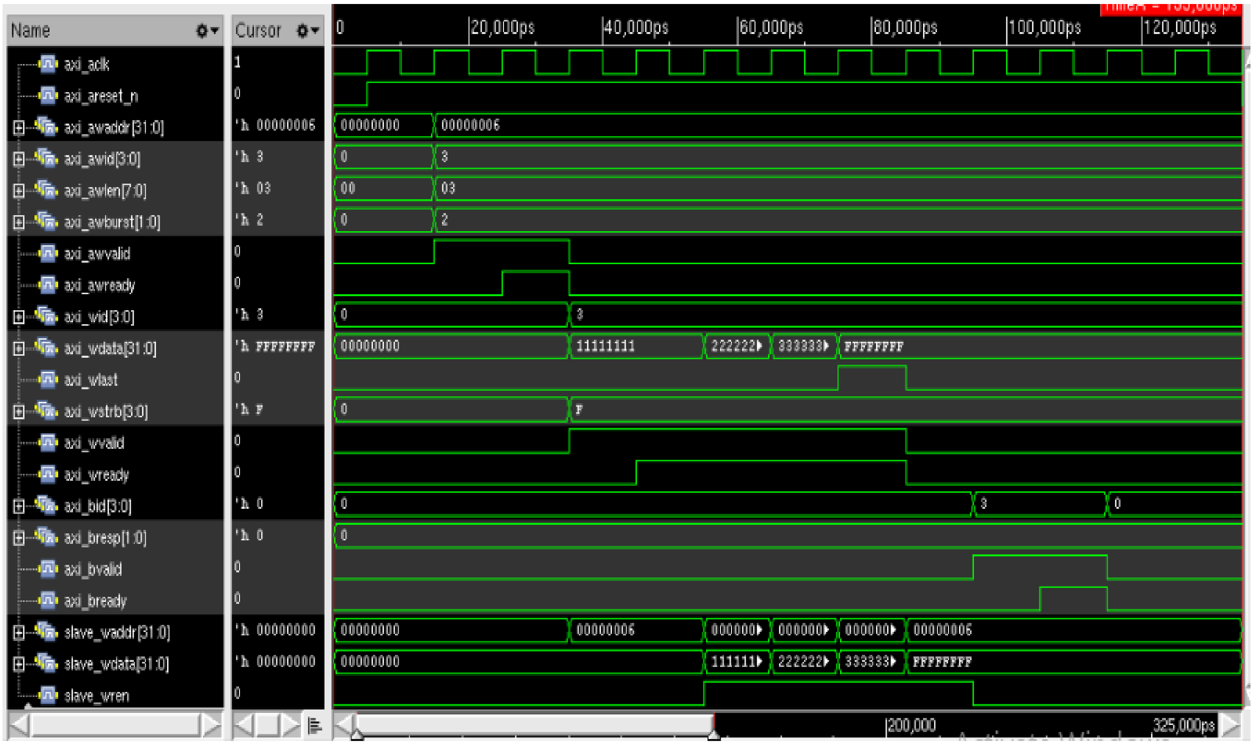
- All the address and data should be driven as per AXI handshake mechanism.
- Only burst mode data is acceptable by the slave.
- MASTER system should begin each burst by driving control information and the address of the first transfer in the burst.
- All VALID and READY signals of AXI Slave block have default value as low.
- Burst should start with start address and control information and end with LAST signal (axi_wlast or axi_rlast).

7. Test Cases

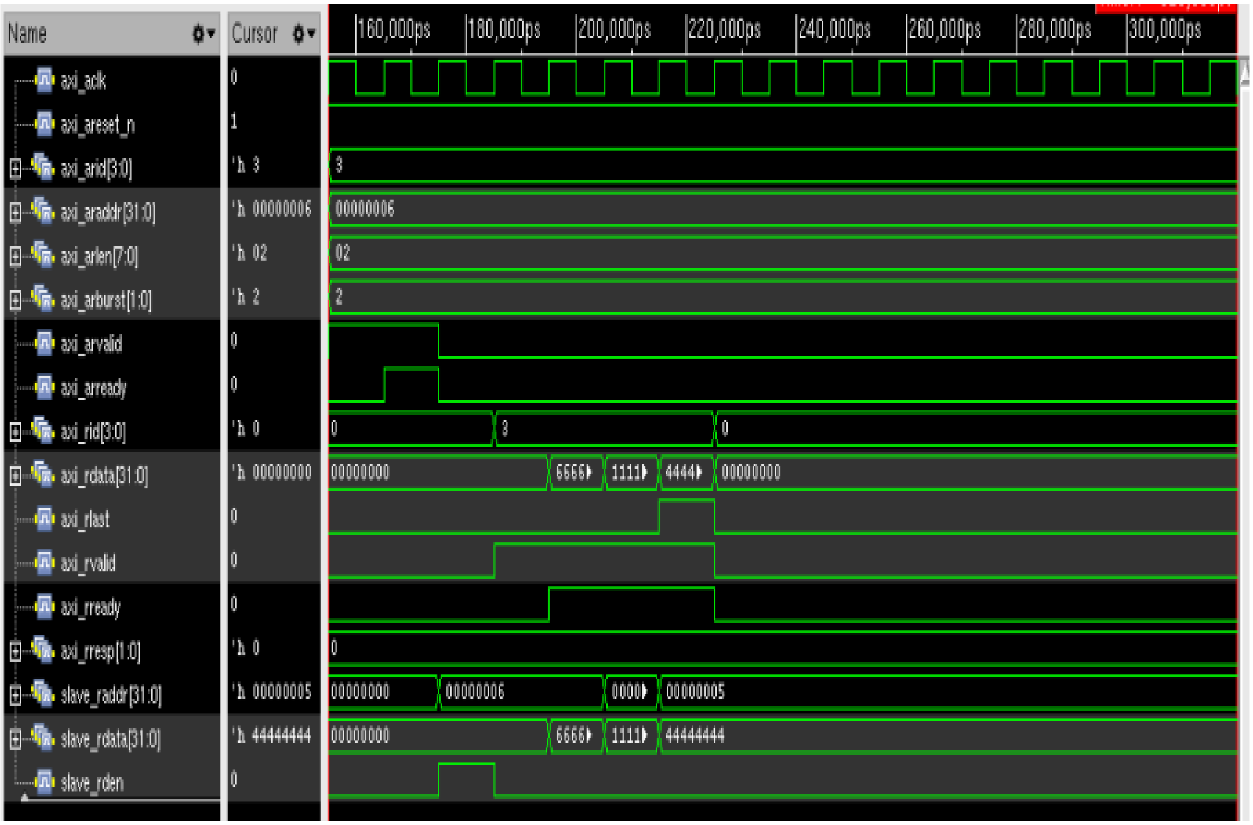
Serial No.	Test Case	Results
1.	Fixed burst for write address calculation	Start address is registered. The address is calculated for fixed burst with the same address for each beat.
2.	Incremental burst for write address calculation	The start address is registered and is incrementented for each clock cycle upto the burst length.
3.	Wrap burst for write address calculation	The address increments properly and the address wraps around to the base address after reaching the burst length.
4.	Axi_awvalid high for three clock cycles	The axi_awready is high after receiving the axi_awvalid and is not deasserted until awvalid is deasserted.
5.	Axi_wdata with axi_wvalid is given before the awaddr	The slave does not start the write operation.
6.	Different values of axi_wstrb	The data is registered only for the bytes with axi_wstrb valid bits.
7.	Interleaving data	The slave interface moves to wait state properly and the interleaving data is registered accordingly.
8.	Axi_bready is sent after two clock cycles	The axi_bvalid signal deassertion is dependent on axi_bready and is deasserted after receiving bready signal for one clock cycle
9.	Axi_arvalid kept high for more than two clock cycles	The axi_arready is produced after one clock cycle of receiving the arvalid signal and is high until the arvalid signal is high.
10.	Fixed burst for read address calculation	The read address for fixed burst is registered and transferred to the memory properly.
11.	Incremental burst for write address calculation	The read address is registered and incremented.
12.	Wrap burst for write address calculation	The address is incremented and wraps around to the base address.

8. Simulation Results

Write Burst



Read Burst


Name
Cursor
160,000ps
180,000ps
200,000ps
220,000ps
240,000ps
260,000ps
280,000ps
300,000ps
axi_aclk 0
axi_areset_n 1
axi_arid[3:0] 'h 3
3
axi_araddr[31:0] 'h 00000006
00000006
axi_arlen[7:0] 'h 02
02
axi_arburst[1:0] 'h 2
2
axi_arvalid 0
axi_arready 0
axi_rid[3:0] 'h 0
0
3
0
axi_rdata[31:0] 'h 00000000
00000000
6666
1111
4444
00000000
axi_rlast 0
axi_rvalid 0
axi_rready 0
axi_rresp[1:0] 'h 0
0
slave_raddr[31:0] 'h 00000005
00000000
00000006
0000
00000005
slave_rdata[31:0] 'h 44444444
00000000
6666
1111
44444444
slave_rden 0