MINILAB 1 REPORT

Name: Harshul Jalan

Email: hjalan2@wisc.edu

Overview

This Minilab required the implementation of a top-level module for matrix-vector multiplication. The design reads input data from the provided memory module and performs row-wise multiply-accumulate operations to generate the final output vector.

Design

- The first 8 rows of memory were considered as eight separate FIFOs corresponding to matrix A. The final row of memory was considered as a single FIFO corresponding to vector B.

- To compute the matrix-vector product:

  - Eight MAC (Multiply-Accumulate) units were instantiated — one for each row of matrix A.

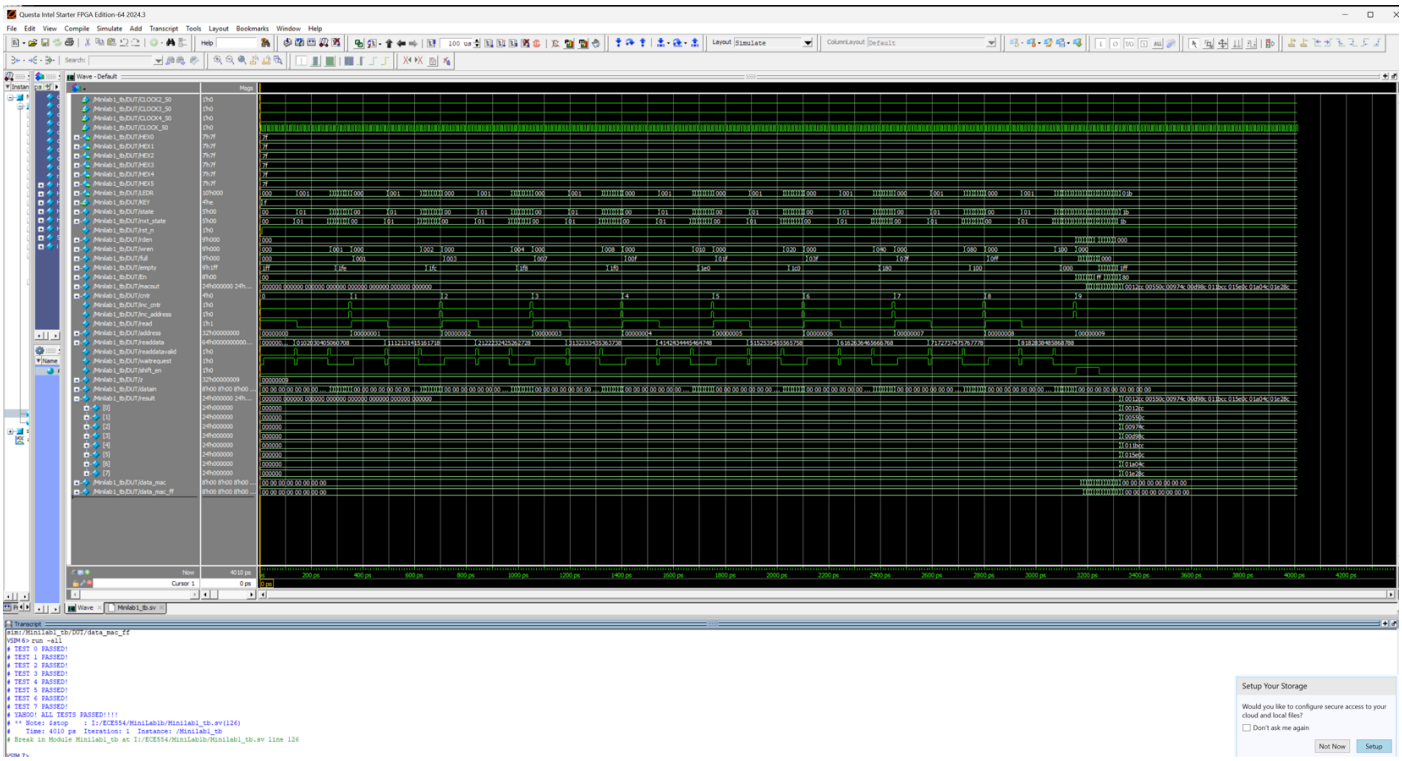
  - The enable signals for the MAC units were activated sequentially.

  - After multiplying a00 and b00, the value of b00 advanced to the next row multiplication, ensuring proper reuse of the vector element across all rows.

  - Existing FIFO and MAC IPs from Quartus were used to implement these to ensure smooth implementation and correctness

Screenshots and Images

- Waveform and Simulation Log



- Timing Analyzer before

Slow 1100mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	51.55 MHz	51.55 MHz	altera...ed_tck	
2	127.47 MHz	127.47 MHz	CLOCK_50	

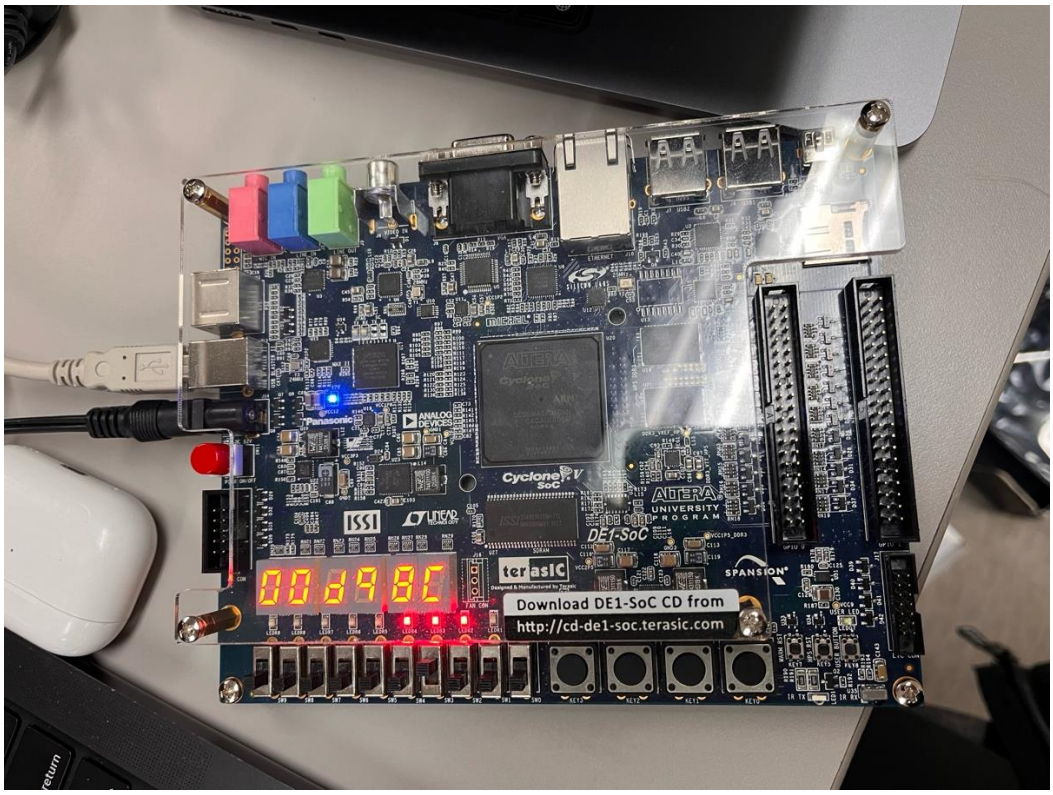
- Timing Analyzer after

Slow 1100mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	51.55 MHz	51.55 MHz	altera...ed_tck	
2	267.09 MHz	267.09 MHz	CLOCK_50	

- Working Board at SW3



How did we fix timing?

We initially had a frequency of 127.47 with the timing constraints of 200 MHz not satisfied. To fix this:

- We flopped the outputs of the FIFO to delay them by a cycle which required me to modify the state machine and delay the calculations by a cycle.

- We modified the IP of the LPM_MULT module to be pipelined by 2 cycles which required me to account for this in the state machine as well.

These changes helped us achieve a final frequency of 267.09 MHz. The screenshots for these before and after the fix are also in the images folder.