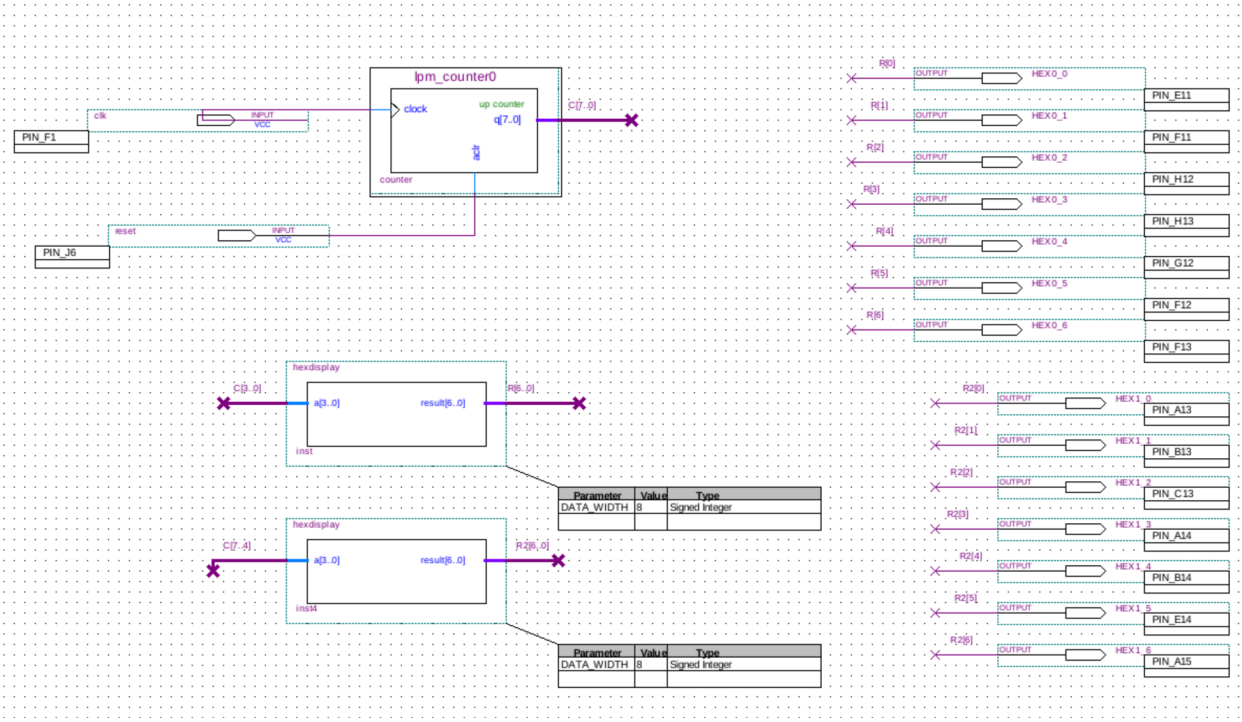
CS232 Project 2: Double Sevens

September 23rd, 2024

Task 1: 7-Segment Display Driver

This task involves designing a circuit to control a 7-segment display on a DE0 board. Each display is made up of seven individual segments, which can light up in specific patterns to represent numbers or letters. The circuit will take a 4-bit binary input and convert it to the corresponding hexadecimal digit by activating the correct segments to display the number.

Circuit:



Simulation:

https://drive.google.com/file/d/1o3QmtBe5nn8NbEJb5gR3ay8QR9D-K00E/view?usp=sharing

Explanation:

3 things needed to happen with this circuit: One, the reset switch, which I did at the very beginning, when it flipped to 00. Another was the display, as I manually coded which of the 7 different segments to light up for the digits 0-9 and A-F. In the video, you can see it cycle through to spell each of the letters, so I think that works. Lastly was the counter button, which was meant to add 1 to the number every time I pressed the key. Since the display continuously cycled from 0 to F without issue, it shows that the counter was working as intended, increasing the input value on each press and properly updating the 7-segment

displays. Fortunately, the way hexadecimal worked was that everytime the least significant digit completed a full circle the most significant would increment. As seen in the video, this is the case.

Task 2: Add Two Numbers

In this task, we create a circuit that adds two unsigned 4-bit binary numbers. The sum of these two numbers is displayed as a hexadecimal value using two 7-segment displays, allowing for easy visualization of the result.

Circuit:

In the VHDL File titled test2.vhd

Simulation:

https://drive.google.com/file/d/1AfG6YGZvaOc7ZBJXsyX9EHmwApUgY5AF/view?usp=sharing

Explanation:

I used the first 4 switches to represent the first 4-bit binary number, and the second set of 4 switches for the second 4-bit binary number. After setting the switches for both numbers, the circuit correctly calculated their sum and displayed it in hexadecimal on the two 7-segment displays. To further test the circuit, I flipped the positions of the two 4-bit numbers by switching their input configurations. The sum was still calculated correctly regardless of which set of switches controlled which number, confirming that the adder was functioning as expected.

Extension 1: Adding 2 Bit 2's Complement Numbers (Output in Decimal)

https://drive.google.com/file/d/1dYhhk2Tu-KNgCsrYIeb4VSbsnC5x9NCz/view?usp=sharing

In this extension (project name: twoscomplement), I engineered the switches to account for 2's complement 2 bit numbers, by appending 1 extra switch per each binary input to tell the circuit the sign. The LED display shows what the sum of the 2 inputs (SW5 - SW3 for the first number, and SW2- SW0 for the second) looks like, so when for example, when I had the switches flipped as 100 each, (equivalent to -4 in binary) the circuit outputted the sum, -8 accordingly. Everything else works the same, except the VHDL file now detects when to put a negative sign as the second LED.

Extension 2: Operating on 3 Bit 2's Complement Numbers (Output in Hex)

https://drive.google.com/file/d/1C6WhbmDpT4wICJ6LnVeSvEV-bplUofDP/view?usp=sharing

This extension (project name: operations) takes the previous extension a lot further, by not only adding another bit to each of the outputs, but relegating the two leftmost switches to determine what sort of operation would occur between the two inputted numbers: 00 for addition, 01 for subtraction, 10 for multiplication, and 11 for division. I would add more bits if I could, but there are only so many switches

that I can use! This extension also takes into account how to represent 2's complement Hexadecimal numbers, (8-F second digit means negative!) although unfortunately, I do not have the time to implement decimals. (The explanation that the circuit works is in the video this time!)

Acknowledgements

At least 4 different TAs helped with troubleshooting and VHDL coding, since it's a completely new language for me… Most importantly, Professor Li helped herself with understanding what my code was actually doing, so it should be useful knowledge going forward.