AMBA
AXI to AHB Bridge

Date: Dec, 17, 2024

Table of Contents

Introduction 6
AXI to AHB Bridge 6
Interconnect 7
Interface 7
AXI Protocol Specifications: 8
Channels 8
AHB Protocol Specifications 11
Phases 11
Design of Bridge: 13
ARBITER 14
AXI Controller 14
AHB Controller 16
READ and WRITE FIFOs 18
Simulation Results 19
1. Read latency: 19
2. Write latency: 20
3. Reset test: 20
4. Arbiter Test Case 21
5. Write Transfers 21
- Write Burst of 5 beats with okay response 24
- Write fixed burst with 5 beats, okay response and wait states 25
- Write fixed burst with 11 beats and okay response 26
- Write fixed burst with 11 beats, okay response and wait states 27
- Write fixed burst with 11 beats, error response 28
- Write increment burst with 7 beats, with okay response 28
- Write increment burst with 13 beats, with okay response 29
- Write increment burst with 13 beats, with error response 30
- Write increment burst with 13 beats, with okay response and wait states 31
- Write wrap burst with 8 beats, with okay response 32
- Write wrap burst with 8 beats, with error response 32
- Write wrap burst with 8 beats, with okay response and wait state 33

6. Read Transfers 34
- Read response with okay of fixed and of length 9 34
- Read response with error of fixed and length 15 34
- Read response with okay and fixed of length 15 35
- Read response with okay and with increment of length 15 36
- Read response with okay and with increment of length 14 36
- Read response with okay and with wrap of length 16 37
- Read response with okay and with wrap wait state 38

Implementation results 39
1. RTL design: 39
2. Synthesize design 39
3. Implemented design: 39
4. Power Reports 40
5. Utilization Reports 41
6. Timing summary: 42

Conclusion 42

Table of Figures

Figure 1: Soc with different protocols implemented (heterogenous system) 6
Figure 2: interconnect 8
Figure 3: AXI Channels and Transfers 10
Figure 4: Handshaking of AXI protocol 11
Figure 5: AHB Transfer with no wait state 12
Figure 6: Block diagram 13
Figure 7: Arbiter FSM 14
Figure 8: Control logic of AXI (FSM) 15
Figure 9: Control logic of AHB (FSM) 17
Figure 10: Read latency 19
Figure 11: Write latency 20
Figure 12: Reset test 20
Figure 13: Arbiter Verification through multiple request at a time 21
Figure 14: Single write transfer with okay response 21
Figure 15: Single write transfer with error response 22
Figure 16: Single write transfer with okay response and wait states 22
Figure 17: Single read transfer with okay response 23
Figure 18: Single read transfer with error response 23
Figure 19: Single read transfer with error response and wait states 24
Figure 20: Write Burst of 5 beats with okay response 24
Figure 21: Write fixed burst with 5 beats, okay response and wait states 25
Figure 22: Write fixed burst with 11 beats and okay response 26
Figure 23: Write fixed burst with 11 beats, okay response and wait states 27
Figure 24: Write fixed burst with 11 beats, error response 28
Figure 25: Write increment burst with 13 beats, with okay response 29
Figure 26: Write increment burst with 13 beats, with error response 30
Figure 27: Write increment burst with 13 beats, with okay response and wait states 31
Figure 28: Write wrap burst with 8 beats, with okay response 32
Figure 29: Write wrap burst with 8 beats, with error response 32
Figure 30: Write wrap burst with 8 beats, with okay response and wait state 33
Figure 31: Read response with okay of fixed and of length 9 34
Figure 32: Read response with error of fixed and length 15 34
Figure 33: Read response with okay and fixed of length 15 35
Figure 34: Read response with okay and with increment of length 15 36
Figure 35: Read response with okay and with increment of length 14 36
Figure 36: Read response with okay and with wrap of length 16 37
Figure 37: Read response with okay and with wrap wait state 38
Figure 38: RTL design 39
Figure 39: Synthesize design 39
Figure 40: Implemented design 40
Figure 41: Power Hierarchy Report 40
Figure 42: Power summary report 40
Figure 43: Utilization Hierarchy Report 41

Figure 44: Utilization summary report 41
Figure 45: Timing summary 42

Introduction

Modern System-on-Chip (SoC) designs involve multiple protocols for efficient and effective data communication between different subsystems. In context of communication between different subsystems operating at different frequencies, with different communication protocols and different supported bandwidths, this AXI to AHB Bridge is a critical component that enables smooth communication between the AXI masters (high speed, high bandwidth) and AHB slaves (relatively low speed, low bandwidth), ensuring high compatibility, supporting all type of read/write transfers such as single transactions, bursts (fixed, incrementing, wrapping) with variable length and size.

This report discusses the design, simulation, and implementation of the AXI to AHB Bridge, highlighting its features, protocol specifications, control logic, and performance.

AXI to AHB Bridge

A bridge is an important component of SoC with different communication protocols, enabling data exchange. It performs protocol translation/transaction mapping to ensure data transfer operations between systems having different interfaces and protocols. In the context of an AXI to AHB bridge, it basically enables communication between AXI manager/master (CPU, Controller or Interconnect/BUS) and AHB subordinate/slave device (Memory, Peripheral or I/O device) by converting AXI's advanced channel's signals such as independent read/write/response into AHB's single channel. This ensures compatibility between high-performance and power-efficient components, making the bridge an essential element in heterogeneous SoC designs as shown in figure-1 below.

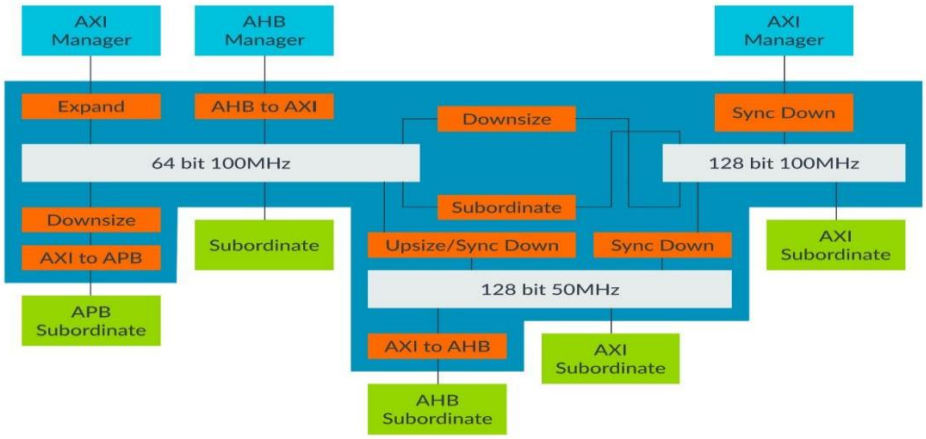

Figure 1: Soc with different protocols implemented (heterogenous system)

AXI to AHB Bridge is a high-performance, configurable, and reliable IP which is designed to integrate into modern System-on-Chip (SoC) architectures. As the design of modern SoC is growing increasingly complex, achieving smooth communication between various on-chip communication protocols becomes very crucial. This bridge addresses this challenge by offering a robust and efficient solution for connecting advanced extensible Interface (AXI) masters to Advanced High-performance Bus (AHB) slaves, ensuring efficient data transfer and seamless protocol conversion.

This bridge supports a wide range of applications, from high-performance processors and embedded systems to custom ASICs, making it a versatile component in advanced semiconductor solutions. Designed for ease of integration and flexibility this bridge reduces development time and simplifies system design while maintaining optimal performance.

The bridge also incorporates advanced features such as configurable parameters such as DATA_WIDTHS, ADDRESS_WIDTHS and STROBE_WIDTHS, ensuring a high degree of adaptability to various design requirements. The RTL is designed in SystemVerilog and its implementation is done on Artix-7 FPGA, however it can be targeted for any FPGA or ASIC. Its modular architecture facilitates straightforward customization and extension.

Interconnect

An interconnect is like the backbone of communication protocols specifically in System-on-Chip (SoC) designs as it enables efficient data exchange between processors, memory, input output devices and peripherals. It ensures scalability, performance optimization, and protocol compatibility, often supporting advanced features like burst transactions parallel read and write transaction requests etc. by managing arbitration, data routing, and synchronization. An interconnect between different subsystems simplifies complex data transfers across various components and protocols, forming the foundation for system communication between heterogeneous systems.

Interface

An interface refers to the collection of input and output ports (signals) that connect the various components of a System-on-Chip (SoC). These ports enable

communication and interaction between different components within the SoC. Simple SoC with multiple masters and slave devices are depicted in figure-2 below.

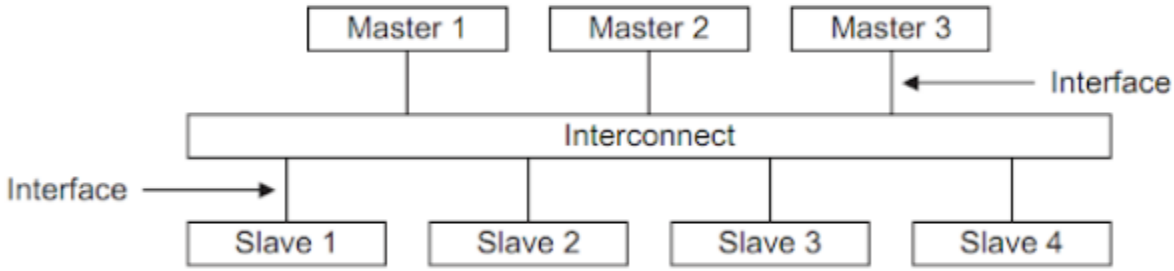

Figure 2: interconnect

AXI Protocol Specifications:

Channels

The AXI protocol uses five independent channels for communication, enabling high-performance, low-latency data transfers with separate handshakes for each channel.

The AXI protocol defines **five independent channels** for communication:

1. **Write Address Channel (AW):** Transmits control/address information for write transactions.
2. **Write Data Channel (W):** Transmits write data.
3. **Write Response Channel (B):** Returns the write transaction response.
4. **Read Address Channel (AR):** Transmits control/address information for read transactions.
5. **Read Data Channel:** Returns read data.

Write Address Channel (AW):

Purpose: Transmits the address and control information for write transactions.

Key Signals:

AWADDR: Write address which can be address for single transfer or starting address for burst transfer with ADDRESS_WIDTH (8,16,32,64)

AWPROT: Protection type (privileged, secure, or instruction/data).

AWLEN: Burst length which decides number of beats in a transaction.

AWSIZE: Transfer size such that size of each beat (which can not be greater than (DATA_WIDTH/8).

AWBURST: Burst type (fixed, incrementing, or wrapping).

AWVALID/AWREADY: Handshake signals for address validity and acceptance.

Write Data Channel (W):

Purpose: Transmits the data to be written.

Key Signals:

WDATA: Write data with different DATA_WIDTHS (8,16,32,64,128)

WSTRB: Write strobes (indicates active bytes).

WLAST: Indicates the last data transfer.

WVALID/WREADY: Handshake signals for data validity and acceptance.

Write Response Channel (B):

Purpose: Returns write transaction status to the master.

Key Signals:

BRESP: Write response (OKAY, EXOKAY, SLVERR, or DECERR).

BVALID/BREADY: Handshake signals for response validity and acceptance.

Read Address Channel (AR):

Purpose: Transmits the address and control information for read transactions.

Key Signals:

ARADDR: Read address which can be address for single transfer or starting address for burst transfer with ADDRESS_WIDTH (8,16,32,64)

ARPROT: Protection type (privileged, secure, or instruction/data).

ARLEN: Burst length which decides number of beats in a transaction.

ARSIZE: Transfer size that size of each beat (which can not be greater than (DATA_WIDTH/8).

ARBURST: Burst type (fixed, incrementing, or wrapping).

ARVALID/ARREADY: Handshake signals for address validity and acceptance.

Read Data Channel:

Purpose: Returns the data to the master during read transactions.

Key Signals:

RDATA: Read data with different DATA_WIDTHS (8,16,32,64,128).

RRESP: Read response (OKAY, EXOKAY, SLVERR, or DECERR).

RLAST: Indicates the last data transfer in a burst.

RVALID/RREADY: Handshake signals for data validity and acceptance.

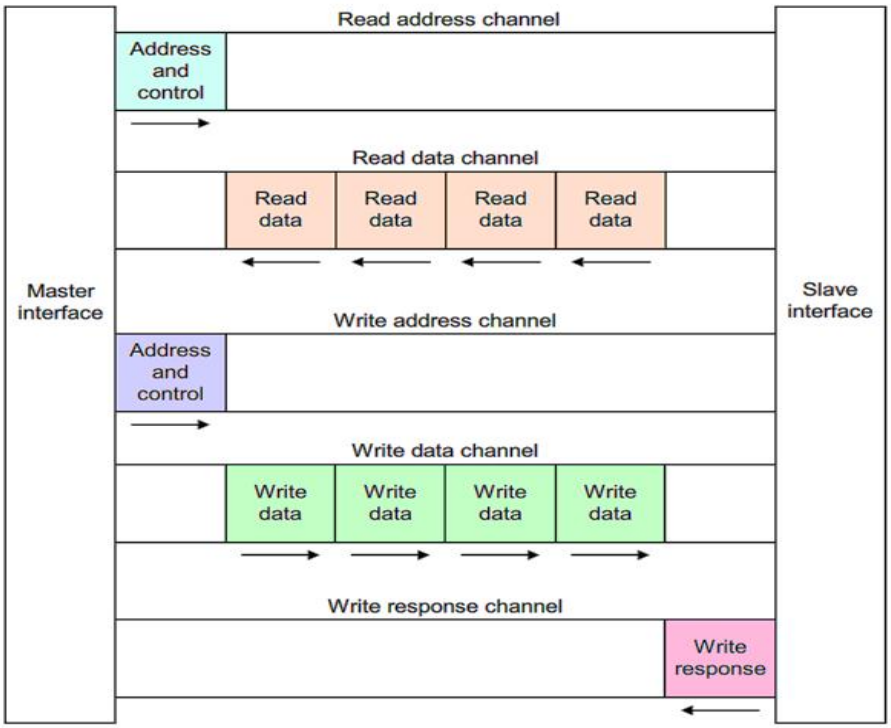

Figure 3: AXI Channels and Transfers

Handshake Mechanism

Each channel uses a two-way handshake mechanism (VALID and READY) to ensure robust data transfer. The handshake process is as follows:

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

Figure 4: Handshaking of AXI protocol

The source asserts VALID to indicate valid data or address.

The destination asserts READY when it is ready to accept the data.

The transfer is complete when both VALID and READY are asserted.

AHB Protocol Specifications

Phases

The AHB protocol operates in distinct phases for each transaction. The primary phases are:

Address Phase

Purpose: The master initiates a transaction by driving the address and control signals.

Key Signals:

HADDR: Address of the transfer with different ADDRESS_WIDTHS (8,16,32,64).

HTRANS: Transfer type (IDLE, BUSY, NONSEQ, or SEQ).

HSIZE: Transfer size (byte, halfword, word, etc.).

HBURST: Burst type (SINGLE, INCR (undefined length), WRAP4, INCR4, WRAP8, INCR8, WRAP16, INCR16).

HWRITE: Indicates read (0) or write (1) operation.

Data Phase

Purpose: Transfers the data between master and slave.

Key Signals:

HWDATA: Data from master to slave during write operations with different DATA_WIDTHS (8,16,32,64,128).

HWSTRB: Write strobes (indicates active bytes).

HRDATA: Data from slave to master during read operations with different DATA_WIDTHS (8,16,32,64,128).

HREADY: Indicates when the slave is ready for the next transaction.

HRESP: Transfer response (0 for OKAY, 1 for ERROR).

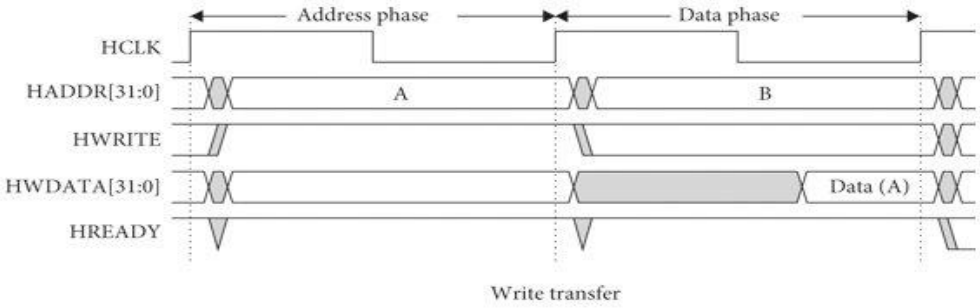

Figure 5: AHB Transfer with no wait state

Channels and Corresponding Signals:

AHB transactions involve a single shared address/control bus and data bus:

Address/Control Bus:

HADDR, HTRANS, HSIZE, HBURST, HWRITE

Data Bus:

HWDATA, HRDATA

Control Signals:

HREADY: Handshake signal to indicate data phase completion.

HRESP: Response signal for error handling.

Handshake Mechanism

HREADY: Indicates the slave's readiness for the next transaction. The transaction proceeds only when HREADY is high.

HRESP: Provides the status of the transaction, signaling whether it was successful (OKAY) or encountered an error (ERROR).

Design of Bridge:

At the frontend of design as shown in figure-7 below, the arbiter monitors read/write channels of AXI along with the state of the previous transfer and accordingly trigger AXI controller about the start of the next transfer. The AXI controller after receiving command from arbiter enables one of the READ/WRITE ADDRESS HOLD REGISTERS to buffer control information (address and transaction type) for the transfer. It also controls write enable and read enable of WRITE FIFO and READ FIFO accordingly. It also triggers AHB Controller to start transaction towards AHB slave. The AHB controller controls AHB ADDRESS & CONTROL REGISTER where address generation for bursts, wrap starting address, wrap boundary calculation and HRESP are controlled. AHB controller controls write enable and read enable READ FIFO and WRITE FIFO accordingly. After completing transfer, both controllers communicate each other through handshake mechanism and provide signal to arbiter so that arbiter become active for next transfer.

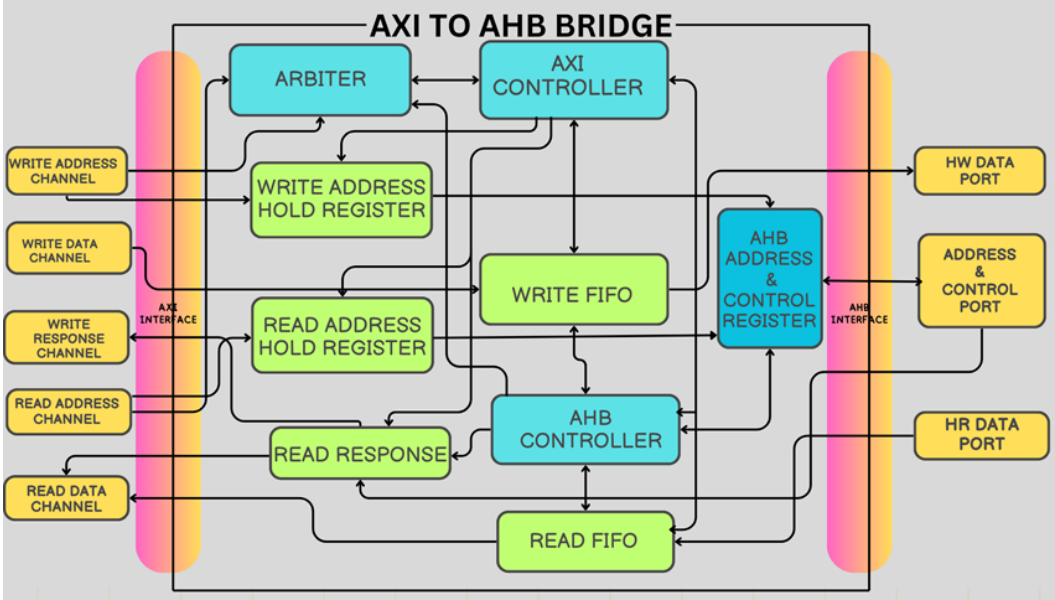

Figure 6: Block diagram

Details of main logical component are further explained below.

ARBITER

A round robin arbiter with initial priority upon reset to write channel has been implemented with Moore FSM having 5 states is shown below in figure-6.

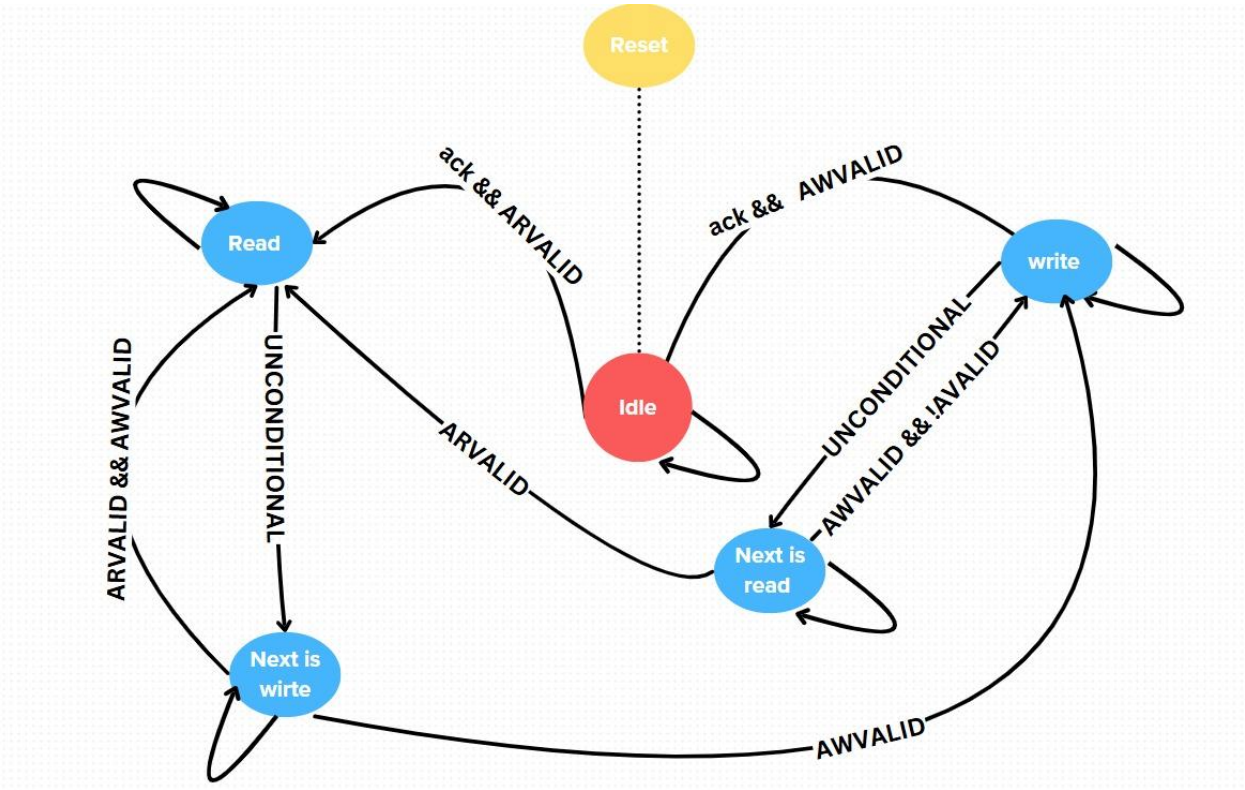

Figure 7: Arbiter FSM

This FSM start in idle state on reset when the ack signal is asserted the request signals (ARVALID and AWVALID) decides which request is to give priority. The priority can be set by memorizing the previous transfer which is achieved through unconditional transition from read and write into Next_is_write and Next_is_read accordingly. In the read state the output start_reading is asserted while in the write state the output start_writing is asserted. These signals trigger AXI Controller which then starts transaction for that channel (read or write).

AXI Controller

The AXI controller is implemented with Moore FSM consist of 11 states, having two separate branches from idle which decide the type of transfer (read/write). It

initially gets trigger from arbiter which makes sure that this can safely transit in one branch accordingly. FSM design of the AXI controller is shown in figure-7 below.

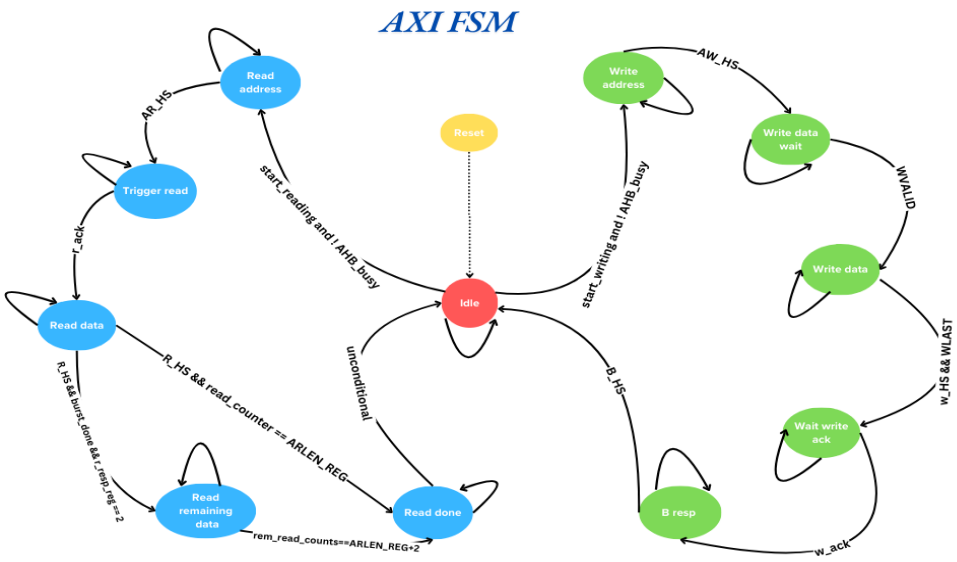

Figure 8: Control logic of AXI (FSM)

State Transitions are self-explanatory while in each states different control signals are generated to perform different functions which are as follows

IDLE: This state reset all read/write address hold registers and counters that is used in AXI controller and wiat for trigger (start_reading and start_writing)

Write_Address: this state provides handshake to AW channel and latches control information of AW channel in a buffer.

Write_data_wait: this state waits for valid data start in case of write transfer only.

Write_data: this state provide handshake to W channel and latches incoming valid data and strobes into WRITE FIFO. It also provides trigger to AHB controller to start writing data on AHB interface. It also counts the data that has been written to WRITE FIFO. After writing the whole transaction into FIFO the transition take place into wait_write_ack state.

Wait_write_ack: this state waits for AHB controller trigger signal (the signal AHB provides when it write data successfully from WRITE FIFO into AHB interface or an error response received from AHB slave). Upon receiving that signal transition take place into B_resp state.

B_resp: This state put the response coming from AHB slave into B channel and wait for handshake to complete. Upon completion of handshake, it transits to idle. This state also provide signal to AHB controller that writing we are done with the transfer so that AHB slave go to its idle state.

Read_address: this state provides handshake to AR channel and latches control information of AR channel in a buffer.

Trigger_Read: this state trigger AHB controller to start reception from AHB slave. It monitors ack signal coming from AHB slave (AHB will assert that signal once it reads data and write it into READ FIFO). Upon reception of ack signal, it transits to read data state.

Read_data: This state read data from READ FIFO and provide it to AXI master R channel with response that is captured from AHB slave. Upon reading all data successfully from READ FIFO it transits to read_done state. If error response coming from AHB during burst transaction it transits to Read_remaining_data state.

Read_remaining_data: this state completes the read transfer with garbage data and error with each data that was not provided by AHB slave after error response. After completing the burst, it transits to Read_done state.

Read_done: this state provides done signal to AHB controller and transit unconditionally to idle state for next transfer.

AHB Controller

The HB controller is implemented with Moore FSM consist of 7 states, having two separate branches from idle which decide the type of transfer (read/write). It initially gets trigger from AXI controller which makes sure that this can safely transit in one branch accordingly. FSM design of the AHB controller is shown in figure-8 below.

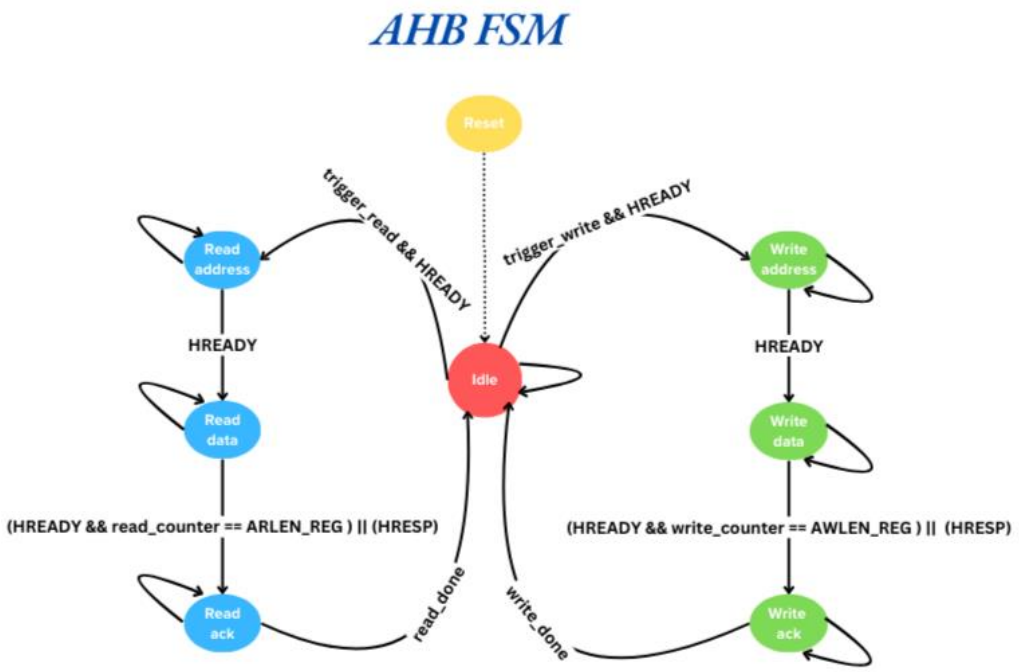

Figure 9: Control logic of AHB (FSM)

State Transitions are self-explanatory while in each states different control signals are generated to perform different functions which are as follows

IDLE: This state reset all counters and hold registers and counters that is used in AHB controller and wait for trigger (trigger_write and trigger_read) and HREADY (AHB slave is ready). If each condition occurs then it transits to one branch safely.

Write_Address: This state asserts HWRITE and provides starting address and control parameters of the transfer as per mapping of AXI to AHB such as (single transfer, fixed, burst, incrementing burst and wrapping burst). It also calculates wrap_start, address and wrap boundary for wrapping burst that is used in subsequent state. It monitors HREADY if it is still asserted it transit into write_data state.

Write_data: This states continuously monitors HREADY, read data from WRITE FIFO, count number of transfers, and calculate next address according to transfer type (single, fixed burst, incrementing burst and wrapping burst). If HREADY is still asserted it provides the information (data, transfer_type and next address) to AHB slave. Upon completion of transfer, it transits to write_ack state.

Write_ack: this state provides ack signal to AXI controller that writing is done and then upon handshake with AXI controller it transits to idle state.

Read_address: This state deasserts HWRITE and provides starting address and control parameters of the transfer as per mapping of AXI to AHB such as (single transfer, fixed, burst, incrementing burst and wrapping burst). It also calculates wrap_start, address and wrap boundary for wrapping burst that is used in subsequent state. It monitors HREADY if it is still asserted it transit into read_data state.

Read_data: This states continuously monitors HREADY, write data into READ FIFO, count number of transfers, and calculate next address according to transfer type (single, fixed burst, incrementing burst and wrapping burst). If HREADY is still asserted it takes HRDATA and write it into RAD FIFO and provides information (data, transfer_type and next address) to AHB slave. states also trigger AXI controller once it writes initial data into READ FIFO so that AXI starts providing the received data stored in READ FIFO to AXI R channel. Upon completion of transfer, it transits to read_ack state.

Read_ack: this state give signal to AXI controller that reading has been completed. It then waits for AXI ack signal. Once AXI transmit all the data on R channel it receives ack and transits into idle state.

READ and WRITE FIFOs

Two synchronous FIFOs with asynchronous reset has been used in the design. They are generated using vivado IP generator having fixed depth = 64 and data_width = 36 bits. These FIFOs can be configured for different data widths and depths as per requirement. But as pre configuration was required to use Xilinx IP, that's why it is configured with the above-mentioned parameters.

It is configured so as it handles data_width = 32 and strobe_width = 4. WRITE FIFO stores data and strobes receiving from AXI master W channel and READ FIFO stores data receiving from AHB slave without any loss of data and to reduce bridge latency.

For targeting ASICs or any other vendor FPGAs, these FIFO can easily be replaced accordingly while making the functionality ineffective.

Output signals empty and full of both fifos are monitored by AXI and AHB controllers while AXI controller controls write enable of WRITE FIFO and read enable of READ FIFO. On the other hand AHB conteroller controls write enable of READ FIFO and read enable of WRITE FIFO. And this is certain as AXI controller writes data to WRITE FIFO and reads data from READ FIFO while AHB controller writes data to READ FIFO and reads data from WRITE FIFO.

Simulation Results

1. **Read latency:** Simulation results demonstrate a small read latency under various conditions.

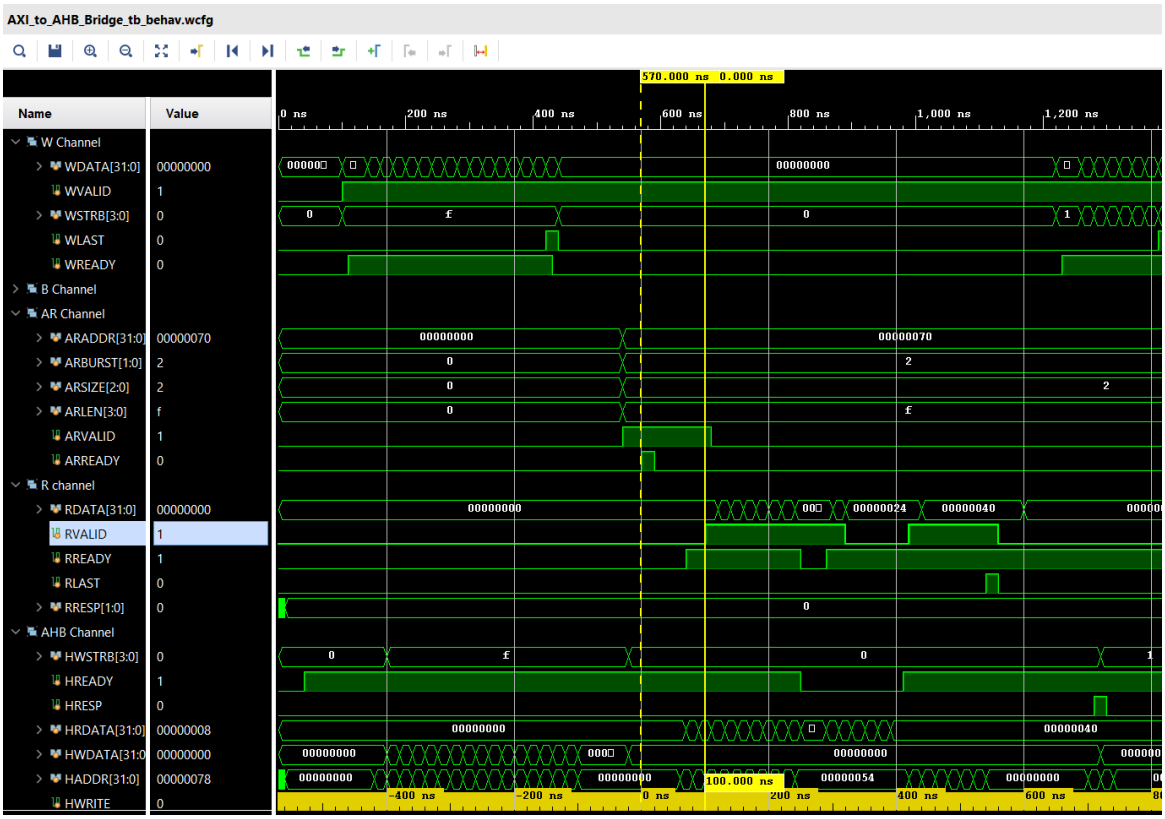

Figure 10: Read latency

2. **Write latency:** Write operations have efficient throughput and small delay.

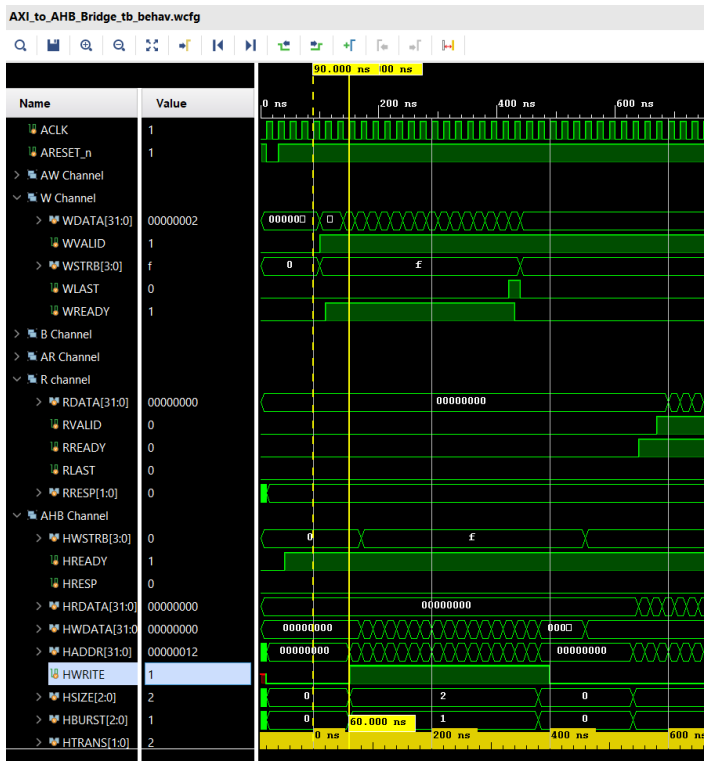

Figure 11: Write latency

3. **Reset test:** Verifies bridge behavior under reset conditions.

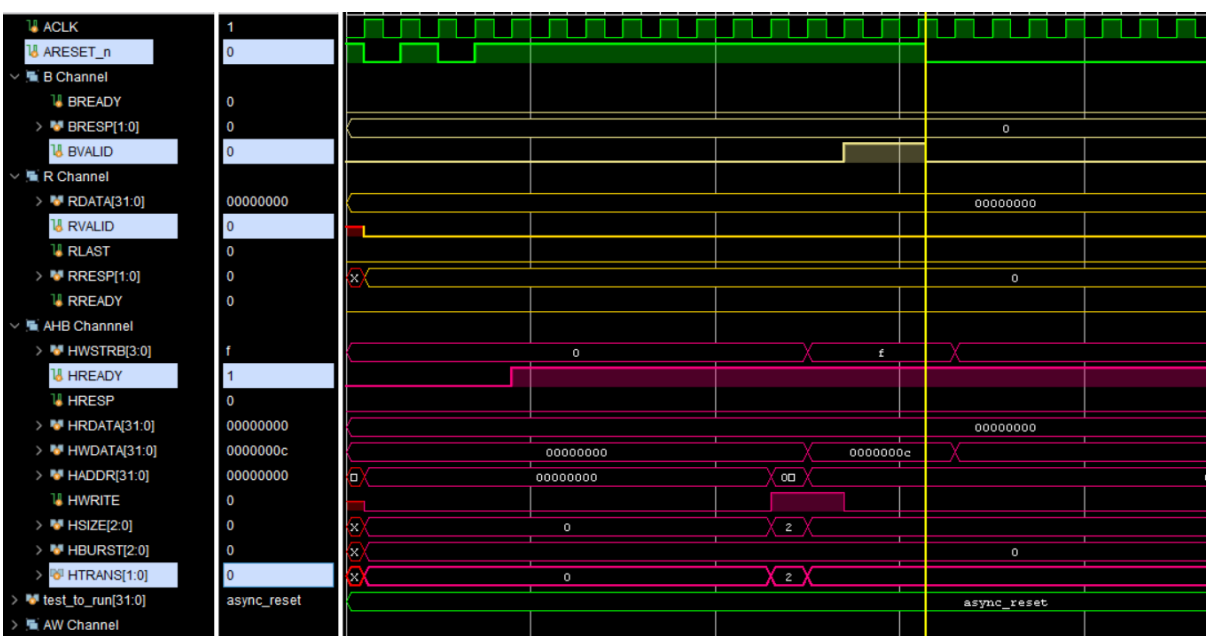

Figure 12: Reset test

```
Running Asynchronous Reset Test.
AHB Address = 4, AXI Address = 4 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
Single Write Transaction Test Passed.
Reset Test Passed.
```

4. Arbiter Test Case

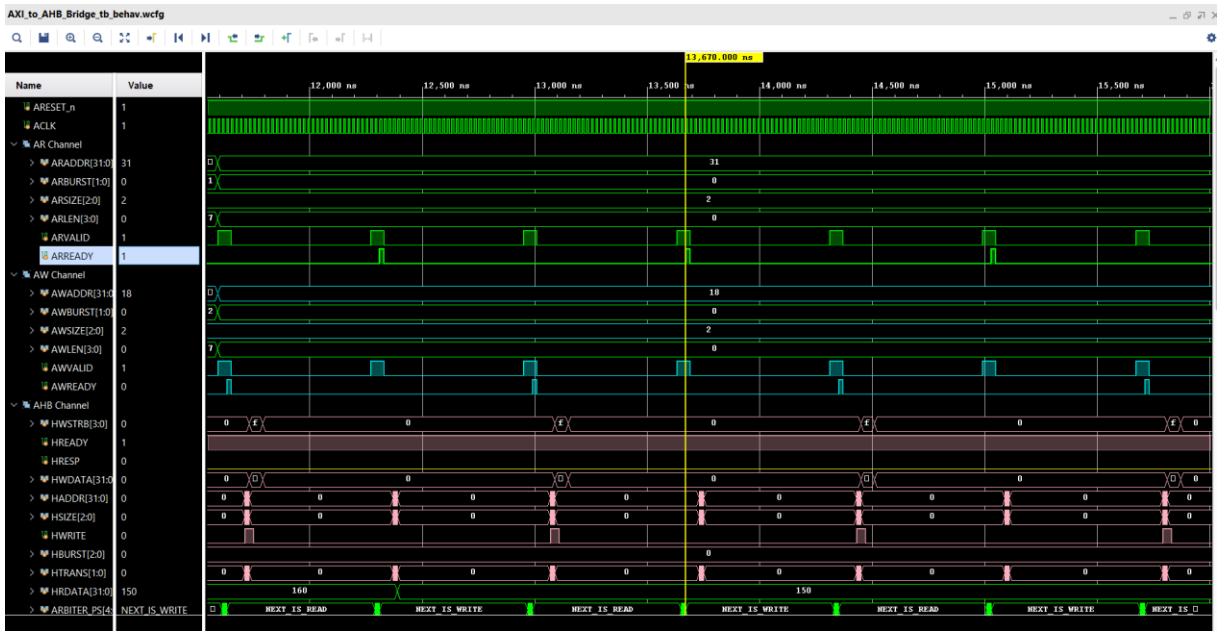

Figure 13: Arbiter Verification through multiple request at a time

5. Write Transfers

- Single write transfer with okay response:

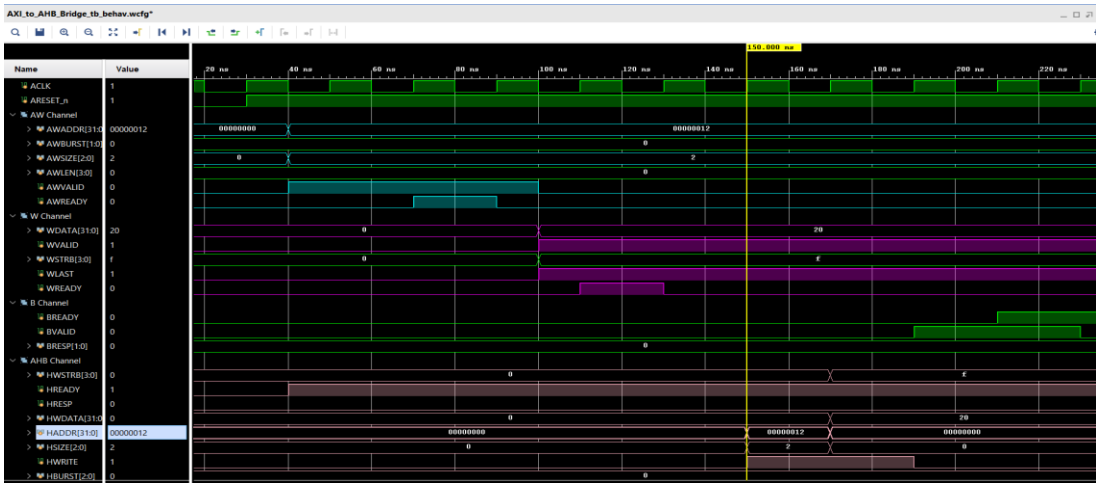

Figure 14: Single write transfer with okay response

- Single write transfer with error response:

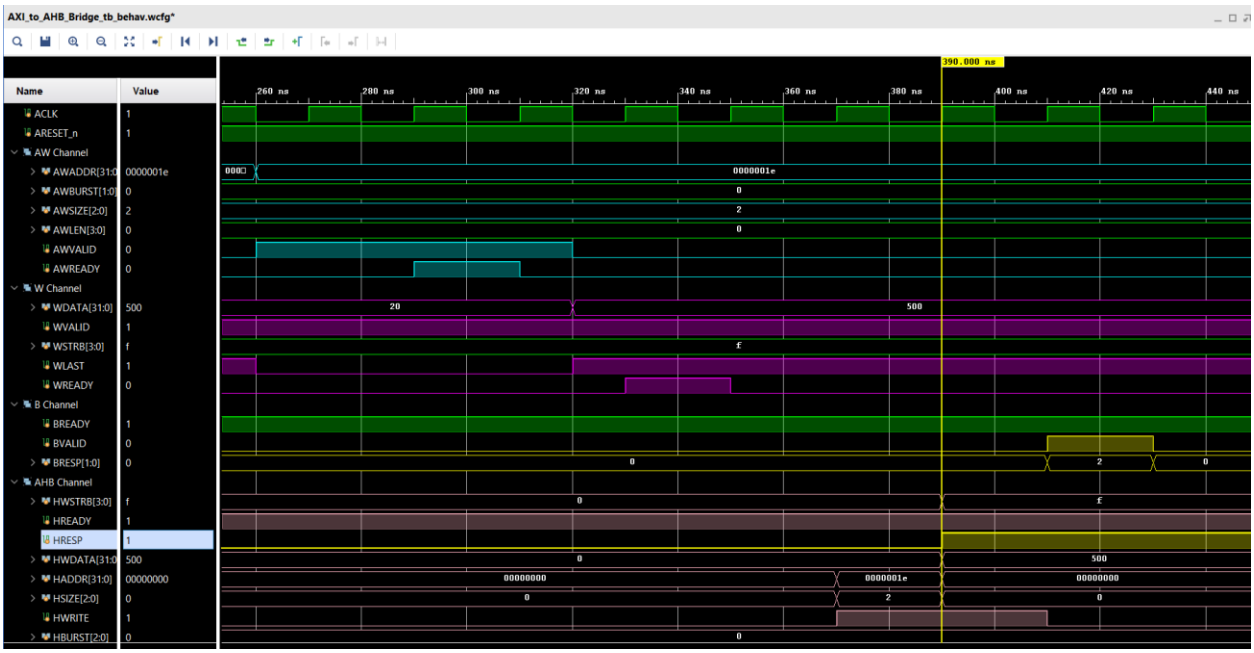

Figure 15: Single write transfer with error response

- Single write transfer with okay response and wait states:

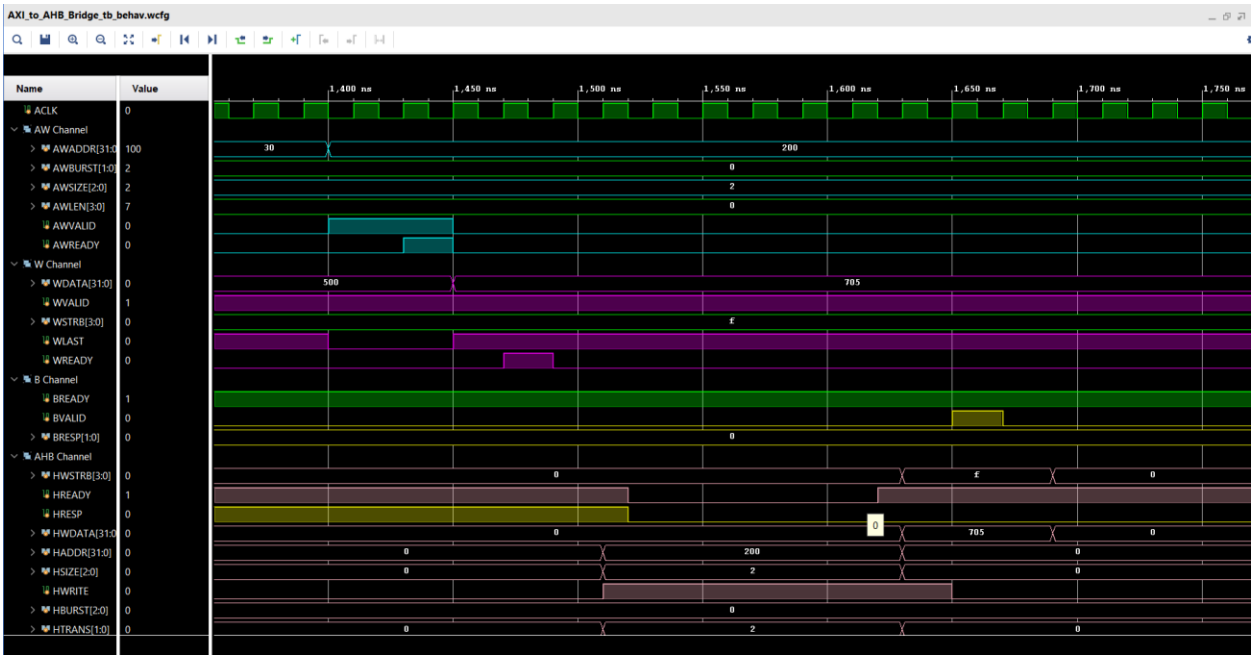

Figure 16: Single write transfer with okay response and wait states

- Single read transfer with okay response:

Figure 17: Single read transfer with okay response

- Single read transfer with error response:

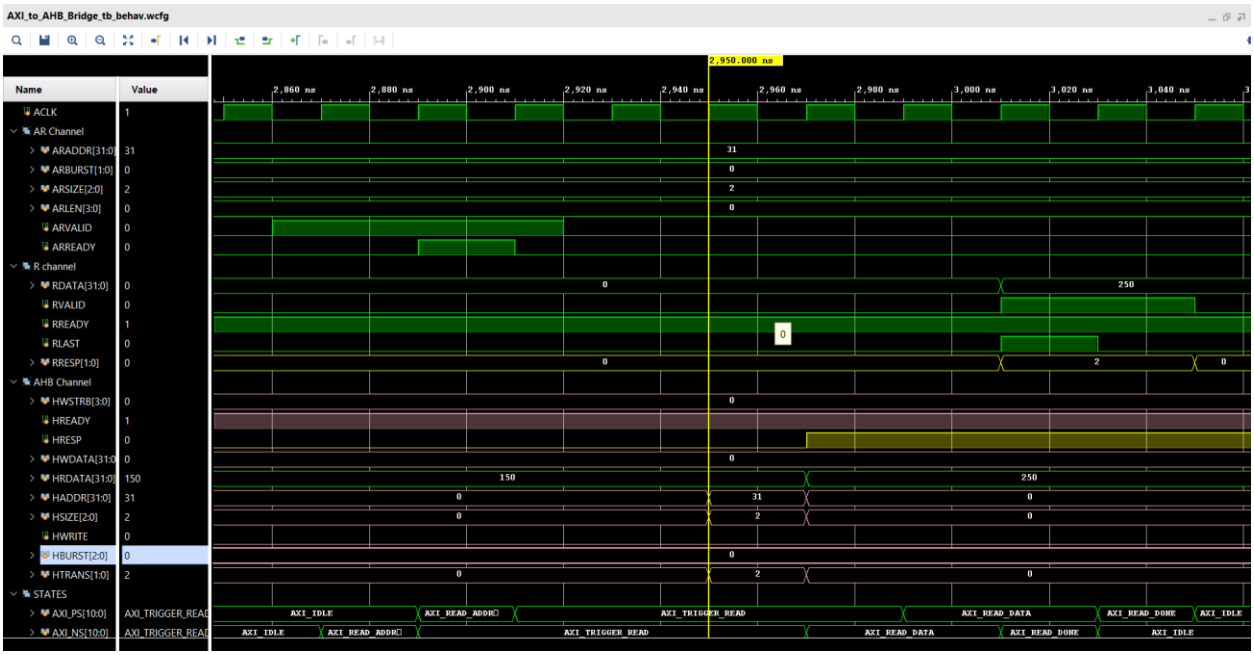

Figure 18: Single read transfer with error response

- Single read transfer with error response and wait states:

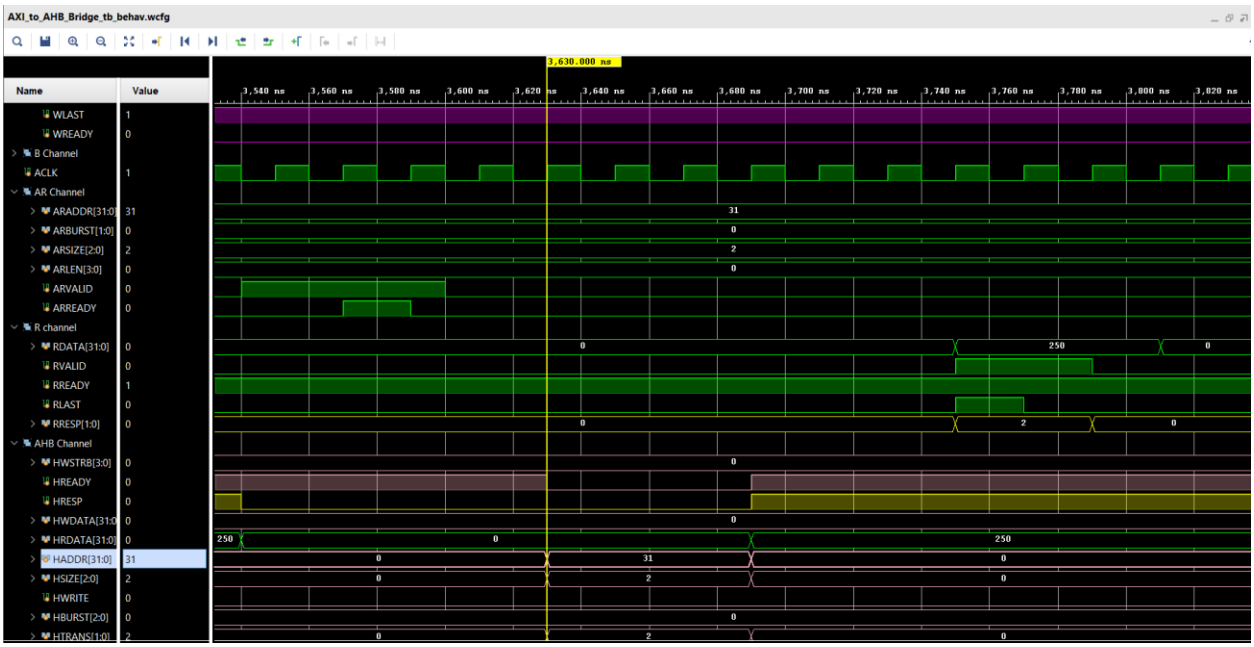

Figure 19: Single read transfer with error response and wait states

- **Write Burst of 5 beats with okay response**

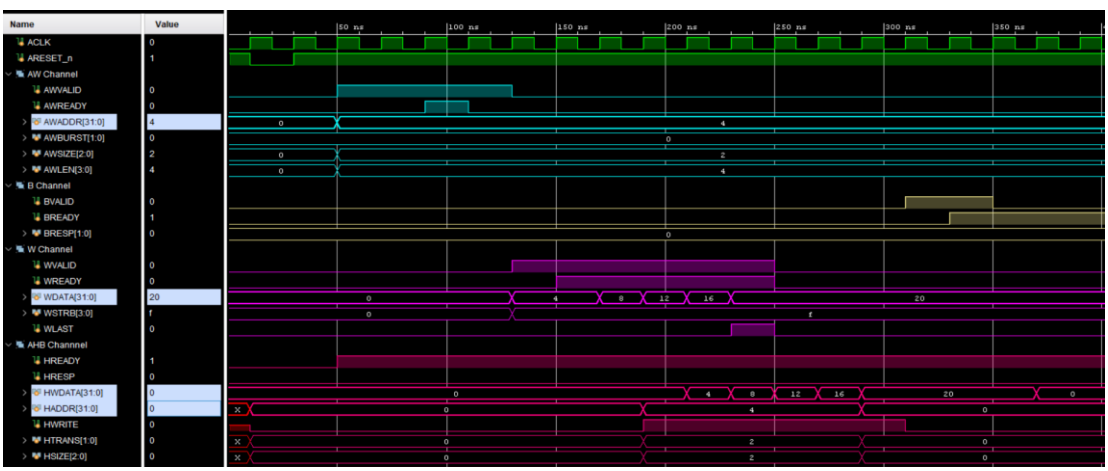

Figure 20: Write Burst of 5 beats with okay response

```
Running Fixed Burst Write Transaction Test
AHB Address = 4, AXI Address = 4 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 4, AXI Address = 4 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 4, AXI Address = 4 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 4, AXI Address = 4 at time: 270
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 290
AHB Address = 4, AXI Address = 4 at time: 290
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 310
Burst Fixed Write Transaction Test Passed.

Simulation Complete.
```

- **Write fixed burst with 5 beats, okay response and wait states**

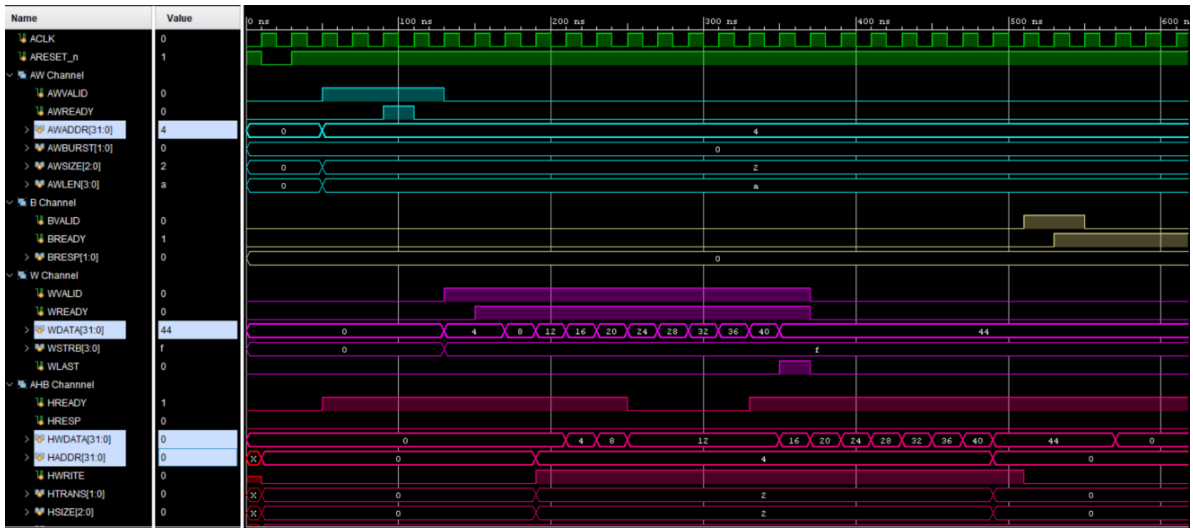

Figure 21: Write fixed burst with 5 beats, okay response and wait states

```
Running Fixed Write Transaction Test With Changing HREADY
AHB Address = 4, AXI Address = 4 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 4, AXI Address = 4 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 4, AXI Address = 4 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 4, AXI Address = 4 at time: 350
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 370
AHB Address = 4, AXI Address = 4 at time: 370
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 390
AHB Address = 4, AXI Address = 4 at time: 390
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 410
AHB Address = 4, AXI Address = 4 at time: 410
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 430
AHB Address = 4, AXI Address = 4 at time: 430
Write Transfer Passed. AHB Data = 32, AXI Data = 32 at time: 450
AHB Address = 4, AXI Address = 4 at time: 450
Write Transfer Passed. AHB Data = 36, AXI Data = 36 at time: 470
AHB Address = 4, AXI Address = 4 at time: 470
Write Transfer Passed. AHB Data = 40, AXI Data = 40 at time: 490
AHB Address = 4, AXI Address = 4 at time: 490
Write Transfer Passed. AHB Data = 44, AXI Data = 44 at time: 510
Burst Fixed Write Transaction Test Passed.

Simulation Complete.
```

- **Write fixed burst with 11 beats and okay response**

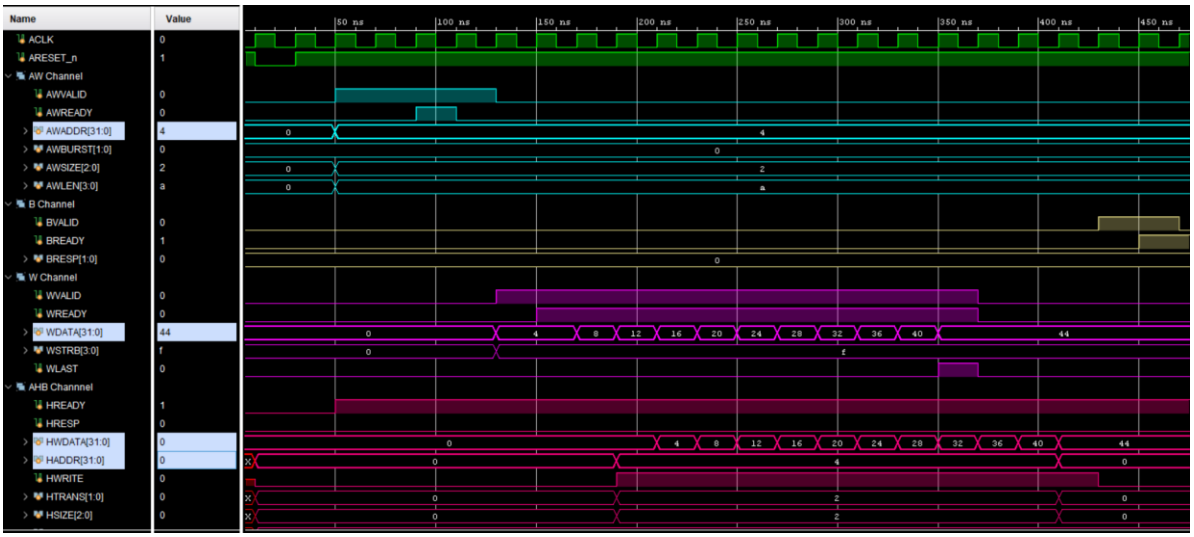

Figure 22: Write fixed burst with 11 beats and okay response

```
Running Fixed Burst Write Transaction Test
AHB Address = 4, AXI Address = 4 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 4, AXI Address = 4 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 4, AXI Address = 4 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 4, AXI Address = 4 at time: 270
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 290
AHB Address = 4, AXI Address = 4 at time: 290
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 310
AHB Address = 4, AXI Address = 4 at time: 310
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 330
AHB Address = 4, AXI Address = 4 at time: 330
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 350
AHB Address = 4, AXI Address = 4 at time: 350
Write Transfer Passed. AHB Data = 32, AXI Data = 32 at time: 370
AHB Address = 4, AXI Address = 4 at time: 370
Write Transfer Passed. AHB Data = 36, AXI Data = 36 at time: 390
AHB Address = 4, AXI Address = 4 at time: 390
Write Transfer Passed. AHB Data = 40, AXI Data = 40 at time: 410
AHB Address = 4, AXI Address = 4 at time: 410
Write Transfer Passed. AHB Data = 44, AXI Data = 44 at time: 430
Burst Fixed Write Transaction Test Passed.

Simulation Complete.
```

- **Write fixed burst with 11 beats, okay response and wait states**

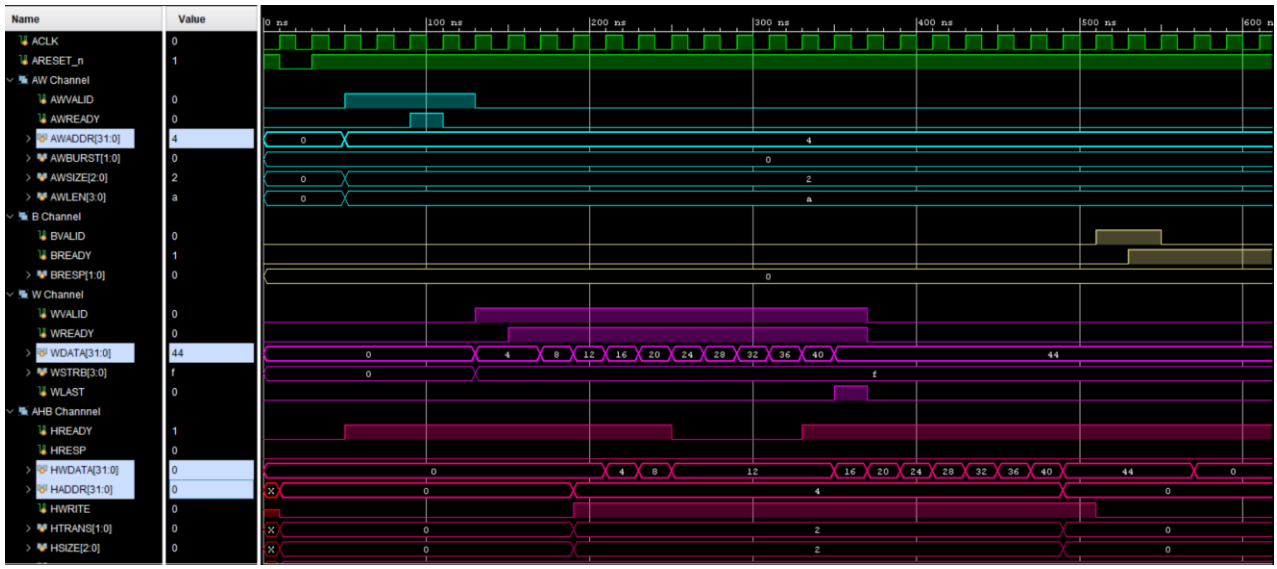

Figure 23: Write fixed burst with 11 beats, okay response and wait states

```
Running Fixed Write Transaction Test With Changing HREADY
AHB Address = 4, AXI Address = 4 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 4, AXI Address = 4 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 4, AXI Address = 4 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 4, AXI Address = 4 at time: 350
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 370
AHB Address = 4, AXI Address = 4 at time: 370
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 390
AHB Address = 4, AXI Address = 4 at time: 390
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 410
AHB Address = 4, AXI Address = 4 at time: 410
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 430
AHB Address = 4, AXI Address = 4 at time: 430
Write Transfer Passed. AHB Data = 32, AXI Data = 32 at time: 450
AHB Address = 4, AXI Address = 4 at time: 450
Write Transfer Passed. AHB Data = 36, AXI Data = 36 at time: 470
AHB Address = 4, AXI Address = 4 at time: 470
Write Transfer Passed. AHB Data = 40, AXI Data = 40 at time: 490
AHB Address = 4, AXI Address = 4 at time: 490
Write Transfer Passed. AHB Data = 44, AXI Data = 44 at time: 510
Burst Fixed Write Transaction Test Passed.

Simulation Complete.
```

- **Write fixed burst with 11 beats, error response**

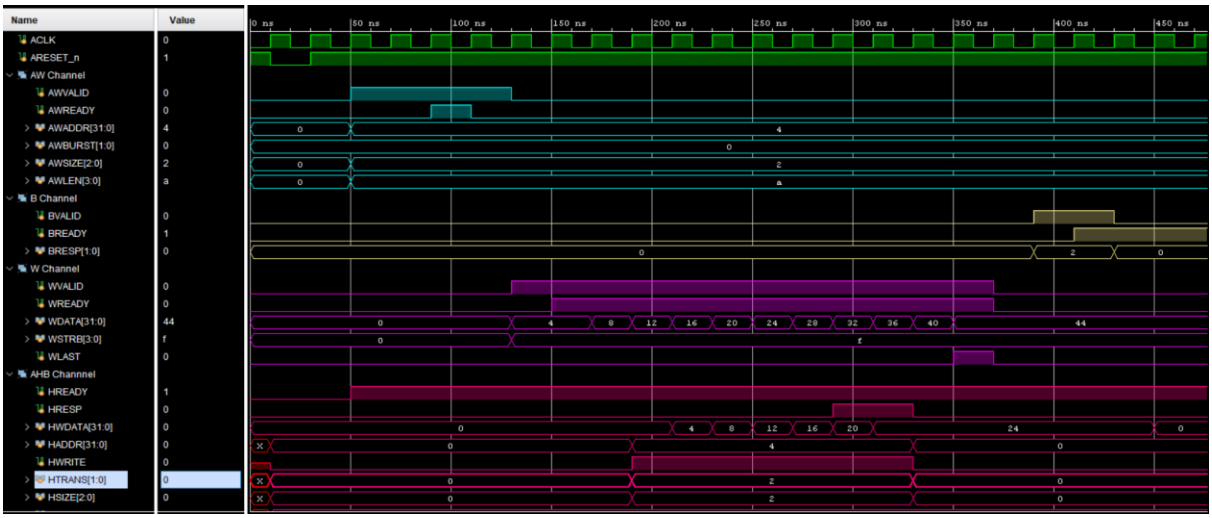

Figure 24: Write fixed burst with 11 beats, error response

```
Running Fixed Read Transaction with changing HRESP Test
AHB Address = 4, AXI Address = 4 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 4, AXI Address = 4 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 4, AXI Address = 4 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 4, AXI Address = 4 at time: 270
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 290
AHB Address = 4, AXI Address = 4 at time: 290
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 310
HTRANS Passed. Expected = 0, Actual = 0 at time: 330
Burst Fixed Write Transaction Test Passed.

Simulation Complete.
```

- **Write increment burst with 7 beats, with okay response**

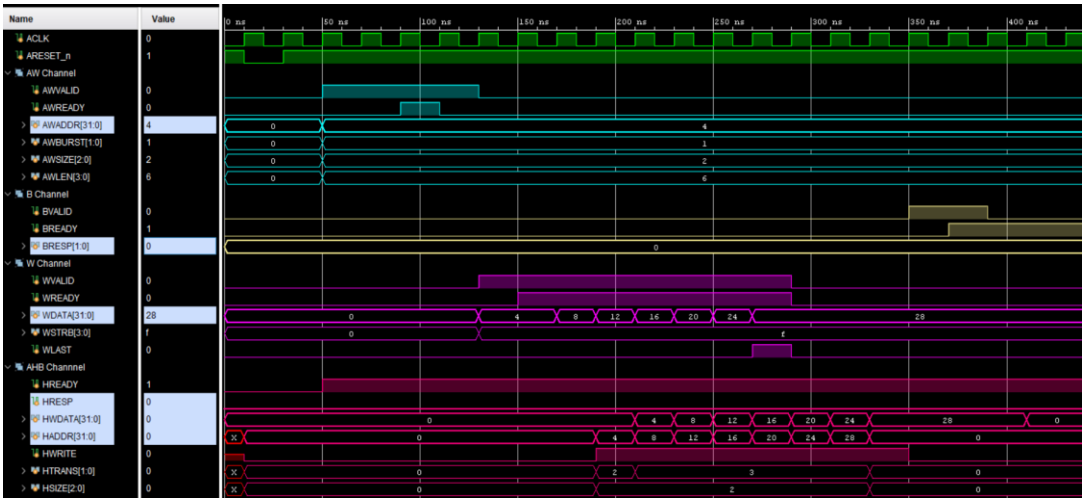

Figure 25: Write increment burst with 7 beats, with okay response

```
Running Increment Write Transaction Test
AHB Address = 4, AXI Address = 4 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 8, AXI Address = 8 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 12, AXI Address = 12 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 16, AXI Address = 16 at time: 270
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 290
AHB Address = 20, AXI Address = 20 at time: 290
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 310
AHB Address = 24, AXI Address = 24 at time: 310
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 330
AHB Address = 28, AXI Address = 28 at time: 330
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 350
Burst Increment Write Transaction Test Passed.

Simulation Complete.
```

- **Write increment burst with 13 beats, with okay response**

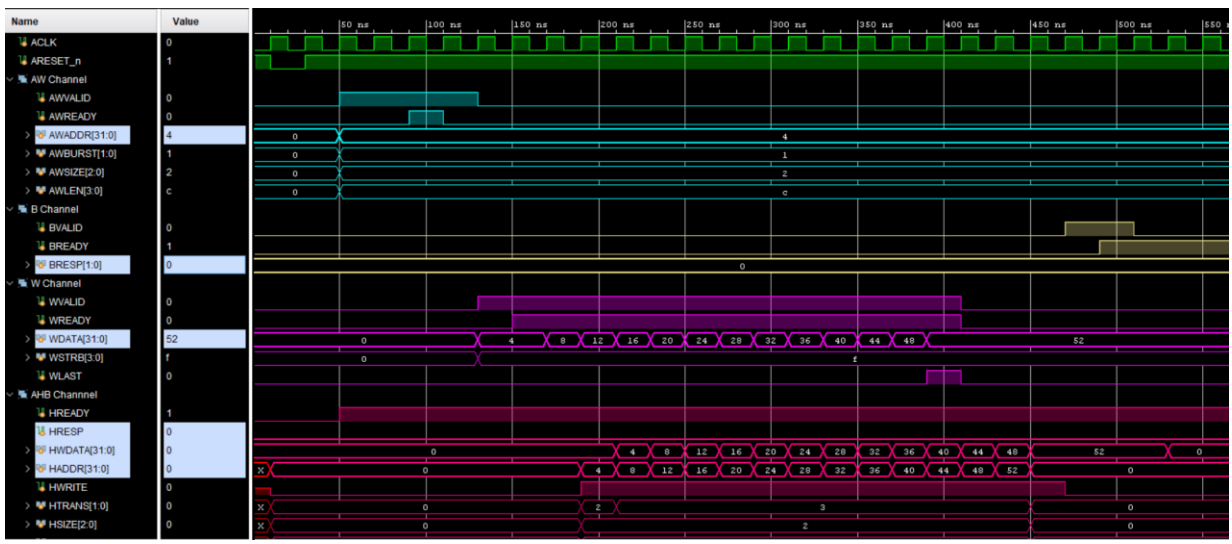

Figure 25: Write increment burst with 13 beats, with okay response

```
Running Increment Write Transaction Test
AHB Address = 4, AXI Address = 4 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 8, AXI Address = 8 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 12, AXI Address = 12 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 16, AXI Address = 16 at time: 270
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 290
AHB Address = 20, AXI Address = 20 at time: 290
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 310
AHB Address = 24, AXI Address = 24 at time: 310
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 330
AHB Address = 28, AXI Address = 28 at time: 330
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 350
AHB Address = 32, AXI Address = 32 at time: 350
Write Transfer Passed. AHB Data = 32, AXI Data = 32 at time: 370
AHB Address = 36, AXI Address = 36 at time: 370
Write Transfer Passed. AHB Data = 36, AXI Data = 36 at time: 390
AHB Address = 40, AXI Address = 40 at time: 390
Write Transfer Passed. AHB Data = 40, AXI Data = 40 at time: 410
AHB Address = 44, AXI Address = 44 at time: 410
Write Transfer Passed. AHB Data = 44, AXI Data = 44 at time: 430
AHB Address = 48, AXI Address = 48 at time: 430
Write Transfer Passed. AHB Data = 48, AXI Data = 48 at time: 450
AHB Address = 52, AXI Address = 52 at time: 450
Write Transfer Passed. AHB Data = 52, AXI Data = 52 at time: 470
Burst Increment Write Transaction Test Passed.

Simulation Complete.
```

- **Write increment burst with 13 beats, with error response**

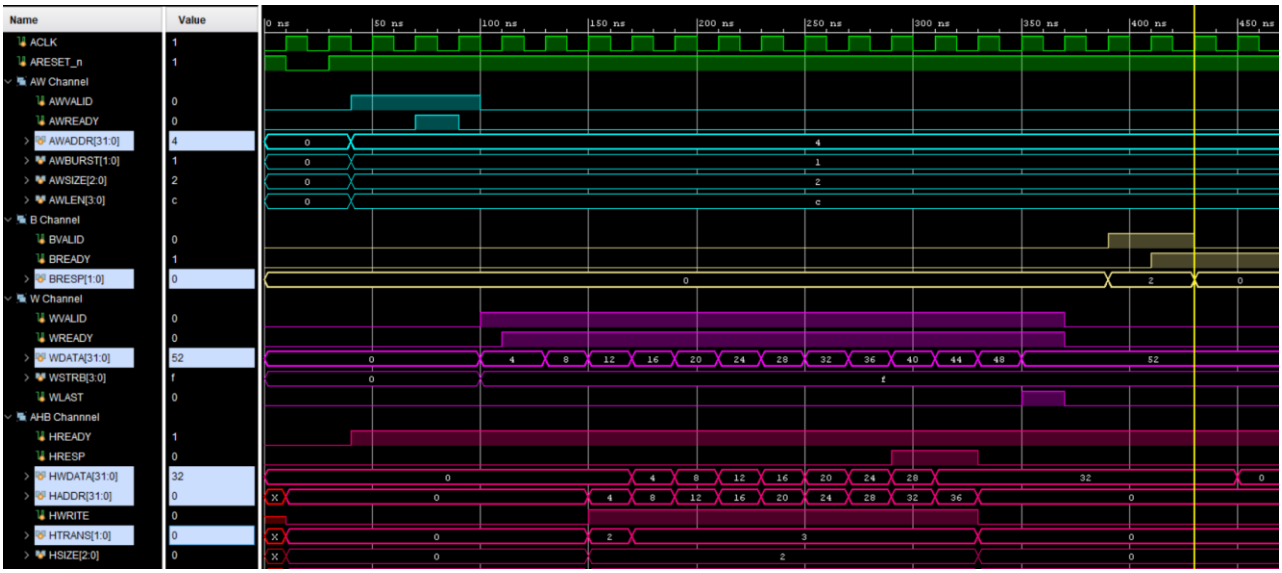

Figure 26: Write increment burst with 13 beats, with error response

```
Running Increment Write Transaction With Changing HRESP Test
AHB Address = 4, AXI Address = 4 at time: 170
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 190
AHB Address = 8, AXI Address = 8 at time: 190
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 210
AHB Address = 12, AXI Address = 12 at time: 210
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 230
AHB Address = 16, AXI Address = 16 at time: 230
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 250
AHB Address = 20, AXI Address = 20 at time: 250
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 270
AHB Address = 24, AXI Address = 24 at time: 270
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 290
AHB Address = 28, AXI Address = 28 at time: 290
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 310
HTRANS Passed. Expected = 0, Actual = 0 at time: 330
Burst Increment Write Transaction with Changing HRESP Test Passed.

Simulation Complete.
```

- **Write increment burst with 13 beats, with okay response and wait states**

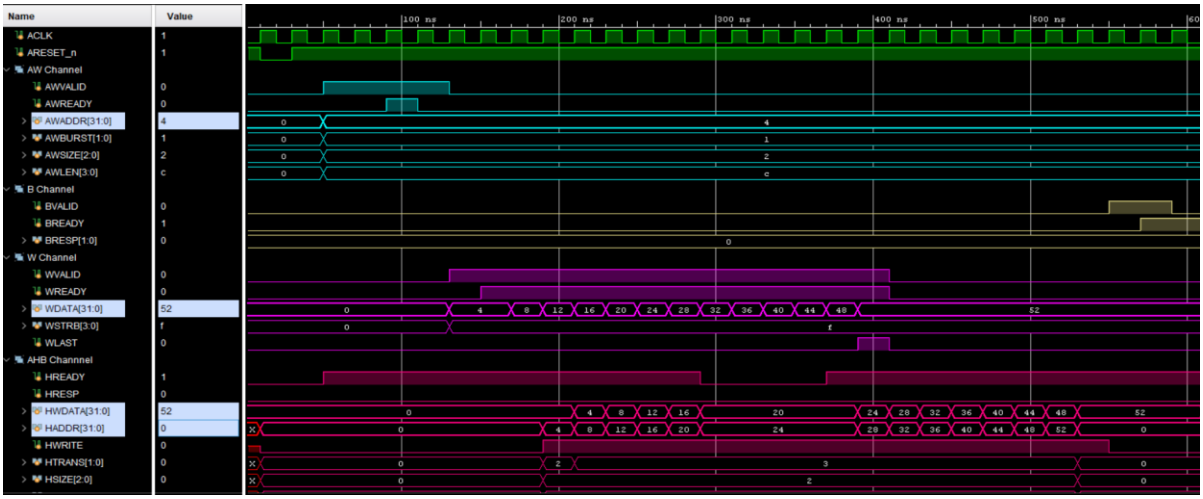

Figure 27: Write increment burst with 13 beats, with okay response and wait states

```
Running Increment Write Transaction Test With Changing HREADY
AHB Address = 4, AXI Address = 4 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 8, AXI Address = 8 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 12, AXI Address = 12 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 16, AXI Address = 16 at time: 270
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 290
AHB Address = 20, AXI Address = 20 at time: 290
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 310
AHB Address = 24, AXI Address = 24 at time: 390
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 410
AHB Address = 28, AXI Address = 28 at time: 410
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 430
AHB Address = 32, AXI Address = 32 at time: 430
Write Transfer Passed. AHB Data = 32, AXI Data = 32 at time: 450
AHB Address = 36, AXI Address = 36 at time: 450
Write Transfer Passed. AHB Data = 36, AXI Data = 36 at time: 470
AHB Address = 40, AXI Address = 40 at time: 470
Write Transfer Passed. AHB Data = 40, AXI Data = 40 at time: 490
AHB Address = 44, AXI Address = 44 at time: 490
Write Transfer Passed. AHB Data = 44, AXI Data = 44 at time: 510
AHB Address = 48, AXI Address = 48 at time: 510
Write Transfer Passed. AHB Data = 48, AXI Data = 48 at time: 530
AHB Address = 52, AXI Address = 52 at time: 530
Write Transfer Passed. AHB Data = 52, AXI Data = 52 at time: 550
Burst Increment Write Transaction with Changing HREADY Test Passed.

Simulation Complete.
```

- **Write wrap burst with 8 beats, with okay response**

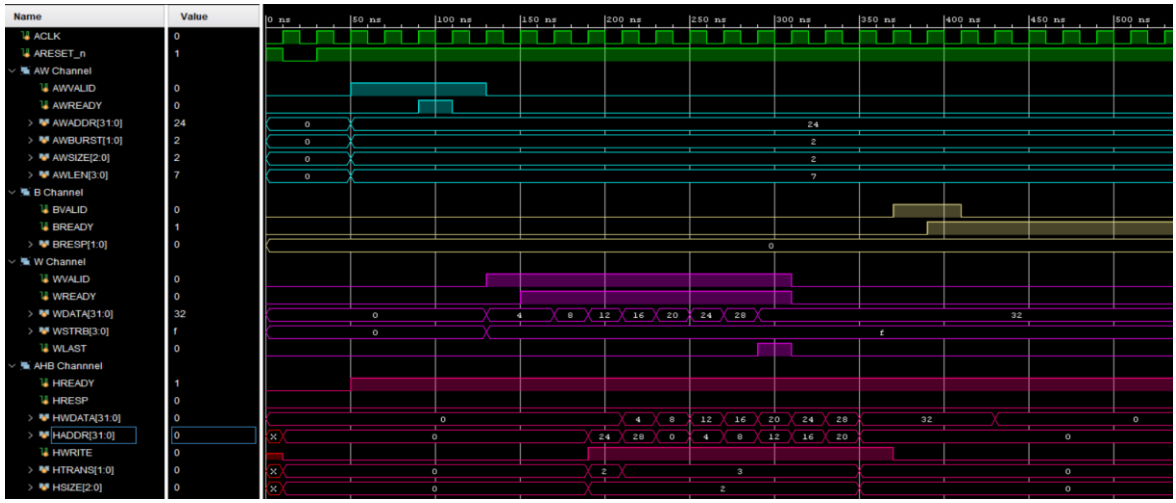

Figure 28: Write wrap burst with 8 beats, with okay response

```
Running Wrap Burst Write Transaction Test
AHB Address = 24, AXI Address = 24 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 28, AXI Address = 28 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 0, AXI Address = 0 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 4, AXI Address = 4 at time: 270
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 290
AHB Address = 8, AXI Address = 8 at time: 290
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 310
AHB Address = 12, AXI Address = 12 at time: 310
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 330
AHB Address = 16, AXI Address = 16 at time: 330
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 350
AHB Address = 20, AXI Address = 20 at time: 350
Write Transfer Passed. AHB Data = 32, AXI Data = 32 at time: 370
Burst Wrap Write Transaction Test Passed.
```

- **Write wrap burst with 8 beats, with error response**

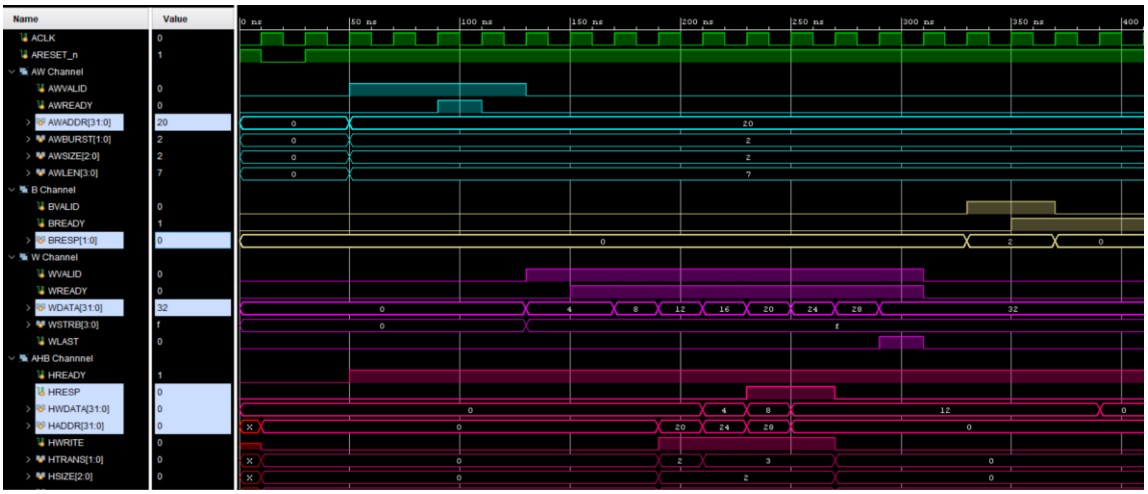

Figure 29: Write wrap burst with 8 beats, with error response

```
Running Wrap Write Transaction With Changing HRESP Test
AHB Address = 20, AXI Address = 20 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 24, AXI Address = 24 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
HTRANS Passed. Expected = 0, Actual = 0 at time: 270
Burst Wrap Write Transaction With Changing HRESP Test Passed.

Simulation Complete.
```

- **Write wrap burst with 8 beats, with okay response and wait state**

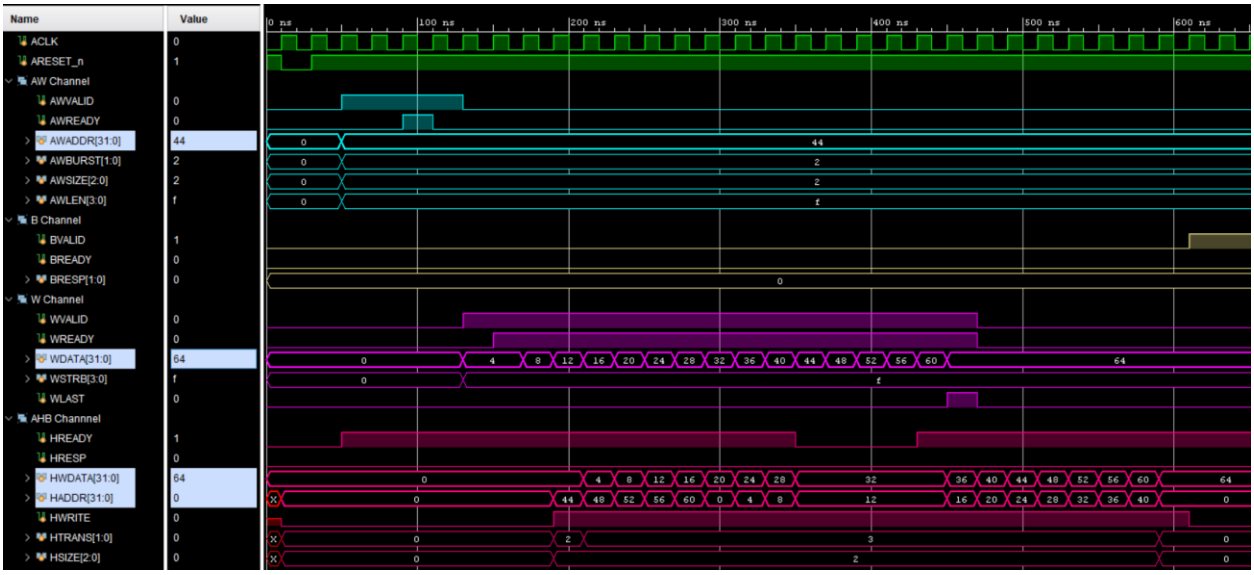

Figure 30: Write wrap burst with 8 beats, with okay response and wait state

```
Running Wrap Write Transaction Test With Changing HREADY
AHB Address = 44, AXI Address = 44 at time: 210
Write Transfer Passed. AHB Data = 4, AXI Data = 4 at time: 230
AHB Address = 48, AXI Address = 48 at time: 230
Write Transfer Passed. AHB Data = 8, AXI Data = 8 at time: 250
AHB Address = 52, AXI Address = 52 at time: 250
Write Transfer Passed. AHB Data = 12, AXI Data = 12 at time: 270
AHB Address = 56, AXI Address = 56 at time: 270
Write Transfer Passed. AHB Data = 16, AXI Data = 16 at time: 290
AHB Address = 60, AXI Address = 60 at time: 290
Write Transfer Passed. AHB Data = 20, AXI Data = 20 at time: 310
AHB Address = 0, AXI Address = 0 at time: 310
Write Transfer Passed. AHB Data = 24, AXI Data = 24 at time: 330
AHB Address = 4, AXI Address = 4 at time: 330
Write Transfer Passed. AHB Data = 28, AXI Data = 28 at time: 350
AHB Address = 8, AXI Address = 8 at time: 350
Write Transfer Passed. AHB Data = 32, AXI Data = 32 at time: 370
AHB Address = 12, AXI Address = 12 at time: 450
Write Transfer Passed. AHB Data = 36, AXI Data = 36 at time: 470
AHB Address = 16, AXI Address = 16 at time: 470
Write Transfer Passed. AHB Data = 40, AXI Data = 40 at time: 490
AHB Address = 20, AXI Address = 20 at time: 490
Write Transfer Passed. AHB Data = 44, AXI Data = 44 at time: 510
AHB Address = 24, AXI Address = 24 at time: 510
Write Transfer Passed. AHB Data = 48, AXI Data = 48 at time: 530
AHB Address = 28, AXI Address = 28 at time: 530
Write Transfer Passed. AHB Data = 52, AXI Data = 52 at time: 550
AHB Address = 32, AXI Address = 32 at time: 550
Write Transfer Passed. AHB Data = 56, AXI Data = 56 at time: 570
AHB Address = 36, AXI Address = 36 at time: 570
Write Transfer Passed. AHB Data = 60, AXI Data = 60 at time: 590
AHB Address = 40, AXI Address = 40 at time: 590
Write Transfer Passed. AHB Data = 64, AXI Data = 64 at time: 610
Burst Wrap Write Transaction with changing HREADY Test Passed.

Simulation Complete.
```

6. Read Transfers

- **Read response with okay of fixed and of length 9**

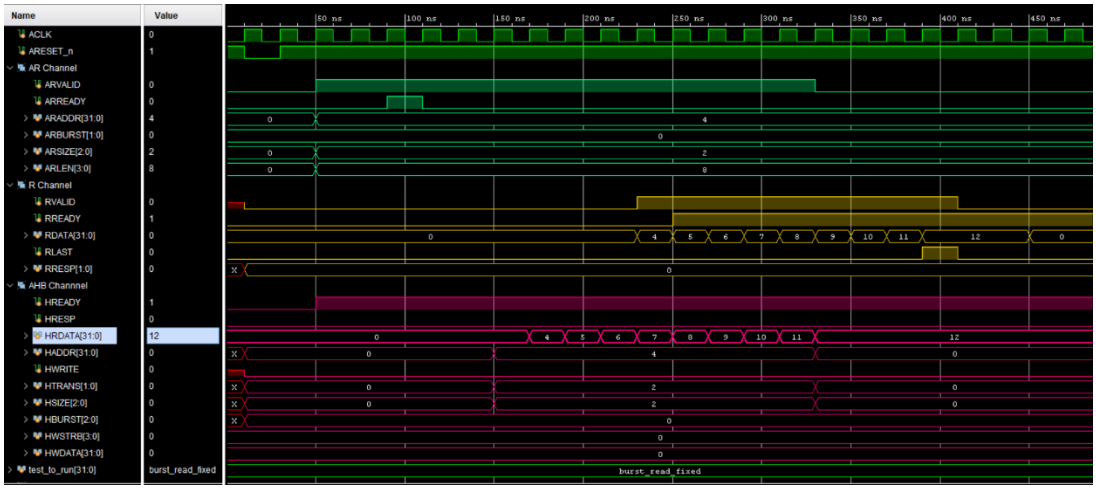

Figure 31: Read response with okay of fixed and of length 9

```
Running Fixed Burst Read Transaction Test
Read Transfer Passed. AHB Data = 4, AXI Data = 4 at time = 250
Read Transfer Passed. AHB Data = 5, AXI Data = 5 at time = 270
Read Transfer Passed. AHB Data = 6, AXI Data = 6 at time = 290
Read Transfer Passed. AHB Data = 7, AXI Data = 7 at time = 310
Read Transfer Passed. AHB Data = 8, AXI Data = 8 at time = 330
Read Transfer Passed. AHB Data = 9, AXI Data = 9 at time = 350
Read Transfer Passed. AHB Data = 10, AXI Data = 10 at time = 370
Read Transfer Passed. AHB Data = 11, AXI Data = 11 at time = 390
Read Transfer Passed. AHB Data = 12, AXI Data = 12 at time = 410
Burst Fixed Read Transaction Test Passed.

Simulation Complete.
```

- **Read response with error of fixed and length 15**

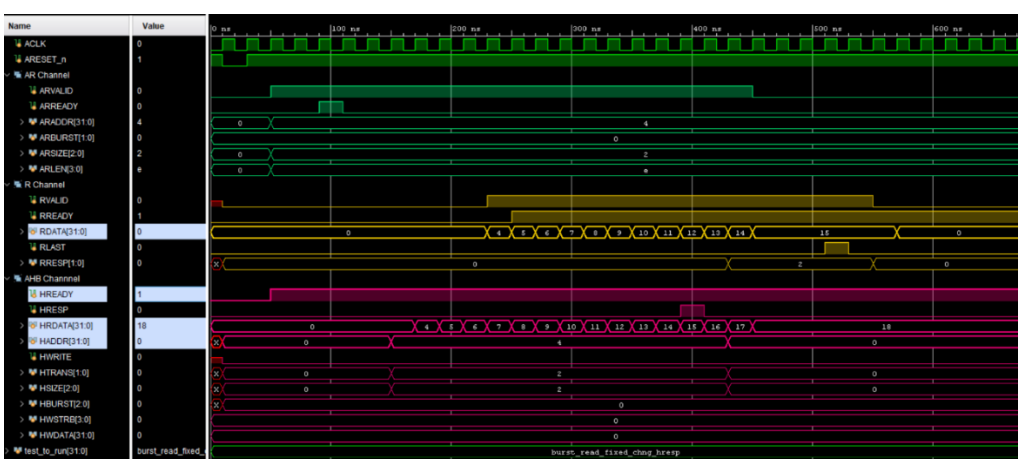

Figure 32: Read response with error of fixed and length 15

```
Running Fixed Read Transaction with changing HRESP Test
Read Transfer Passed. AHB Data = 4, AXI Data = 4 at time = 250
Read Transfer Passed. AHB Data = 5, AXI Data = 5 at time = 270
Read Transfer Passed. AHB Data = 6, AXI Data = 6 at time = 290
Read Transfer Passed. AHB Data = 7, AXI Data = 7 at time = 310
Read Transfer Passed. AHB Data = 8, AXI Data = 8 at time = 330
Read Transfer Passed. AHB Data = 9, AXI Data = 9 at time = 350
Read Transfer Passed. AHB Data = 10, AXI Data = 10 at time = 370
Read Transfer Passed. AHB Data = 11, AXI Data = 11 at time = 390
Read Transfer Passed. AHB Data = 12, AXI Data = 12 at time = 410
Read Transfer Passed. AHB Data = 13, AXI Data = 13 at time = 430
Read Transfer Passed. AHB Data = 14, AXI Data = 14 at time = 450
HTRANS Passed. Expected = 0, Actual = 0 at time: 470
Burst Fixed Read Transaction Test With Changing HRESP  Passed.

Simulation Complete.
```

- **Read response with okay and fixed of length 15**

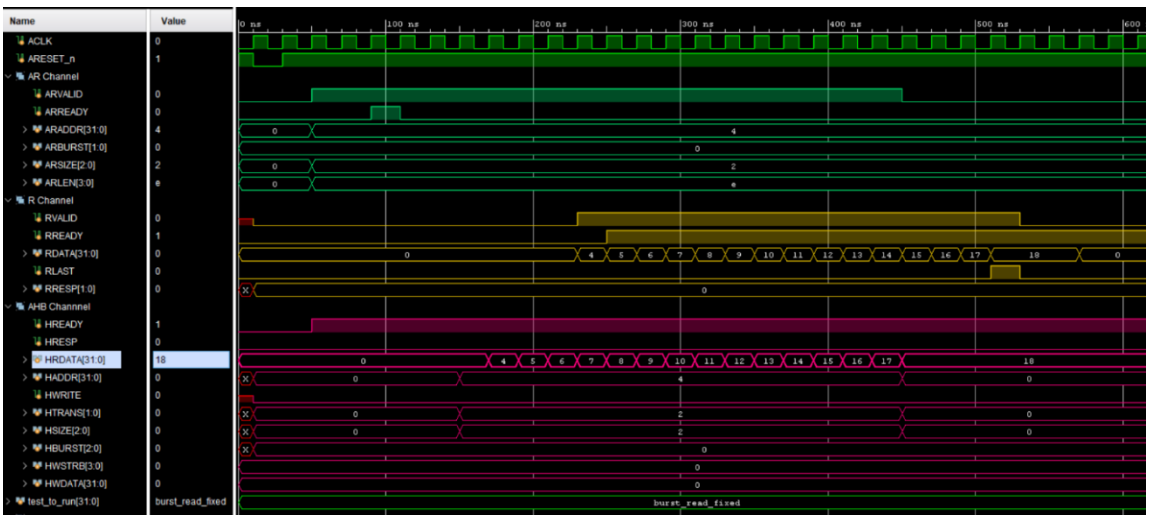

Figure 33: Read response with okay and fixed of length 15

```
Running Fixed Burst Read Transaction Test
Read Transfer Passed. AHB Data = 4, AXI Data = 4 at time = 250
Read Transfer Passed. AHB Data = 5, AXI Data = 5 at time = 270
Read Transfer Passed. AHB Data = 6, AXI Data = 6 at time = 290
Read Transfer Passed. AHB Data = 7, AXI Data = 7 at time = 310
Read Transfer Passed. AHB Data = 8, AXI Data = 8 at time = 330
Read Transfer Passed. AHB Data = 9, AXI Data = 9 at time = 350
Read Transfer Passed. AHB Data = 10, AXI Data = 10 at time = 370
Read Transfer Passed. AHB Data = 11, AXI Data = 11 at time = 390
Read Transfer Passed. AHB Data = 12, AXI Data = 12 at time = 410
Read Transfer Passed. AHB Data = 13, AXI Data = 13 at time = 430
Read Transfer Passed. AHB Data = 14, AXI Data = 14 at time = 450
Read Transfer Passed. AHB Data = 15, AXI Data = 15 at time = 470
Read Transfer Passed. AHB Data = 16, AXI Data = 16 at time = 490
Read Transfer Passed. AHB Data = 17, AXI Data = 17 at time = 510
Read Transfer Passed. AHB Data = 18, AXI Data = 18 at time = 530
Burst Fixed Read Transaction Test Passed.

Simulation Complete.
```

- **Read response with okay and with increment of length 15**

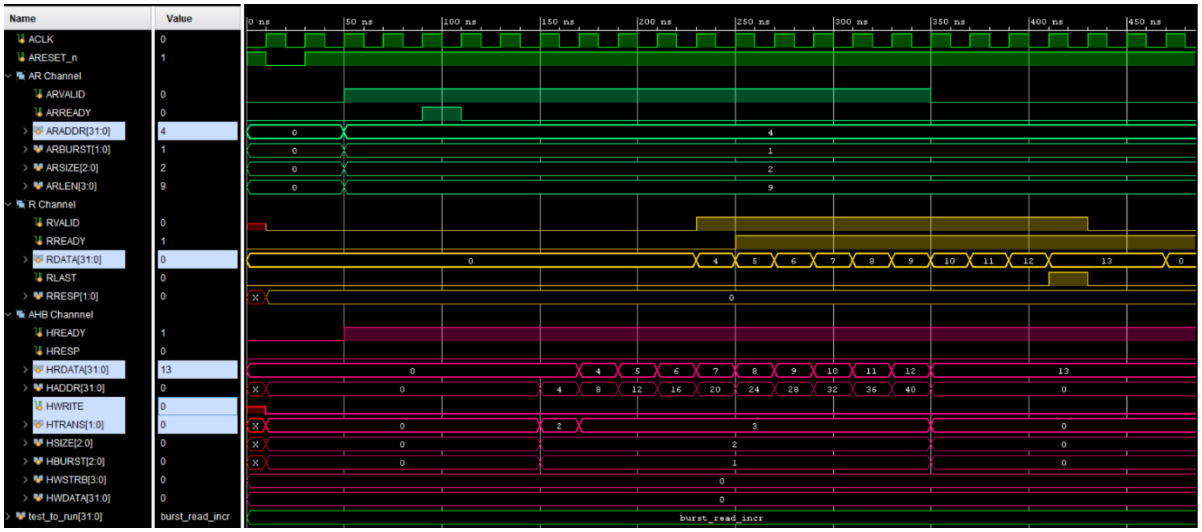

Figure 34: Read response with okay and with increment of length 15

```
Running Increment Read Transaction Test
Read Transfer Passed. AHB Data = 4, AXI Data = 4 at time = 250
Read Transfer Passed. AHB Data = 5, AXI Data = 5 at time = 270
Read Transfer Passed. AHB Data = 6, AXI Data = 6 at time = 290
Read Transfer Passed. AHB Data = 7, AXI Data = 7 at time = 310
Read Transfer Passed. AHB Data = 8, AXI Data = 8 at time = 330
Read Transfer Passed. AHB Data = 9, AXI Data = 9 at time = 350
Read Transfer Passed. AHB Data = 10, AXI Data = 10 at time = 370
Read Transfer Passed. AHB Data = 11, AXI Data = 11 at time = 390
Read Transfer Passed. AHB Data = 12, AXI Data = 12 at time = 410
Read Transfer Passed. AHB Data = 13, AXI Data = 13 at time = 430
Burst Increment Read Transaction Test Passed.

Simulation Complete.
```

- **Read response with okay and with increment of length 14**

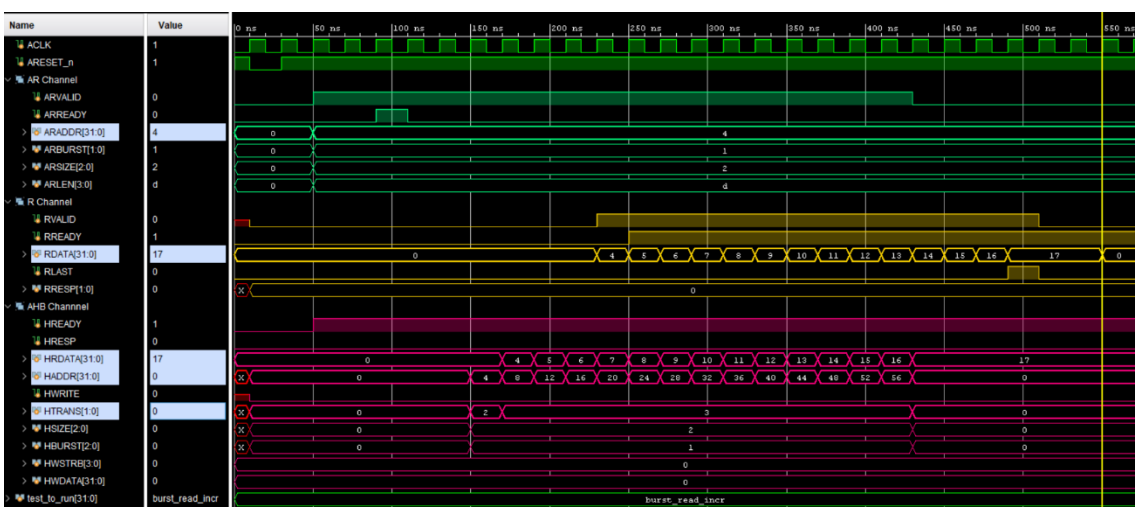

Figure 35: Read response with okay and with increment of length 14

```
Running Increment Read Transaction Test
Read Transfer Passed. AHB Data = 4, AXI Data = 4 at time = 250
Read Transfer Passed. AHB Data = 5, AXI Data = 5 at time = 270
Read Transfer Passed. AHB Data = 6, AXI Data = 6 at time = 290
Read Transfer Passed. AHB Data = 7, AXI Data = 7 at time = 310
Read Transfer Passed. AHB Data = 8, AXI Data = 8 at time = 330
Read Transfer Passed. AHB Data = 9, AXI Data = 9 at time = 350
Read Transfer Passed. AHB Data = 10, AXI Data = 10 at time = 370
Read Transfer Passed. AHB Data = 11, AXI Data = 11 at time = 390
Read Transfer Passed. AHB Data = 12, AXI Data = 12 at time = 410
Read Transfer Passed. AHB Data = 13, AXI Data = 13 at time = 430
Read Transfer Passed. AHB Data = 14, AXI Data = 14 at time = 450
Read Transfer Passed. AHB Data = 15, AXI Data = 15 at time = 470
Read Transfer Passed. AHB Data = 16, AXI Data = 16 at time = 490
Read Transfer Passed. AHB Data = 17, AXI Data = 17 at time = 510
Burst Increment Read Transaction Test Passed.

Simulation Complete.
```

- **Read response with okay and with wrap of length 16**

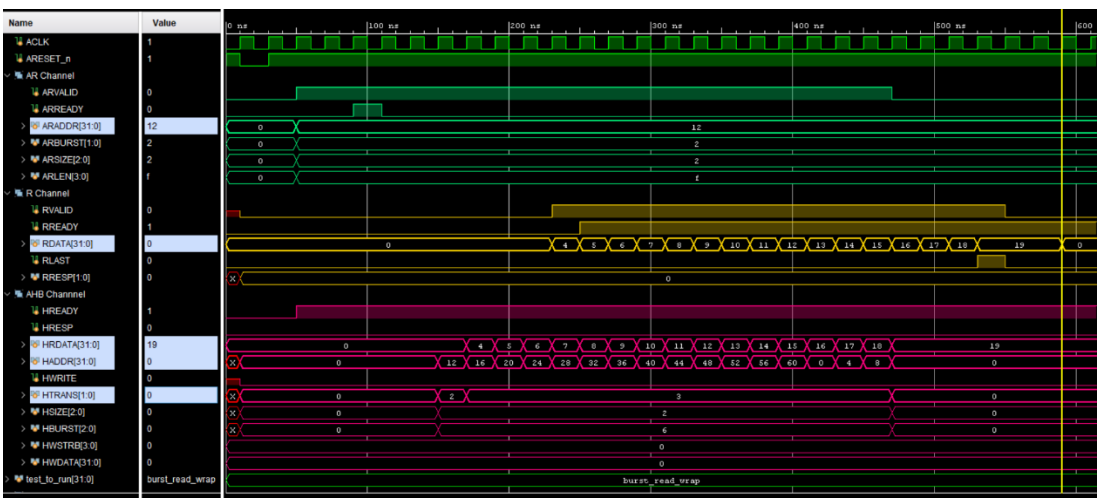

Figure 36: Read response with okay and with wrap of length 16

```
Running Wrap Read Transaction Test
Read Transfer Passed. AHB Data = 4, AXI Data = 4 at time = 250
Read Transfer Passed. AHB Data = 5, AXI Data = 5 at time = 270
Read Transfer Passed. AHB Data = 6, AXI Data = 6 at time = 290
Read Transfer Passed. AHB Data = 7, AXI Data = 7 at time = 310
Read Transfer Passed. AHB Data = 8, AXI Data = 8 at time = 330
Read Transfer Passed. AHB Data = 9, AXI Data = 9 at time = 350
Read Transfer Passed. AHB Data = 10, AXI Data = 10 at time = 370
Read Transfer Passed. AHB Data = 11, AXI Data = 11 at time = 390
Read Transfer Passed. AHB Data = 12, AXI Data = 12 at time = 410
Read Transfer Passed. AHB Data = 13, AXI Data = 13 at time = 430
Read Transfer Passed. AHB Data = 14, AXI Data = 14 at time = 450
Read Transfer Passed. AHB Data = 15, AXI Data = 15 at time = 470
Read Transfer Passed. AHB Data = 16, AXI Data = 16 at time = 490
Read Transfer Passed. AHB Data = 17, AXI Data = 17 at time = 510
Read Transfer Passed. AHB Data = 18, AXI Data = 18 at time = 530
Read Transfer Passed. AHB Data = 19, AXI Data = 19 at time = 550
Burst Wrap Read Transaction Test Passed.

Simulation Complete.
```

- **Read response with okay and with wrap wait state**

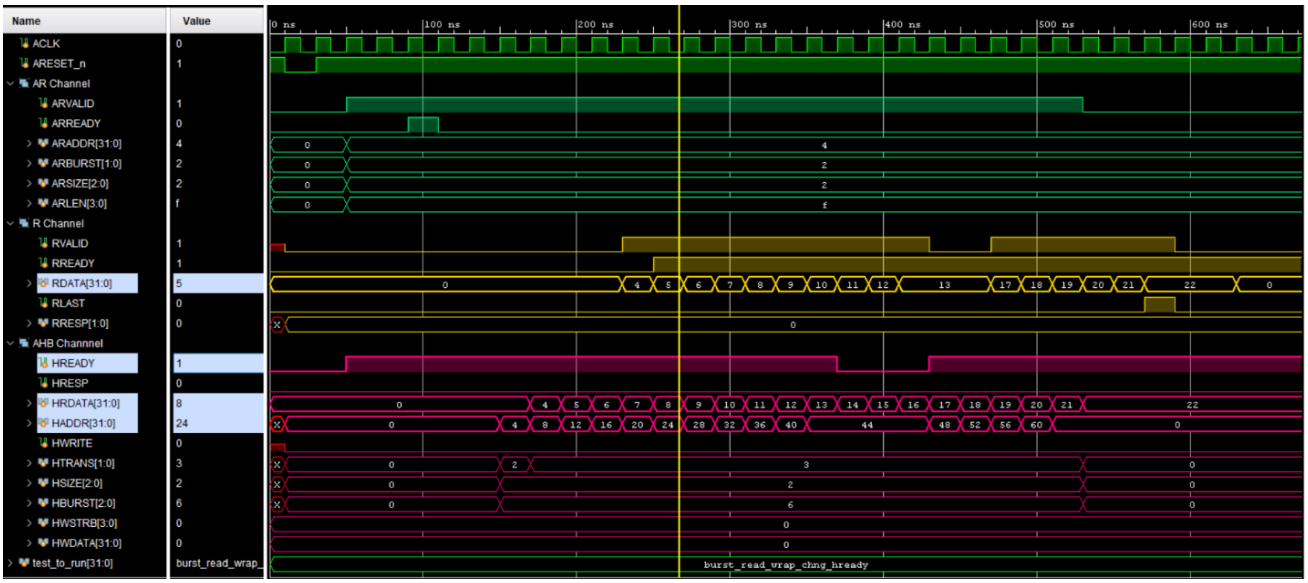

Figure 37: Read response with okay and with wrap wait state

```
Running Wrap Read Transaction Test With Changing HREADY
Read Transfer Passed. AHB Data = 4, AXI Data = 4 at time = 250
Read Transfer Passed. AHB Data = 5, AXI Data = 5 at time = 270
Read Transfer Passed. AHB Data = 6, AXI Data = 6 at time = 290
Read Transfer Passed. AHB Data = 7, AXI Data = 7 at time = 310
Read Transfer Passed. AHB Data = 8, AXI Data = 8 at time = 330
Read Transfer Passed. AHB Data = 9, AXI Data = 9 at time = 350
Read Transfer Passed. AHB Data = 10, AXI Data = 10 at time = 370
Read Transfer Passed. AHB Data = 11, AXI Data = 11 at time = 390
Read Transfer Passed. AHB Data = 12, AXI Data = 12 at time = 410
Read Transfer Passed. AHB Data = 13, AXI Data = 13 at time = 430
Read Transfer Passed. AHB Data = 17, AXI Data = 17 at time = 490
Read Transfer Passed. AHB Data = 18, AXI Data = 18 at time = 510
Read Transfer Passed. AHB Data = 19, AXI Data = 19 at time = 530
Read Transfer Passed. AHB Data = 20, AXI Data = 20 at time = 550
Read Transfer Passed. AHB Data = 21, AXI Data = 21 at time = 570
Read Transfer Passed. AHB Data = 22, AXI Data = 22 at time = 590
Burst Wrap Read Transaction Test With Changing HREADY Passed.

Simulation Complete.
```

Implementation results

1. **RTL design:** Detailed RTL schematic verifies bridge logic.

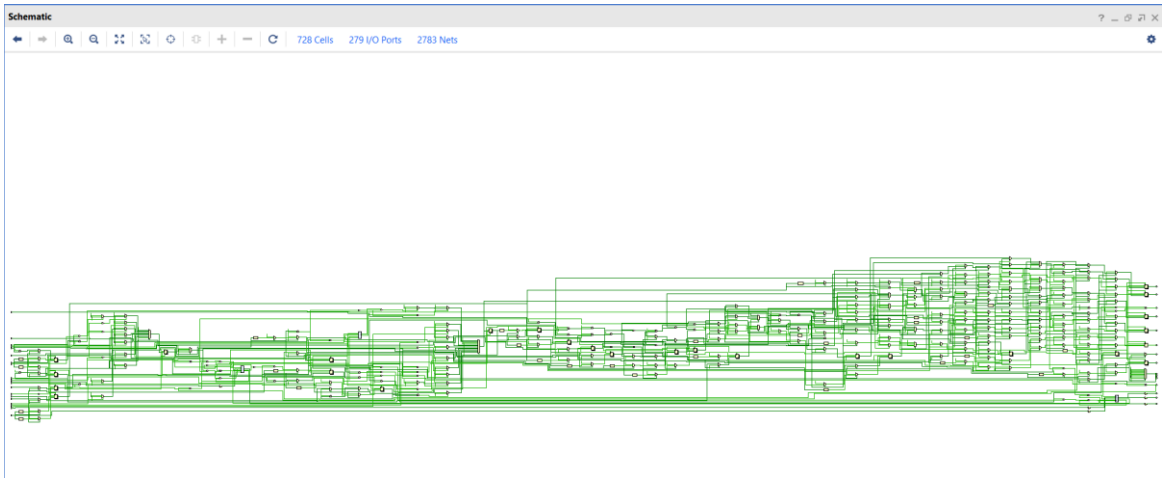

Figure 38: RTL design

2. **Synthesize design**: Design successfully synthesized using industry-standard tool (Vivado 2019.1).

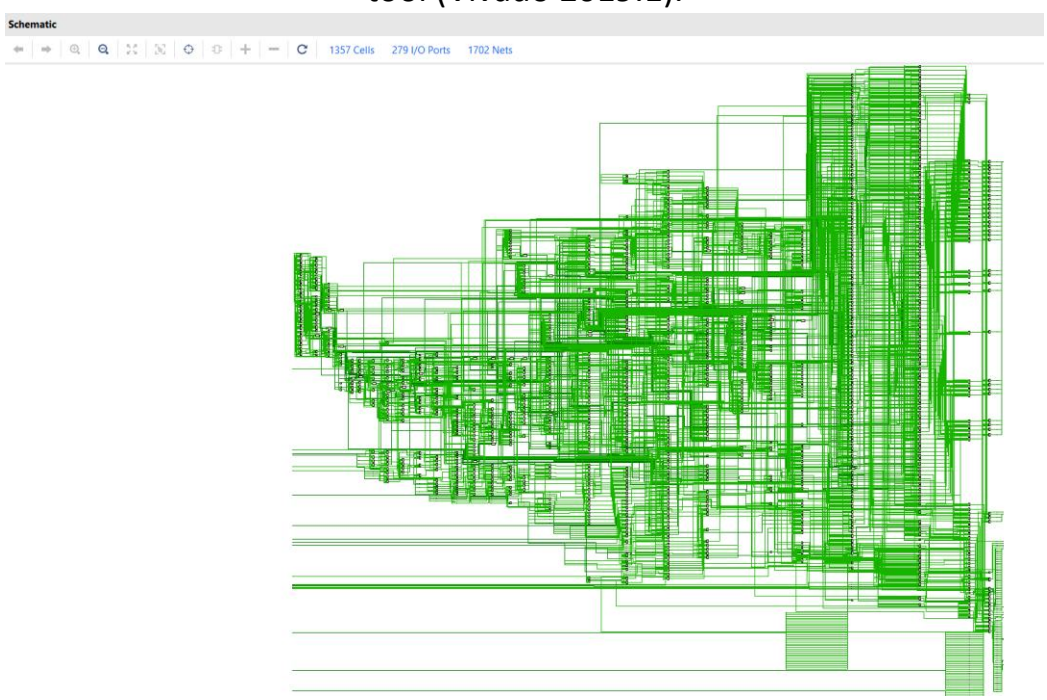

Figure 39: Synthesize design

3. **Implemented design:** Implementation demonstrates functional correctness post-synthesis of the AIX to AHB Bridge design.

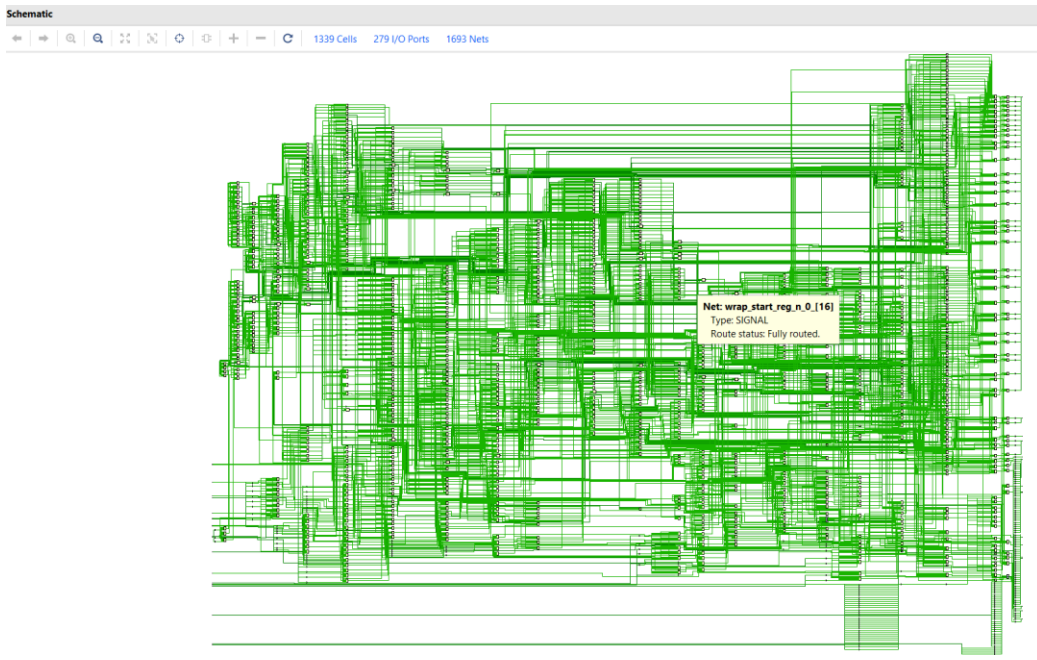

Figure 40: Implemented design

4. Power Reports

- Power hierarchy report:

Utilization	Name	Signals (W)	Data (W)	Clock Enable (W)	Set/Reset (W)	Logic (W)	BRAM (W)	I/O (W)
2.405 W (97% of total)	AXI_to_AHB_Bridge							
2.197 W (88% of total)	Leaf Cells (1356)							
0.09 W (4% of total)	write_data_fifo (data_fifo)	0.004	0.004	<0.001	<0.001	0.004	0.082	<0.001
0.088 W (4% of total)	read_data_fifo (data_fifo)	0.003	0.003	<0.001	<0.001	0.004	0.081	<0.001
0.031 W (1% of total)	Arbitor_inst (Arbiter_Read_Write)	0.013	0.013	<0.001	<0.001	0.018	<0.001	<0.001

Figure 41: Power Hierarchy Report

- Power summary report:

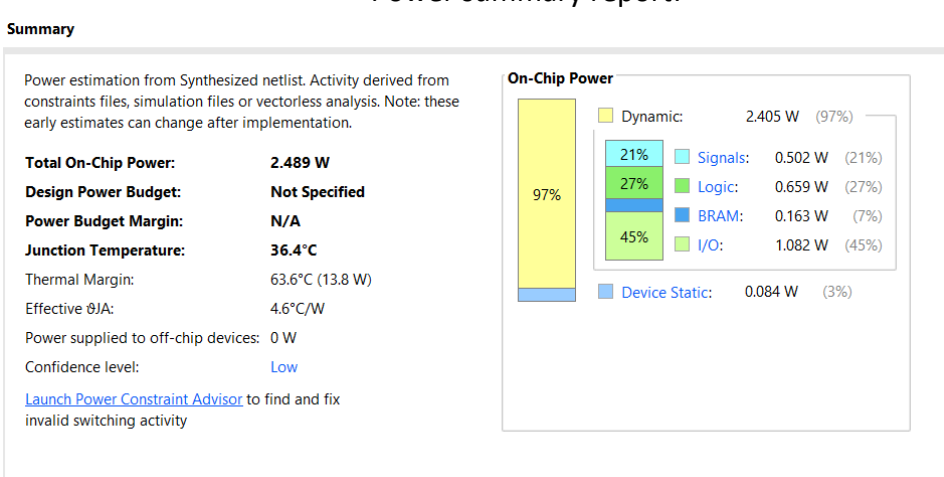

Figure 42: Power summary report

5. Utilization Reports

Utilization hierarchy report:

Utilization

Hierarchy

Name	Slice LUTs (63400)	Slice Registers (126800)	Block RAM Tile (135)	Bonded IOB (210)	BUFGCTRL (32)
AXI_to_AHB_Bridge	659	353	1	279	1
Arbitor_inst (Arbiter_Read_Write)	20	5	0	0	0
read_data_fifo (data_fifo)	25	28	0.5	0	0
write_data_fifo (data_fifo)	25	28	0.5	0	0

Figure 43: Utilization Hierarchy Report

Utilization summary report:

Utilization

Summary

Resource	Utilization	Available	Utilization %
LUT	659	63400	1.04
FF	353	126800	0.28
BRAM	1	135	0.74
IO	279	210	132.86

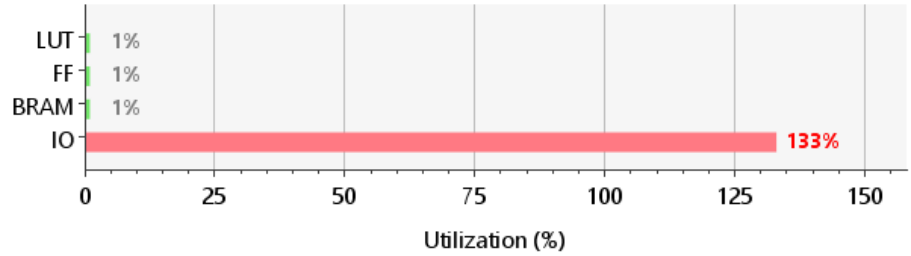

Figure 44: Utilization summary report

6. Timing summary:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.469 ns	Worst Hold Slack (WHS):	0.060 ns	Worst Pulse Width Slack (WPWS):	2.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	759	Total Number of Endpoints:	759	Total Number of Endpoints:	358

All user specified timing constraints are met.

Figure 45: Timing summary

Conclusion

The AXI to AHB Bridge plays an important role in ensuring data communication between AXI-based masters and AHB-based slaves within modern SoC architectures. By efficiently translating AXI's advanced multi-channel transactions into AHB's simplified pipelined operations, the bridge ensures protocol compatibility, data communication between heterogeneous systems, low latency, and optimized data transfer performance. The detailed simulations, including various test cases for single transfers, burst operations, and wait states, validate the bridge's functionality, reliability, and robustness under different scenarios. Furthermore, the implementation results, including power, utilization, and timing reports, demonstrate the design's efficiency and scalability. This bridge is a crucial solution for achieving high-performance and power-efficient interconnectivity in complex SoC environment.