Design & Verification of a Dual-port Memory with an APB subordinate (slave) interface

Nader Alnatsheh

February 3, 2026

Contents

1 Introduction

The purpose of this project is to build a reusable verification environments to be used for my personal projects in the future. The verification enviroments are based on the class - based Object-Oriented Programming (OOP) SystemVerilog (SV) methodology, and the Universal Verification Mehtodology (UVM).

Since the main focus is on building the verification environments, the design used in the project is a simple dual-port memory

2 Design Specs of the Dual-port Memory with an AMBA APB interfaces

2.1 Dual-port Memory Specs

2.1.1 Read Operation

After the rising edge of the PENABLE signal the PRDATA provides the data after 2 clock cycles. As soon as PREADY signal is high.

2.1.2 Write Operation

1. Addresses ranges from (0x0 - 0xf) are READ ONLY, and the manager (master) should not write to them, when the manager tries to write to a READ ONLY address, the subordinate (slave) raises the PSLVERR signal after 1 clock cycle (from when PENABLE rise to 1) to indicates an error.

2. For all other Addresses the write operation takes 4 clock cycles. And the memory contents updates on the 5th clock cycle.

3 Verification Specs of the Dual-port Memory with an AMBA APB interface

3.1 Test-plan

3.2 OOP SV Environment

3.2.1 Transaction (Transaction.sv)

3.2.2 Generator

3.2.3 Driver

3.2.4 Monitor

3.2.5 Scoreboard

3.2.6 Enviromnent

3.3 UVM Environment

3.3.1 apb_dpmem_transaction

3.3.2 apb_dpmem_sequence

3.3.3 apb_dpmem_sequencer

3.3.4 apb_dpmem_driver

3.3.5 apb_dpmem_monitor

3.3.6 apb_dpmem_agent

3.3.7 apb_dpmem_coverage

3.3.8 apb_dpmem_ref_model

3.3.9 apb_dpmem_scoreboard

3.3.10 apb_dpmem_environment

3.3.11 apb_dpmem_test

3.3.12 top

4 References