**GitHub Repository**

**Link:** <https://github.com/jredmundson/ECE554SP25_Minilab0>

**GitHub Usernames**

* **jredmundson** (Jacob Edmundson)
* **jecampos** (Jaime Campos)

**Verilog Files**

**No IP**

* **fifo.sv, mac.sv**: Implemented to enable the top-level module to properly display the dot product.
* **Minilab0.v**: Unmodified top-level module designed to display the dot product on the FPGA without IP. (Expected output: 16'h1B58 or 7000 decimal)
* **fifo\_tb.sv, mac\_tb.sv**: Simple testbenches to verify the functionality of mac and fifo modules.
* **Minilab0\_tb.v**: Testbench for Minilab 0 without IP. Runs Minilab0.v and checks whether the result in the DONE state matches 16'h1B58 (7000 decimal).

**IP-Based Design**

* **FIFO\_IP.v, LPM\_MULT\_IP.v, LPM\_ADD\_IP.v**: IP files generated for the FIFO, multiplier, and adder.
* **Minilab0\_IP.v**: Modified top-level module that replaces the old fifo and mac implementations with IP-based versions.
* **Minilab0\_IP\_tb.v**: Testbench for Minilab 0 with IP. Runs Minilab0\_IP.v and checks whether the result in the DONE state matches 16'h1B58 (7000 decimal).

**Other Files**

* **Minilab0\_Resource\_Usage\_Summary**\*.rpt\*\*: Saved resource usage summaries for Minilab0 with and without IP.
* **minilab0\_sim\_log.png**: Simulation log of Minilab0\_tb.v.
* **minilab0\_sim.png**: Waveform output of Minilab0\_tb.v with the result wave selected.
* **minilab0\_ip\_sim\_log.png**: Simulation log of Minilab0\_IP\_tb.v.
* **minilab0\_ip\_sim.png**: Waveform output of Minilab0\_IP\_tb.v with the result wave selected.

**Difference in Resource Utilization Between Reports**

Based on the two reports, the design using IP was significantly more efficient in terms of logic/resource utilization. Our design used roughly 2x the logic registers of the IP design, and 1.15x more LUTs for logic. They both used the same number of IO blocks as expected, but a big difference was the IP design synthesized block memory, while ours did not. This is likely the reason why ours used many more registers - it was using them as memory (distributed RAM) instead of the block RAM. This could also explain the higher fanout, as more memory cells were scattered around the chip instead of routing to a few compact block rams.