AHB-to-APB Bridge: Design and Implementation

Project Report

Name : Vivek N Raj

Date : 13/08/2025

Contents

List of Figures

List of Tables

I. Objectives

The objective of this project is to design and simulate a synthesizable AHB to APB bridge interface using Verilog and run single read and single write tests using AHB Master and APB Slave testbenches. The bridge unit converts system bus transfers into APB transfers and performs the following functions:

- Latches the address and holds it valid throughout the transfer.
- Decodes the address and generates a peripheral select, PSELx. Only one select signal can be active during a transfer.
- Drives the data onto the APB for a write transfer.
- Drives the data onto the APB for a write transfer.
- Drives the APB data onto the system bus for a read transfer.
- Generates a timing strobe, PENABLE, for the transfer.
- Can implement single read and write operations successfully.

II. Implementation tools

- HDL Used : Verilog
- Simulator Tool Used: ModelSIM
- Synthesis Tool Used: Quartus Prime
- Family: Cyclone V
- Device: 5CSXFC6D6F31I7ES

III. Introduction

Before proceeding to our project, its necessary to know some basic concepts of our system and the components that make our system.

A. About the AMBA Bus

The Advanced Microcontroller Bus Architecture (AMBA) specification defines an on-chip communications standard for designing high-performance embedded microcontrollers. Three distinct buses are defined within the AMBA specification:

- Advanced High-performance Bus (AHB)
- Advanced System Bus (ASB)
- Advanced Peripheral Bus (APB).

B. Advanced High-performance Bus (AHB)

The AMBA ASB is for high-performance system modules. AMBA ASB is an alternative system bus suitable for use where the high-performance features of AHB are not required. ASB also supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macrocell functions.

C. Advanced System Bus (ASB)

The AMBA ASB is for high-performance system modules. AMBA ASB is an alternative system bus suitable for use where the high-performance features of AHB are not required. ASB also supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macrocell functions.

D. Advanced Peripheral Bus (APB)

The AMBA APB is for low-power peripherals. AMBA APB is optimized for minimal power consumption and reduced interface complexity to support peripheral functions. APB can be used in conjunction with either version of the system bus.

E. Typical AMBA based system

An AMBA-based microcontroller typically consists of a high-performance system backbone bus (AMBA AHB or AMBA ASB), able to sustain the external memory bandwidth, on which the CPU, on-chip memory and other Direct Memory Access (DMA) devices reside. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. Also located on the high-performance bus is a bridge to the lower bandwidth APB, where most of the peripheral devices in the system are located. AMBA APB provides the basic peripheral macrocell communications infrastructure as a secondary bus from the higher bandwidth pipelined main system bus. Such peripherals typically:

- have interfaces which are memory-mapped registers
- have no high-bandwidth interfaces
- are accessed under programmed control.

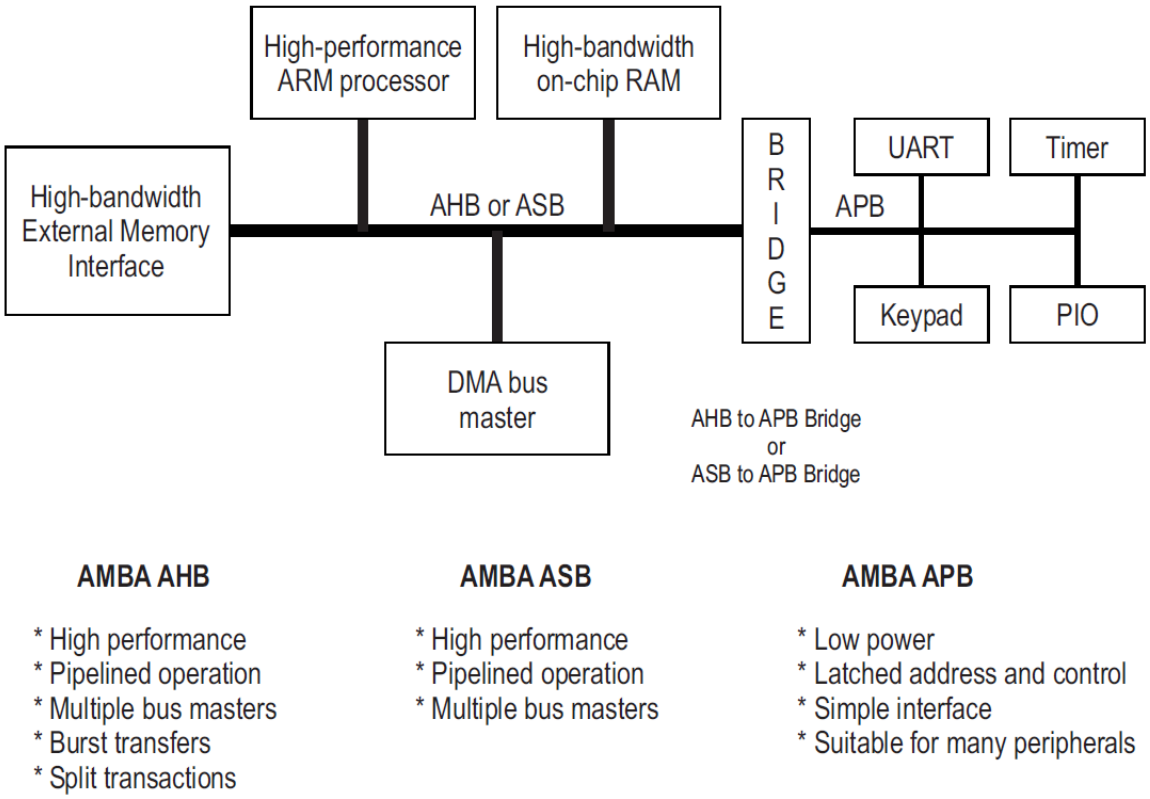

Fig. 1: AMBA system

IV. Basic Terminology

A. Bus cycle

A bus cycle is a basic unit of one bus clock period and for the purpose of AMBA AHB or APB protocol descriptions is defined from rising-edge to rising-edge transitions.

B. Bus transfer

An AMBA ASB or AHB bus transfer is a read or write operation of a data object, which may take one or more bus cycles. The bus transfer is terminated by a completion response from the addressed slave. An AMBA APB bus transfer is a read or write operation of a data object, which always requires two bus cycles.

C. Burst operation

A burst operation is defined as one or more data transactions, initiated by a bus master, which have a consistent width of transaction to an incremental region of address space. The increment step per transaction is determined by the width of transfer (byte, halfword, word). No burst operation is supported on the APB.

V. AMBA SIGNALS

All AMBA signals are named such that the first letter of the name indicates which bus the signal is associated with. A lower case n in the signal name indicates that the signal is active LOW, otherwise signal names are always all upper case. Test signals have a prefix T regardless of the bus type.

Prefix	**Description**
AHB Signal Prefixes	
H	Indicates an AHB signal. Example: `HREADY` indicates that the data portion of an AHB transfer can complete. It is active HIGH.
ASB Signal Prefixes	
A	Unidirectional signal between ASB bus masters and the arbiter.
B	Indicates an ASB signal. Example: `BnRES` is the ASB reset signal, active LOW.
D	Unidirectional ASB decoder signal.
APB Signal Prefixes	
P	Indicates an APB signal. Example: `PCLK` is the main clock used by the APB.

TABLE I: AMBA Bus Signal Prefixes

A. AMBA AHB Signals

- This section contains an overview of the AMBA AHB signals . All signals are prefixed with the letter H, ensuring that the AHB signals are differentiated from other similarly named signals in a system design.
- AMBA AHB also has a number of signals required to support multiple bus maste operation (see Table 2). Many of these arbitration signals are dedicated point to point links and in Table 2 the suffix x indicates the signal is from module X. For example

Name	Source	Description
HCLK	Clock source	This clock times all bus transfers. All signal timings are related to the rising edge of HCLK.
HRESETn	Reset controller	The bus reset signal is active LOW and is used to reset the system and the bus. This is the only active LOW signal.
HADDR[31:0]	Master	The 32-bit system address bus.
HTRANS[1:0]	Master	Indicates the type of the current transfer, which can be NONSEQUENTIAL, SEQUENTIAL, IDLE or BUSY.
HWRITE	Master	When HIGH this signal indicates a write transfer and when LOW a read transfer.
HSIZE[2:0]	Master	Indicates the size of the transfer, typically byte (8-bit), halfword (16-bit) or word (32-bit). Protocol allows up to 1024 bits.
HBURST[2:0]	Master	Indicates if the transfer forms part of a burst. Supports 4, 8, and 16 beat bursts, incrementing or wrapping.
HPROT[3:0]	Master	Optional protection control signals for access permissions.
HWDATA[31:0]	Master	Write data bus from master to slaves during write operations. Minimum 32-bit width recommended.
HSELx	Decoder	Each AHB slave has its own select signal. Indicates current transfer is for selected slave.
HRDATA[31:0]	Slave	Read data bus from slaves to master during read operations. Minimum 32-bit width recommended.
HREADY	Slave	HIGH indicates transfer finished. LOW extends transfer. Required as both input and output by slaves.
HRESP[1:0]	Slave	Transfer response: OKAY, ERROR, RETRY, SPLIT.

TABLE II: AMBA AHB Signals

there will be a number of HBUSREQx signals in a system, such as HBUSREQarm, HBUSREQdma and HBUSREQtic.

B. AMBA APB Signals

All AMBA APB signals use the single letter P prefix. Some APB signals, such as the clock, may be connected directly to the system bus equivalent signal. Table 3 shows the list

of AMBA APB signal names, along with a description of how each of the signals is used.

Name	**Source**	**Description**
PCLK	Clock source	Rising edge times all APB transfers.
PRESETn	Reset controller	Active LOW reset, usually connected to system bus reset.
PADDR[31:0]	Master	APB address bus, up to 32 bits, driven by bridge unit.
PSELx	Decoder	Select signal to each peripheral bus slave from secondary decoder in bridge unit.
PENABLE	Master	Strobe signal indicating second cycle of APB transfer. Rising edge occurs mid-transfer.
PWRITE	Master	HIGH for write access, LOW for read access.
PRDATA[31:0]	Slave	Read data bus driven by selected slave during read cycles. Up to 32 bits.
PWDATA[31:0]	Master	Write data bus driven by bridge unit during write cycles. Up to 32 bits.

TABLE III: AMBA APB Signals

VI. AHB to APB Bridge

The AHB to APB bridge is an AHB slave, providing an interface between the high-speed AHB and the low-power APB. Read and write transfers on the AHB are converted into equivalent transfers on the APB. As the APB is not pipelined, wait states are added during transfers to and from the APB when the AHB is required to wait for the APB.

The main sections of this module are:

- AHB slave bus interface
- APB transfer state machine, which is independent of the device memory map
- APB output signal generation.

The bridge unit converts system bus transfers into APB transfers and performs the following functions: • Latches the address and holds it valid throughout the transfer. • Decodes the address and generates a peripheral select, PSELx. Only one select signal can be active during a transfer. • Drives the data onto the APB for a write transfer. • Drives the APB data onto the system bus for a read transfer. • Generates a timing strobe, PENABLE, for the transfer.

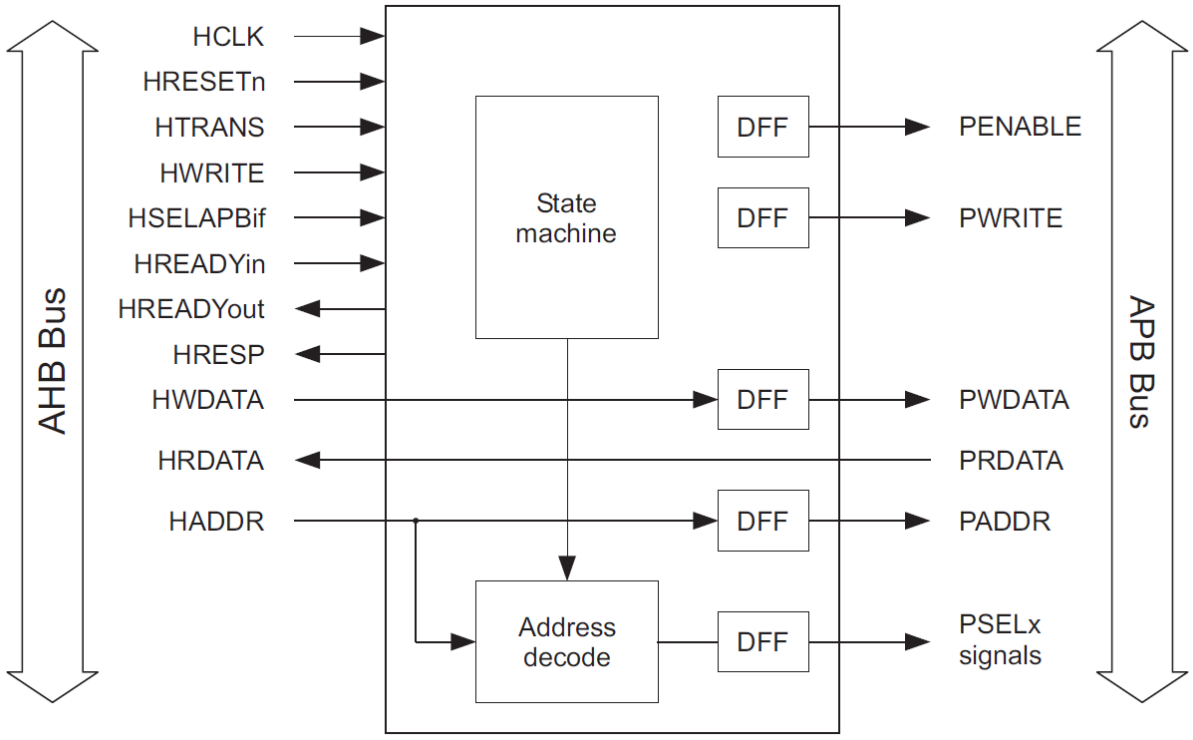

Fig. 2: Block diagram of Bridge module

VII. Design Modules

A. AHB Slave Interface

An AHB bus slave responds to transfers initiated by bus masters within the system. The slave uses a HSELx select signal from the decoder to determine when it should respond to a bus transfer. All other signals required for the transfer, such as the address and control information, will be generated by the bus master.

B. APB transfer state machine

The AHB to APB bridge comprises a state machine, which is used to control the generation of the APB and AHB output signals, and the address decoding logic which is used to generate the APB peripheral select lines. Hence it is also referred to as APB controller.

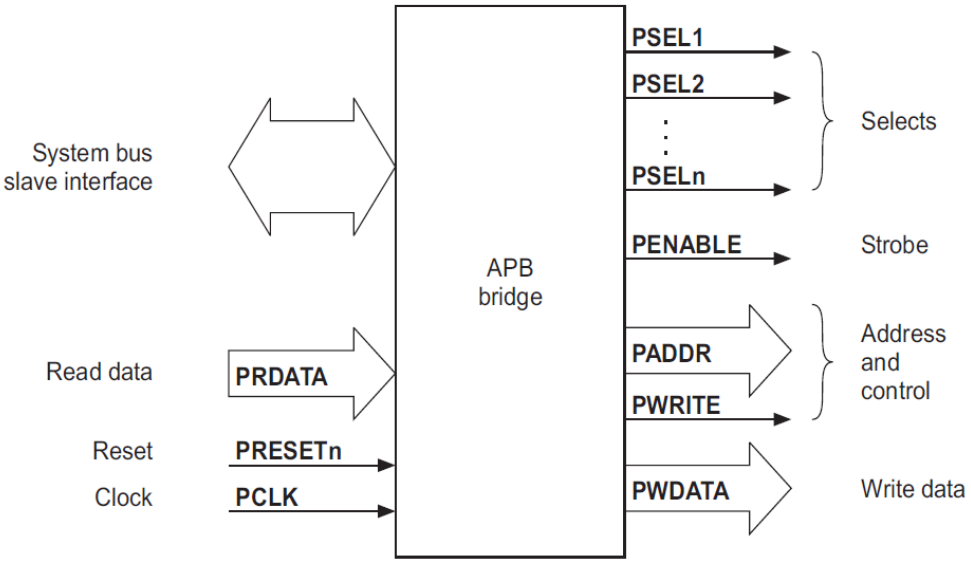

Fig. 3: AHB Slave Interface/APB bridge interface diagram

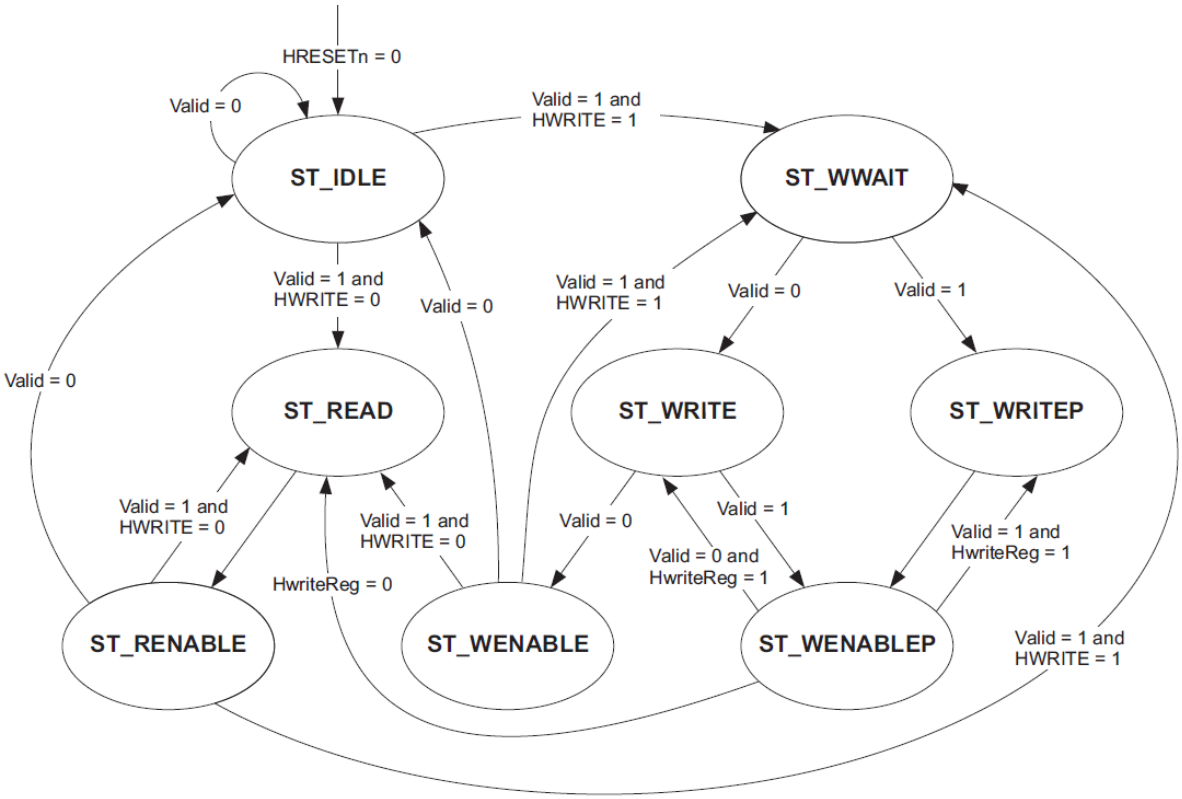

Fig. 4: State machine for AHB to APB interface

VIII. Implementation of the Project

This section describes how the HDL code for the APB bridge is set out. A simple system block diagram, with information about the main parts of the HDL code, is followed by details of the registers, inputs, and outputs used in the module. This should be read in conjunction with the HDL code.

The AHB to APB bridge comprises a state machine, which is used to control the generation of the APB and AHB output signals, and the address decoding logic which is used to generate the APB peripheral select lines. All registers used in the system are clocked from the rising edge of the system clock HCLK, and use the asynchronous reset HRESETn.

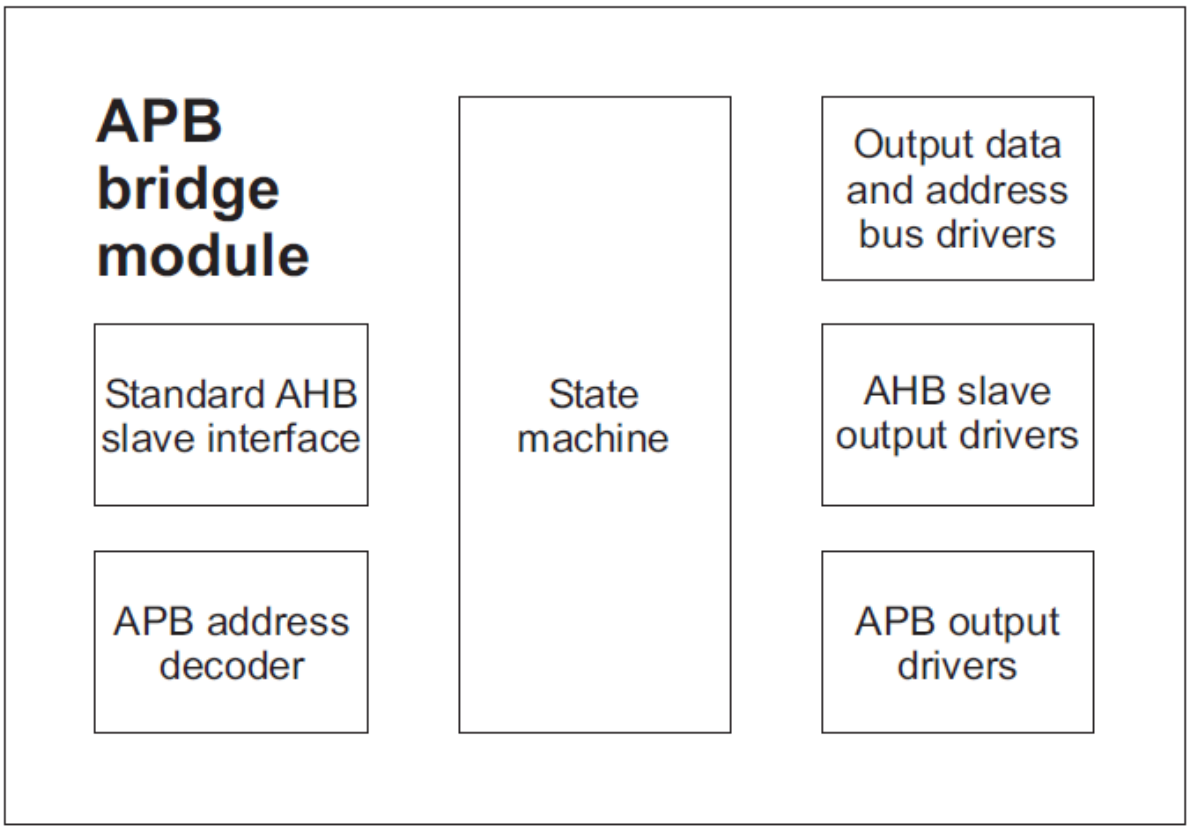

Fig. 5: APB bridge module block diagram

IX. Results

A. Simulation Results

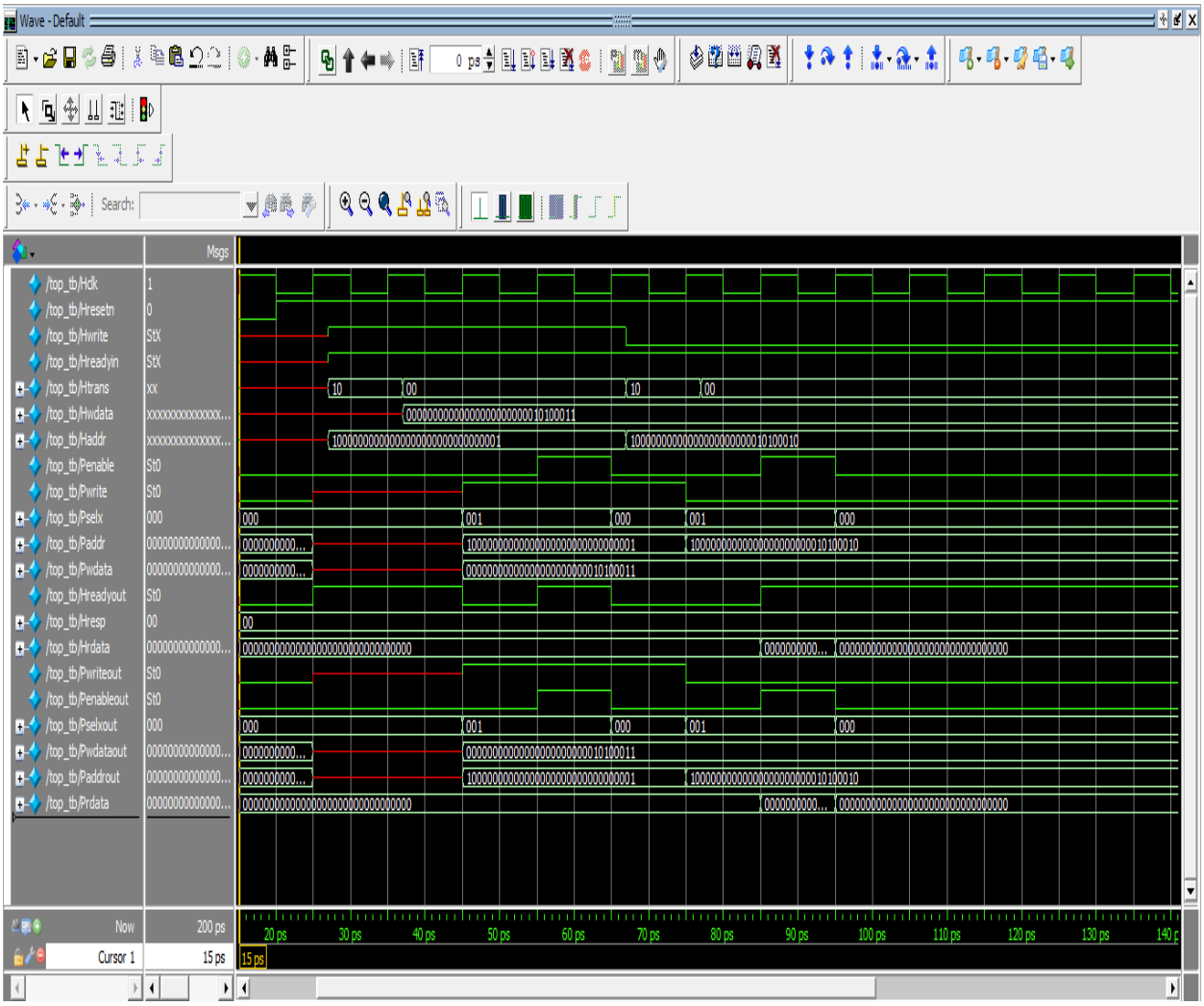

Fig. 6: Output waveforms

B. Synthesis Results

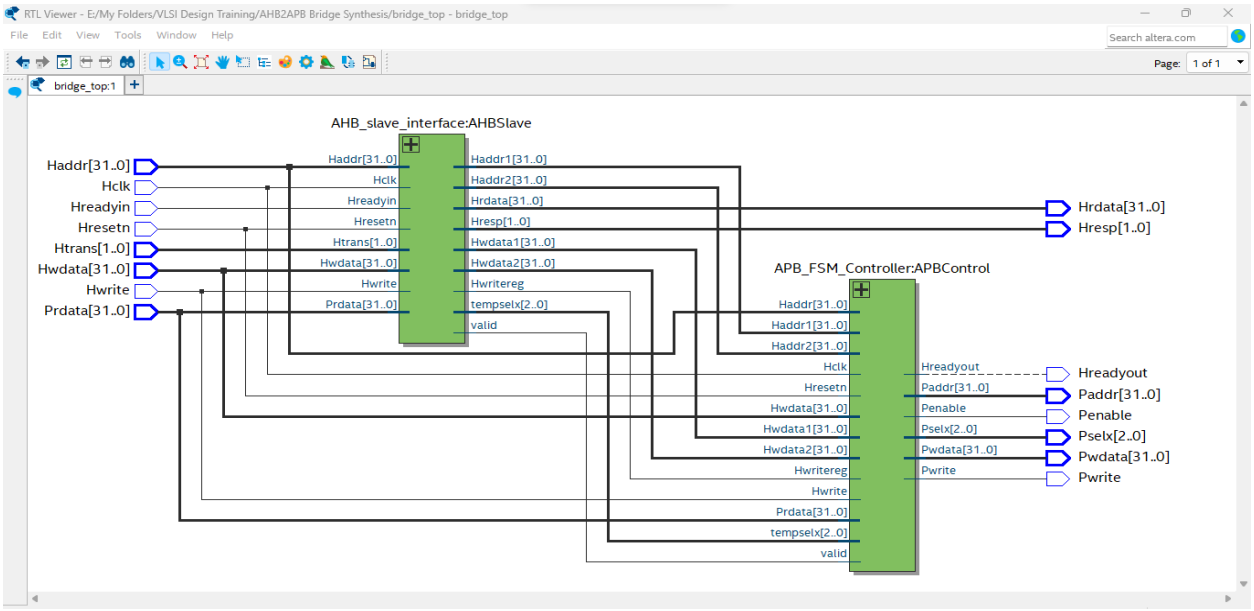

Fig. 7: RTL model

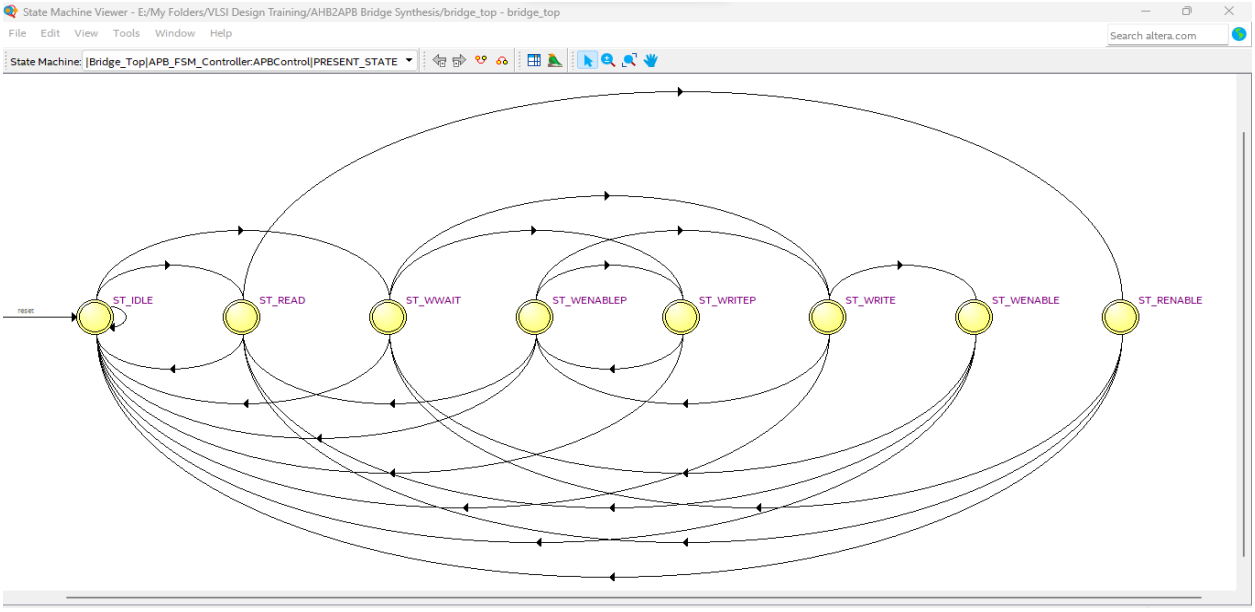

Fig. 8: Finite State Machine for the bridge

X. Conclusion

The AHB-to-APB bridge implemented in this project successfully enables communication between the high-speed AMBA AHB and low-power AMBA APB protocols, correctly translating read/write transactions and managing timing differences. The bridge module is synthesizable and ready for integration into SoCs, while the AHB master and APB slave used in testing were non-synthesizable simulation models for generating stimulus. Functional simulations confirm correct operation of the bridge FSM, address decoding, and data transfer logic. This modular and AMBA-compliant design is reusable and adaptable for connecting performance-oriented cores to APB-based peripherals in real-world systems.