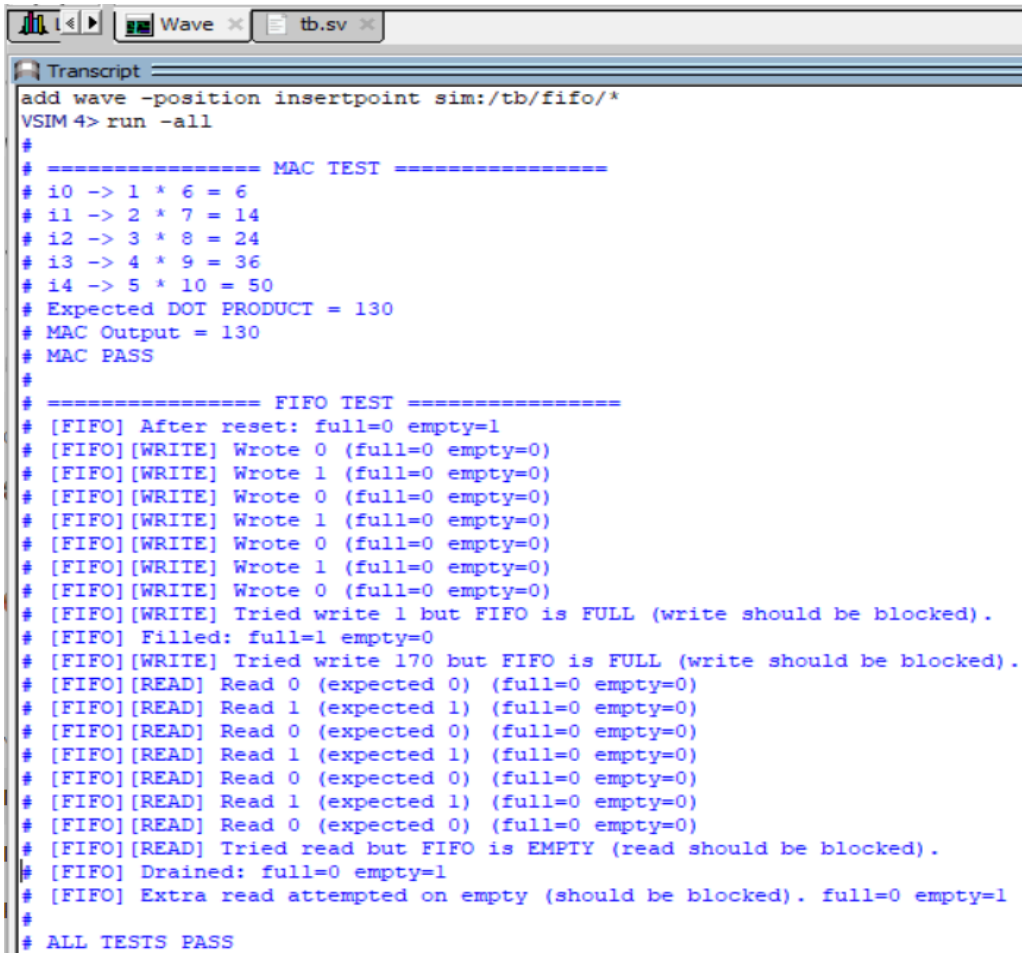
Git Repository: https://github.com/chancehowarth33/ECE554SP26_MiniLab1

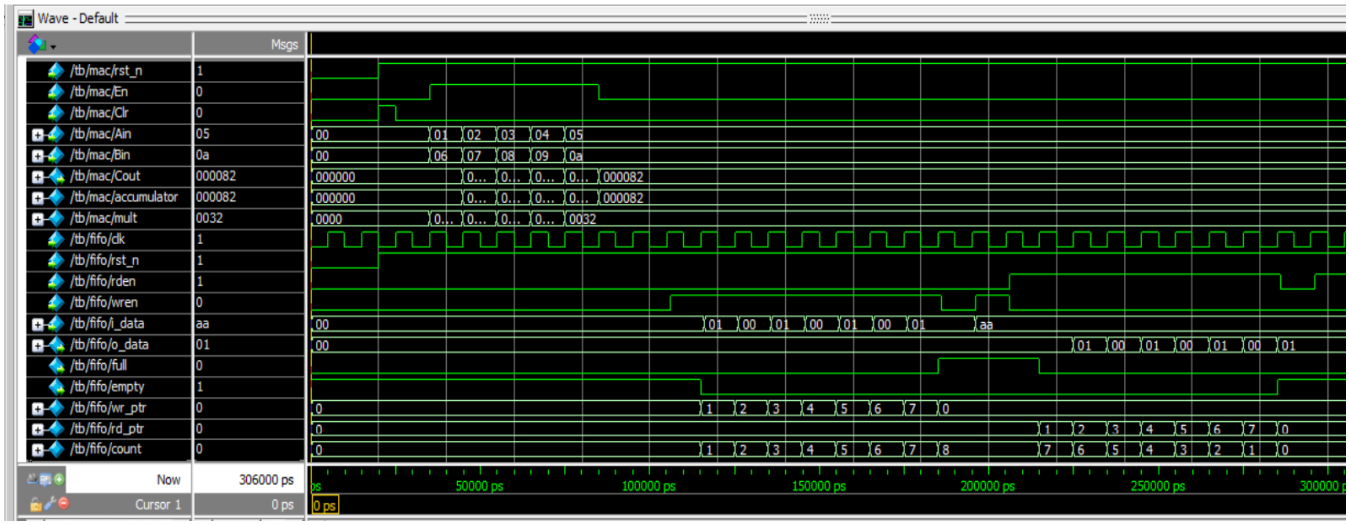
This Github repository is split into 2 parts. The repository contains two directories named MiniLab1-Part_A and MiniLab1-Part_B. Each repository contains the respective files we have worked on for those labs.

To create and populate this repository, we used the following Git commands:
1. Created the repository on GitHub through the web interface
2. Cloned the repository locally: git clone https://github.com/chancehowarth33/ECE554SP26_MiniLab1
3. Added the source files: git add .
4. Committed the changes: git commit -m "Initial commit with Minilab0 files"
5. Pushed to GitHub: git push origin main

Simulation Log:



Waveform:



Resource Utilization Analysis

Comparing the custom Verilog design to the Quartus IP-based design shows that using IP blocks significantly reduces resource usage. The IP version uses 28% fewer ALMs (80 vs 111) and 57% fewer registers (62 vs 144). It also uses simpler logic—the custom design needed 14 complex 7-input functions while the IP design eliminated these entirely. The IP design also has better wiring efficiency, with the maximum fan-out dropping from 152 to 78, which means signals don't need to connect to as many places and the design runs more efficiently. Both designs use the same single DSP block for multiplication.

The main tradeoff is that the IP design uses twice as much block memory (128 bits vs 64 bits). However, this is actually a good thing. The FIFO IP uses dedicated memory blocks instead of regular logic registers to store data, which frees up logic resources for other tasks. This explains why the register count dropped so much—the IP smartly uses the right type of resource for each job. Overall, the IP-based design is more efficient because Quartus has already optimized these blocks specifically for Intel FPGAs, resulting in a cleaner design that uses fewer logic resources while achieving the same functionality.