Lab 2 Report

Matthew Crump A01841001
Nicholas Williams A02057223

Objectives

Implement a stopwatch on the DE-10 Lite development board. Become familiar with how to instantiate memories and how to drive the DE-10 Lite seven segment displays. While also furthering knowledge and familiarity with VHDL.

Procedure

We started out by reading the lab-2 requirements for the stopwatch project. We then used the System Builder tool to initialize the Quartus Prime project. We determined which I/0 would be needed by this lab, those being: the system clock, push buttons, seven segment displays, and LEDs (so they don't glow dimly). We started by drawing up a basic block diagram for the stopwatch. We had a separate 'tens' counter for each seven segment display and chained them together so that when one rolled over the next one incremented. To drive the first digit counter, we added a counter that sent out a tick every 100th of a second.

Since the seven segment displays are not multiplex, we added decoders for each segment. This was done by creating an array of vectors to act as a lookup table to decode the counter value. We looked at the board schematic to find out which bit controlled each segment then converted the binary value to hexadecimal. After programming the board we noticed that the displays were inverted, as the segments are active low. This was quickly fixed by inverting the assignment to the displays.

After building the counter and segment decoder modules, we connected and chained them together and tried compiling the design. After fixing a few minor type mismatches the design compiled. However, the resource utilization was zero. After looking at the RTL schematic we did not find the problem. Eventually we discovered that the counter that generates the 100th of a second pulse, did not have enough bits to represent a value of 500,000. This caused everything to be optimized away until nothing remained.

After fixing the last issue, we made a testbench to simulate the stopwatch. Looking at the waveform we discovered a major bug. The 100th counter was working fine but the next counter was stuck incrementing every clock cycle. This was caused by the tick

output not being able to go back to zero because the enable was always high (as the push button is constantly held). We found an easy workaround by adding a second architecture to the counter module. We set one architecture to expect the enable to be high constantly and the second architecture to expect the enable to be a pulse. The 100th counter used the first architecture while the other 6 segment counters used the second architecture.

Results

Simulating the design showed that our implementation worked correctly with both counter architectures (see Figure 1). Having the reset high also showed the counter being reset back to zero for each segment. Each module produced a pulse once the counter rolled over, causing the next counter inline to increment. We also tested out edge cases for button presses in our simulation.

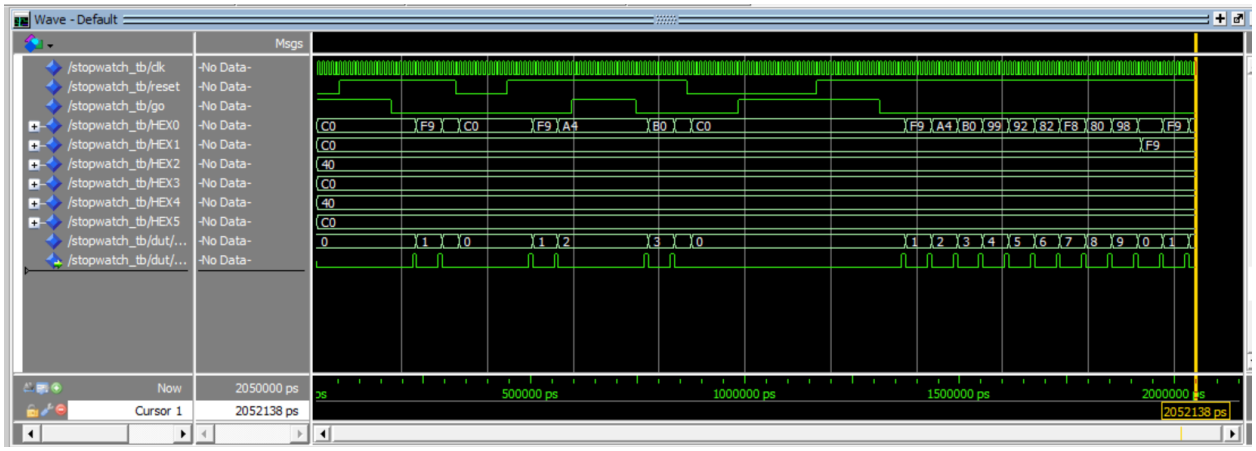
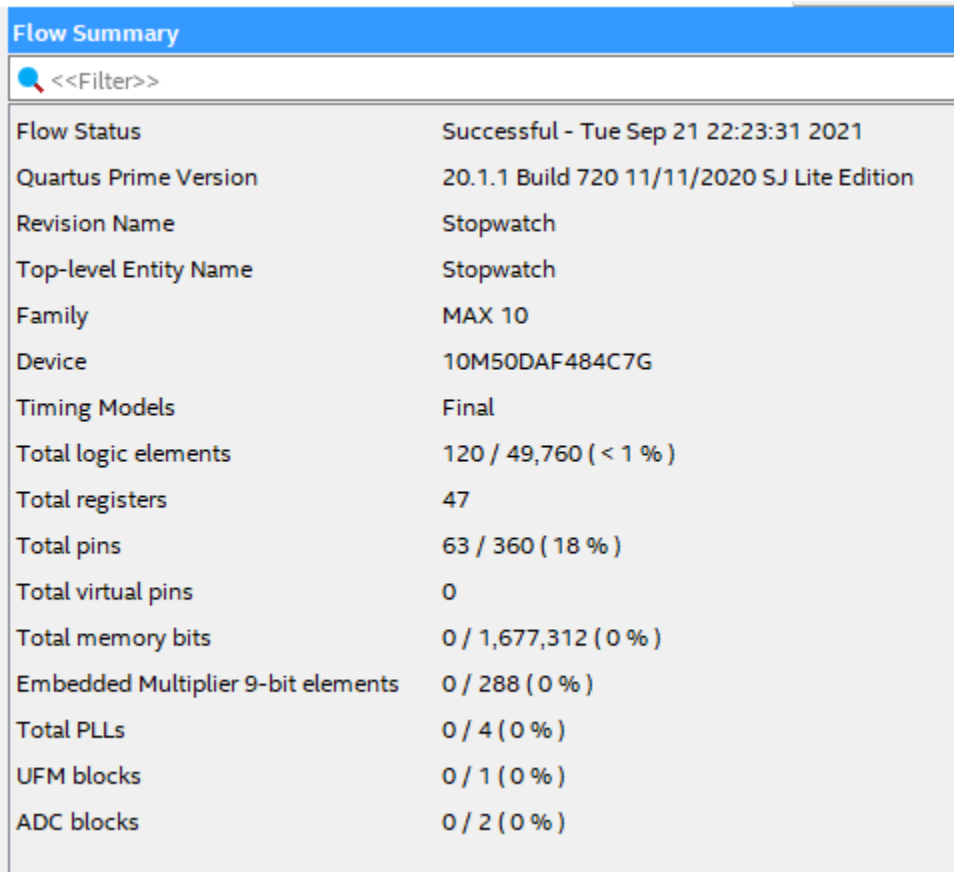


Figure 1. Simulation Waveform

The design worked as expected on the board. We used a stopwatch to time our design for one minute and found they both reached a minute at the same time. But this allowed us to notice a small bug. The tens spot for seconds would count all the way up to 9 when it should count only to 5 then overflow into the minutes' ones spot. Since we used generics, this was a quick change. We modified the seconds' tens spot to have a counter limit of 5 instead of 9. After reuploading to the board, the seconds properly rolled over into the minute's count.

We also designed the segment counters to have a max limit of 15 and the decoder has 16 entries for hexadecimal. This is so that if we need to reuse this lab for another project we can easily modify the generics to allow for a range of 0x0-0xF to be shown

on the displays. Figure 3 shows our lab's flow summary. The total amount of logic elements, registers, and pins used in our design seemed appropriate.



Flow Summary

<<Filter>>

Flow Status	Successful - Tue Sep 21 22:23:31 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	Stopwatch
Top-level Entity Name	Stopwatch
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	120 / 49,760 (< 1 %)
Total registers	47
Total pins	63 / 360 (18 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

Figure 2. Flow Summary

Conclusion

We gained a better understanding of VDHL, and we learned how to better use unsigned and natural numbers. We learned specifically that you can define a natural number as a range but unsigned numbers require a bit width. We learned how to use ModelSim to simulate the design and add signals that were several entities deep. We have become more familiar with how to use different architectures for the same entity.