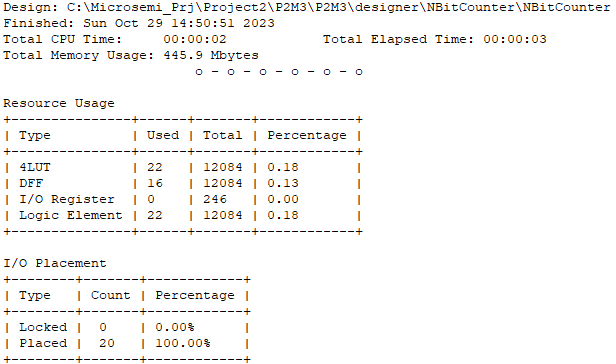
Module III: Load-Testing FPGAs with Counters

Matt Hartnett & Caleb Provost

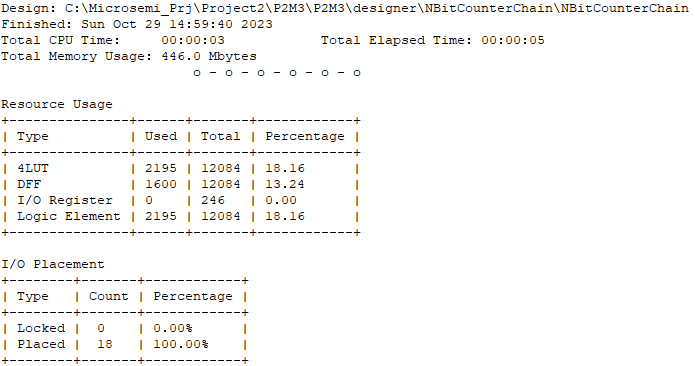
1. Create a 16-bit counter in Verilog or VHDL. Make it into a module or component. Instantiate this module or component into a top-level design, and verify with an RTL simulation that it works. Record your simulation waveform.

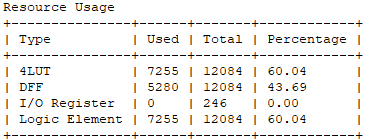




1. Estimate the number of 16-bit counters that will fit into both the Altera MAX10 DE10-lite and the Microsemi SmartFusion2 Maker kit boards. Generate these in your design within the Verilog or VHDL code (Hint: USE FOR...GENERATE) and use the terminal count from one counter to control the count enable of the next counter in a chain. Attach the last terminal count to an I/O pin to force the synthesizer to compile. Try to compile this design in Quartus and Libero respectively, and report the results. How many counters can you fit?  
   NOTE: These compilations may take a long time, many minutes to hours, be patient.

One instance of the counter took 22 4LUTs/LEs with 16 DFFs. I expect some overhead to be added with the generative code’s instantiation of more counters. So I expect 100 counters to take about 2210 4LUTs/LEs and about 1610 DFFs. As seen below, it took less than the amount I originally expected. This is likely due to optimization algorithms while compiling.



If 100 counters takes 2195 4LUTs/LEs and 1600 DFFs, I expect to be able to fit about 550 counters in this chain of counters. However, incrementing the amount of counters above 330 (so far) keeps failing the compilation during the fitting. The below image shows the total utilization at 330 counters:  


1. Now, using Quartus and estimate the maximum number of counters for the DE1-SoC board.
2. Remove counters until each design compiles without error, and report the results. Verify at least one counter output by driving external pins (preferably the LEDs) on one board and using RTL simulation for all boards.
3. Report the number of counters that can be included in each design and still close timing.
4. Record the Fmax in the timing report for each of the 3 boards.
5. Estimate the % utilization of the FPGA logic.
6. Compare all the results from all the boards.
7. BONUS: 5 points each for the team with the verified most number of counters for each of the board designs