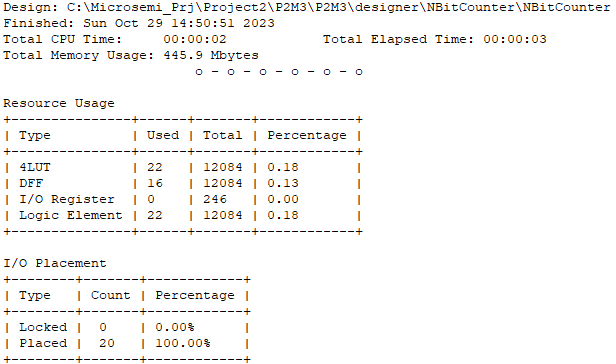
Module III: Load-Testing FPGAs with Counters

Matt Hartnett & Caleb Provost

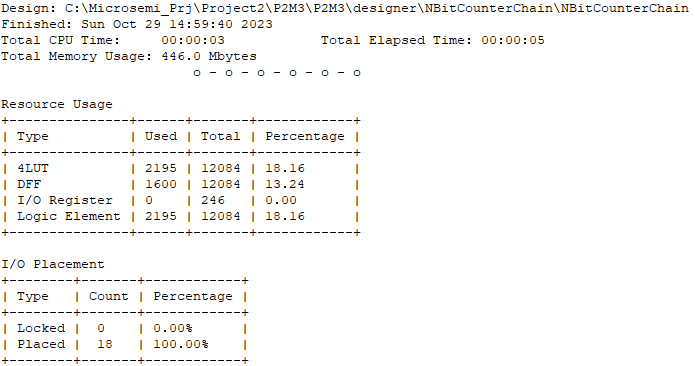
1. Create a 16-bit counter in Verilog or VHDL. Make it into a module or component. Instantiate this module or component into a top-level design, and verify with an RTL simulation that it works. Record your simulation waveform.



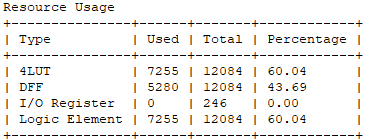


1. Estimate the number of 16-bit counters that will fit into both the Altera MAX10 DE10-lite and the Microsemi SmartFusion2 Maker kit boards. Generate these in your design within the Verilog or VHDL code (Hint: USE FOR...GENERATE) and use the terminal count from one counter to control the count enable 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.

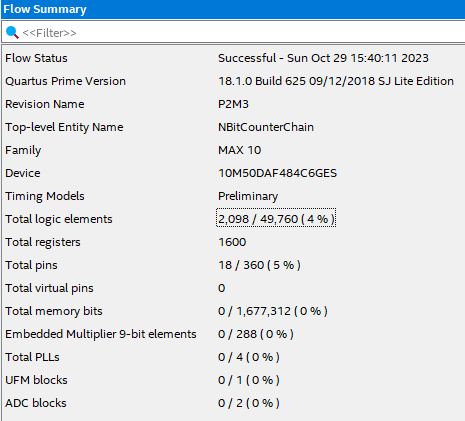
* Microsemi SmartFusion2:
  + One instance of the counter took 22 4LUTs/LEs with 16 DFFs. I expect some overhead to be added with the generative code’s instantiation of more counters. So I expect 100 counters to take about 2210 4LUTs/LEs and about 1610 DFFs. As seen below, it took less than the amount I originally expected. This is likely due to optimization algorithms while compiling.



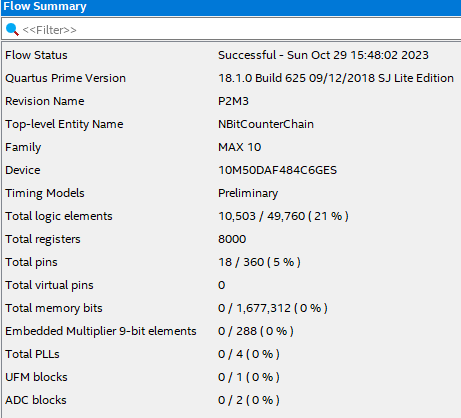
* + If 100 counters takes 2195 4LUTs/LEs and 1600 DFFs, I expect to be able to fit about 550 counters in this chain of counters. However, incrementing the amount of counters above 330 (so far) keeps failing the compilation during the fitting. The below image shows the total utilization at 330 counters:



* Altera MAX10 DE10-lite:
  + 100 counters resulted in about 4% utilization of total logic elements:



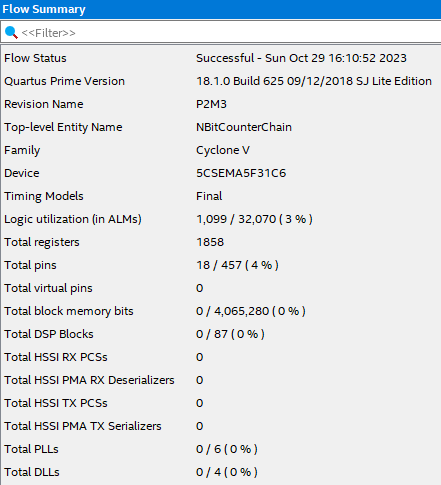
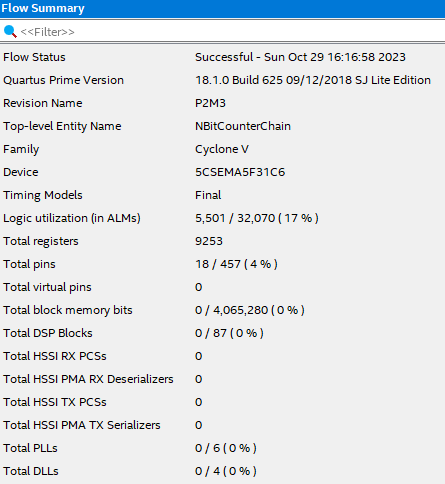
* + We’re able to fit 500 with a resulting change of 8405 more logic elements used. This means there’s about 22 logic elements used per addition of a counter. If the placement is consistent, we should be able to fit roughly 2260 counters onto the DE10-Lite.



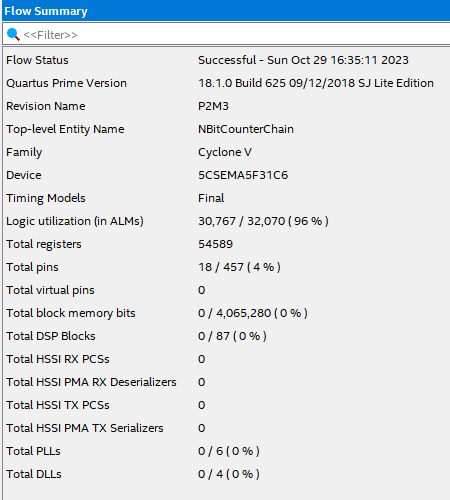
* + Total amount we’re able to fit onto the DE10-Lite is #. Flow summary below as proof:

1. Now, using Quartus and estimate the maximum number of counters for the DE1-SoC board.

Starting the same code onto the DE1-SoC with 100 counters resulted in a total utilization of 1099 ALMs. With 500 counters added resulting in 5501 ALMs utilized, the difference of 4402 ALMs utilized for 400 more counters, we expect to use 11 ALMs per counter resulting in about 2915 counters that can fit onto the DE1-SoC. That is, if the fitting will allow it.

The DE1-SoC is able to fit 2900+ counters resulting in 96% utilization of its ALMs.



1. 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.
2. Report the number of counters that can be included in each design and still close timing.
3. Record the Fmax in the timing report for each of the 3 boards.
4. Estimate the % utilization of the FPGA logic.
5. Compare all the results from all the boards.
6. BONUS: 5 points each for the team with the verified most number of counters for each of the board designs