Lab 1: 10-bit Counter

Reconfigurable Computing ECE 5930
Jonathan Phillips

September 7, 2018
Chris Walker
Xuecong Fan

Procedure

The objective was to construct a 10-bit counter using VHDL (VHSIC Hardware Description Language) on the DE10-nano lite development board. The system needed to display a numeric value represented in binary using the ten on-board LEDs. The system also needed to count up at 1 Hz and include the ability to reset the value by pressing a button.

For simplicity of development, the system was constructed modularly one feature at a time. First, to ensure values were written to the LEDs properly, a VHDL program was synthesized to output a binary value to target LEDs. Following which, a clock divider was engineered to scale down a 10 MHz clock to just 1 Hz. The reset button was integrated into the clock divider to reset values to their default (overriding the divider's functionality). Finally, a 10-bit logic vector was set up as a dedicated binary counter and its values were routed to the 10 LEDs respectively.

A counter was used for the clock divider because no former knowledge of an on-board PLL (phase locked loop) or its implementation were available at the time. An integer was used to count up to one million before rolling over. During each roll-over, the 1 Hz output would toggle creating the desired 1 Hz clock.

Implementing the LED output, clock divider, and reset button when according to plan. However, when the 10-bit counter was implemented, the resulting output produced undefined behavior. After extensive investigation, the problem was identified as having a *rapid* counting experience.

When the 1 Hz clock went active high, the counter would count as fast as the hardware's throughput. Then the counting would stop when the clock went active low. This fault was do to the assumption of C-like synchronous functionality in the concurrent if-statement. The fault was resolved by incrementing the counter every time the clock divider counter rolled over.

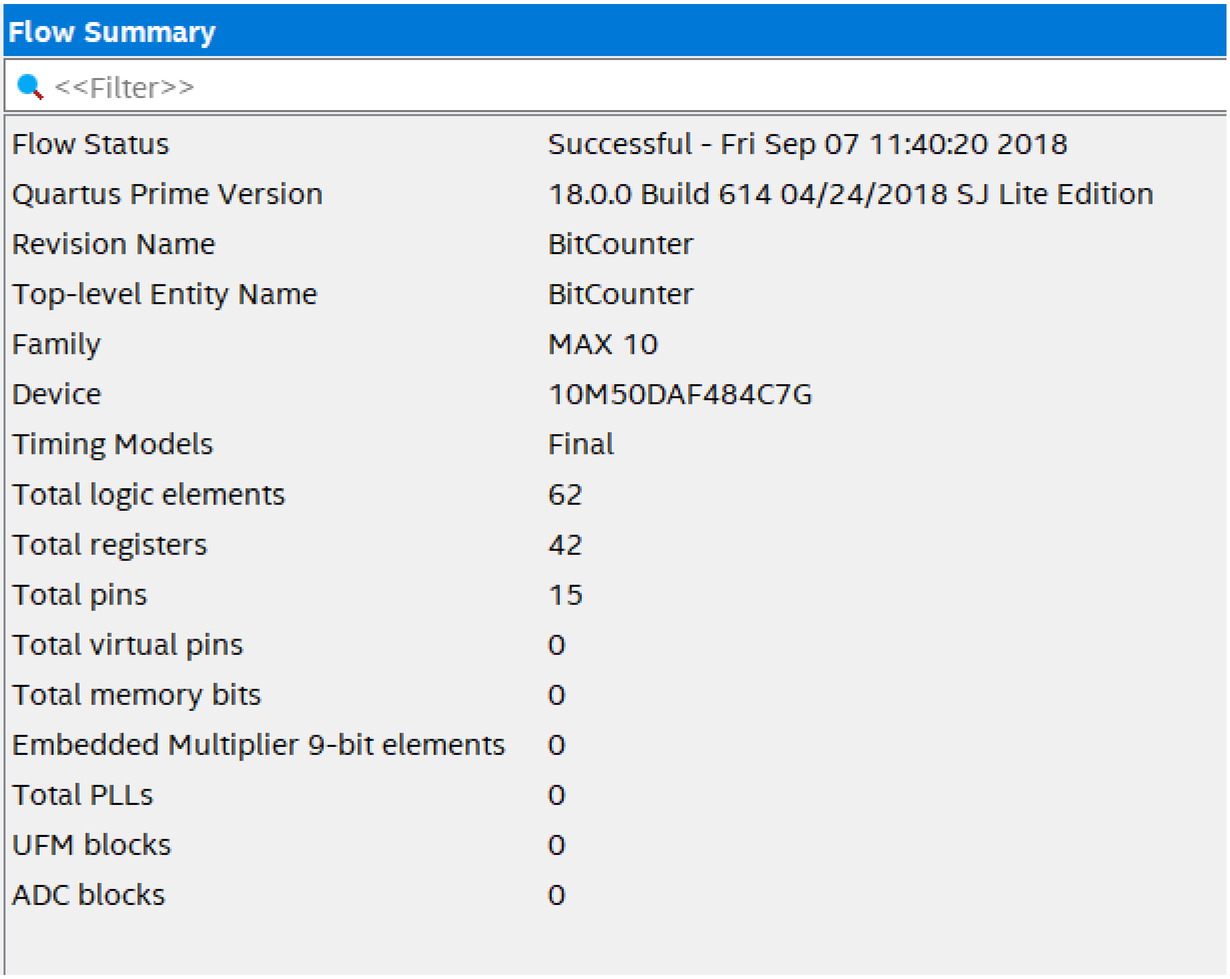
Finally, after all was working, the counter speed was adjusted to twice the speed since it was counting at once every two seconds. The 1 Hz clock was in fact toggling at 1 Hz but was producing an 0.5 Hz output clock. By removing the *active high* condition for the 10-bit counter inside the clock counter roll-over condition, the desired 1 Hz output was achieved.

Results

The results for this lab achieved goal was use VHDL code on the DE10-nano board. In this lab VHDL code let LED on the board light 1Hz and count number in each second. Then add them one by one and show in the number from 1 to 10 LED based on binary pattern. The reset button reset the number on the light and made when reset button pushed the number start from 1 again.

The result for this lab it ends up as expected, but during test result on the DE10-nano board it always show same mistake or wrong light. That’s because in the VHDL code it always run same wrong loop, the counter always unchanged. Therefore, the code will have stuck inside the loop and always show same things. To avoid the logic mistake of the code in VHDL need to see code carefully. If a same mistake always happens then try to follow the code in each line, it will give the answer.

Figures



Flow Status	Successful - Fri Sep 07 11:40:20 2018
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	BitCounter
Top-level Entity Name	BitCounter
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	62
Total registers	42
Total pins	15
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Figure 1. Screenshot of the synthesized result in Quartus II

While the flow summary indicates that 15 pins were used in total, only 12 pins were actually used in the execution of the logical input and output - clock, reset button, and 10 LEDs.

Conclusion

The process was modularized into the most basic components and then synthesized in order of importance. LED output, reset and driving clock inputs, as well as clock scaling and binary counting were all addressed in the design. It was discovered that logic vectors made binary counting and signal busses convenient. The overall project took no more than 40 lines of VHDL code.