

ECE 554 Minilab1 Report

Simulation Testing and Verification

The top-level design was verified through simulation by allowing the testbench to assert the **start** signal, initiating the computation. The testbench then waits for the top-level state machine to reach the **DONE** state and executes a series of self-checking assertions that compare the matrix–vector multiplication results against the expected values.

In addition, the testbench validates the outputs driving the HEX digit signals, which are mapped to the 7-segment displays on the Altera DE1-SoC board. After all test cases pass, the same set of checks is repeated after asserting the **Clr** signal without resetting the system, confirming that the design operates correctly without requiring a full reset.

The following screen captures show the simulation results for designs implemented both **with** and **without** IP blocks (FIFO and MAC). The individual testbench and design *.sv files are available in the repository for reference.

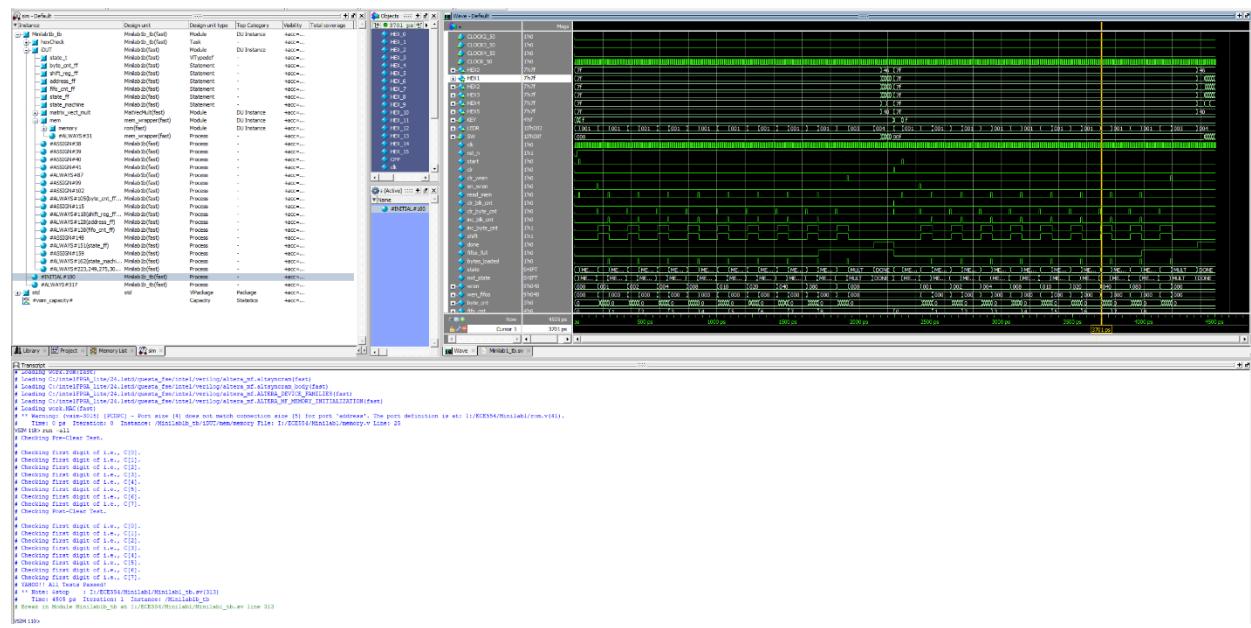


Figure 1. Snapshot showing the log output and waveforms without using IP blocks.

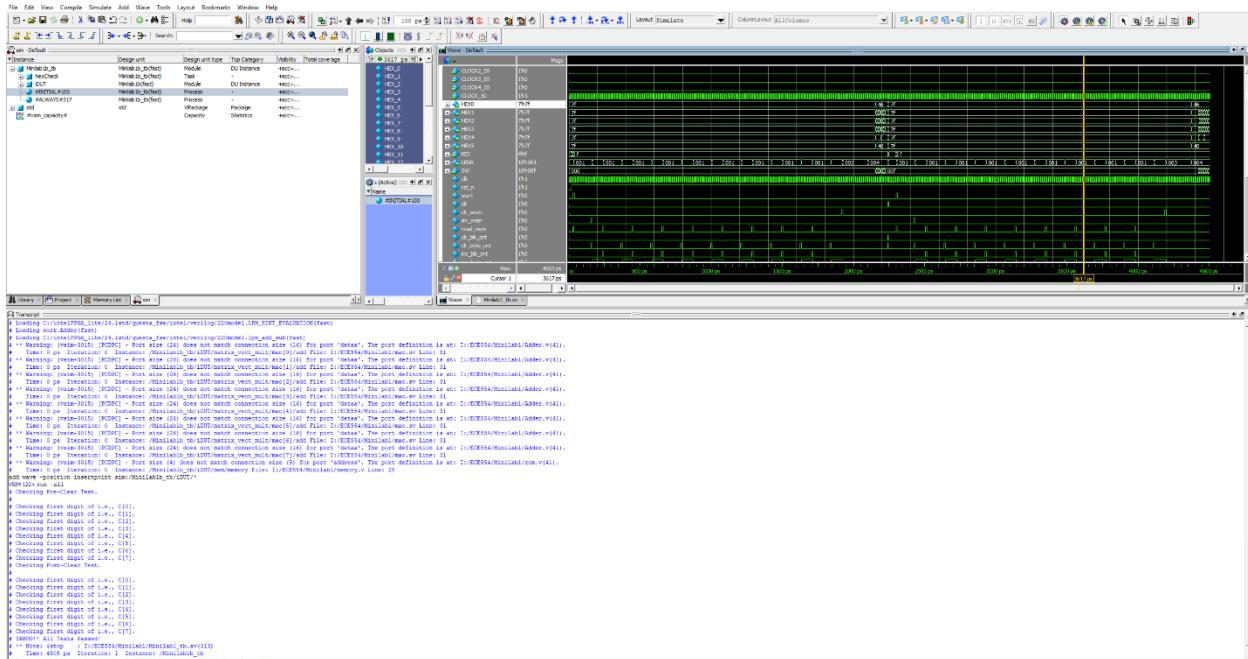


Figure 2. Snapshot showing the log output and waveforms using IP blocks.

Meeting the Timing Constraint

To meet the 200 MHz timing requirement, the design was modified by introducing an additional pipeline stage within the multiplier IP block. Timing analysis revealed that the critical path existed along the FIFO-to-MAC datapath, with the MAC unit contributing the majority of the delay. By pipelining the multiplier, the combinational delay on this path was reduced.

Because this change introduced an additional cycle of latency, the MAC enable signal was correspondingly pipelined to ensure that accumulation occurred during the correct clock cycle. With these modifications, the design exceeded the target clock frequency and achieved positive timing slack across all analyzed PVT corners.

On-Board Testing and Verification

The design was tested on the Altera DE1-SoC board by programming the FPGA with the generated bitstream using Quartus. The pushbuttons were mapped as follows: **KEY0** to `rst_n`, **KEY1** to start, and **KEY2** to `Clr`. For initial validation, the system was reset using **KEY0**, after which the start signal was asserted.

The current state of the top-level state machine was displayed on the lower three LEDs, with **LED2 asserted to indicate the DONE state**. Once the **DONE** state was observed, the output results were verified using the 7-segment displays. Each of the eight elements of the output vector **C** was mapped to the display starting at index 0. **SW3** enabled the display output, while **SW[2:0]** selected values 0–7 (in decimal) to cycle through the eight vector elements.

A photograph of the board demonstrating the correct operation and displayed results is shown below.

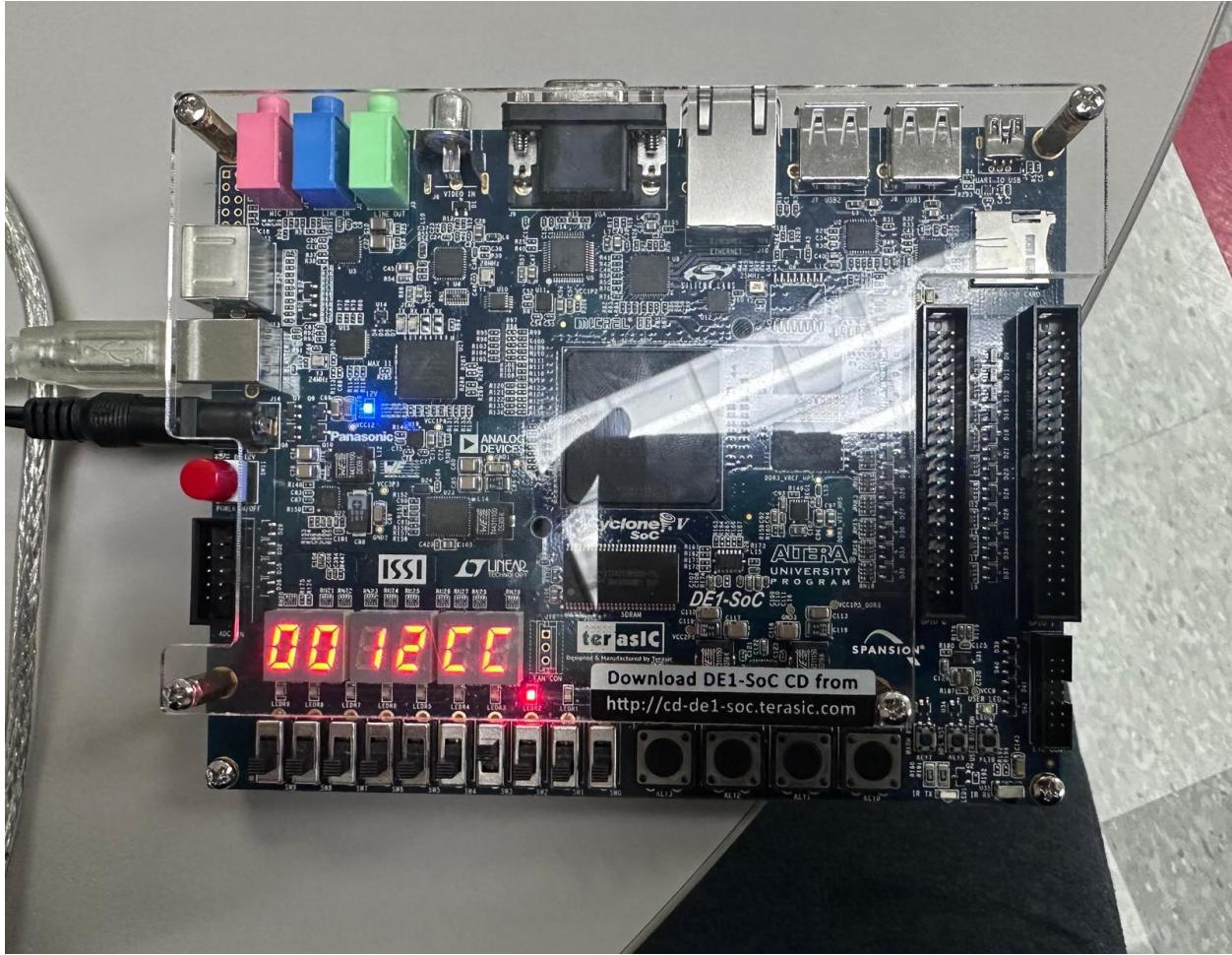


Figure 3. Snapshot showing manual testing of logic on the Alterra DE1-SoC Board

SignalTap Verification

The design was verified using **SignalTap** by capturing waveforms associated with the Avalon-MM slave interface. Signals monitored included address, readdata, read, readdatavalid, and waitrequest, allowing verification that memory transactions were occurring correctly and that the expected data values were returned on read operations.

A trigger condition was set on the address signal equal to **0x00000002**, with the waveform display centered around the trigger event. The captured waveform, shown in the screen capture below, confirms correct read behavior and proper timing relationships among the Avalon-MM signals.

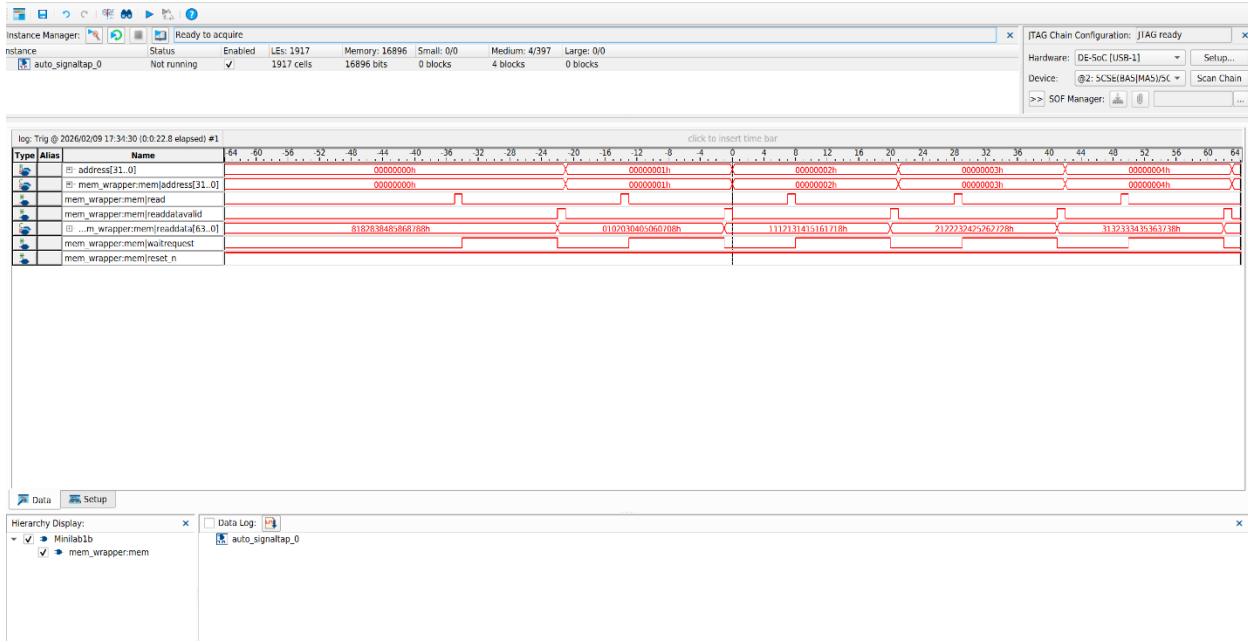


Figure 4. Snapshot showing the usage of SignalTap to debug the Avalon MM Slave Transaction

Development Challenges and Resolutions

Several challenges were encountered during the development process, primarily during the initial design phase due to functional bugs in the RTL. These issues were resolved through iterative debugging, including detailed code inspection, waveform analysis, and examination of simulation output messages, which ultimately led to correct functional behavior.

Additionally, synthesis and place-and-route times were longer than expected, likely due to the relatively intensive utilization of FPGA resources in the design. Despite this, the design successfully met timing requirements after optimization and completed implementation without further issues.