Class:	**CPE100L-1002**	Semester:	**Fall 2020**

Points		Document author:	**Edreese Basharyar**
		Author's email:	**basharya@unlv.nevada.edu**
		Document topic:	**FINAL PROJECT**

Instructor's comments:

1. Description of Project:

I have decided to use the previous fundamental concepts that I have learned in this class to implement my own idea for the final project. I have decided to make a digital clock (with a range of 0:00:00 - 9:59:59) onto the seven-segment displays of the DE0-CV board. This project involves a lot of concepts that we used in this class, such as counters, flip flops, oscillators, combinatorial/sequential logic, and other fundamental concepts. In order to make this digital clock, I first started making the 4-bit counter. This counter was made using 4 JK flip-flops, which will have a range from 0-15. Then, I connected the outputs of the counters into a decoder that we made in the previous lab, which would be shown on the 7-segment display. Then, I realized that a clock functions automatically, so I used the oscillator that we used in the previous lab as the input signal for the 4-bit counter. Now, the 4-bit counter will count up automatically at a frequency of 1hz, which simulates a digital clock. However, the second digit of the seconds needs to go from 0-9. So, I connected the four outputs to a 4-AND gate, so that once the output reaches the value of "10", it resets the 4-bit counter. Thus, the four-bit counter will count up from 0-9 and then reset in a continuous loop with the help of the oscillator. The next step is creating the "tens" place of the seconds, which has a range of 0-5. However, only one oscillator was used for this project, as multiple oscillators had created problems. So, I saved what I had as a symbol block, and made another symbol. The difference with the second one was that I had to divide the frequency by 6, so that the counter goes up once the first four bit counter reaches "10". This frequency can be divided by making another 4-AND gate that makes it so the counter resets once it reaches the value of "6" all while using the same oscillator input. Now, the "seconds" part of the clock has been created, as it goes from 0-59 together. The next step is to create the "minutes" portion of the clock, which has the same process as the "seconds" portion. We now need to divide the frequency again by 10, and create another symbol block. Dividing the frequency by 10 makes it so the counter will go up once the "seconds" has passed the value of 59. We connect the output of the oscillator from the previous symbol block, and make it the input of the new symbol. We then make a 4-AND gate that resets the "ones" minute portion

once it has past the value of 10. We repeat this process for the "tens" minute portion by creating another symbol, and dividing the frequency by 6. Repeating this process again will make it so the "tens" minutes portion goes from 0-5. Finally, the clock has been created so that it will go from 00:00 to 59:59. I have also created a symbol block for the first "hours" digit, so that 1:00:00 can be seen after 59:59 passes. For convenience, I have also created a reset function for the entire clock, so that it will reset the clock back to 0:00:00 at any given time and continue to count up once it has been released. I am proud that I was able to make this project just with the use of the concepts learned in this class, as it is one of the biggest projects I've made in digital logic circuitry.

2. Goal of Final Project:

The goal of this final project was to look at the potential and capability of what I could achieve with the knowledge that I have gained from just this class. With this in mind, I believe that a digital clock combines all of the fundamental concepts of this class to make it a reasonably difficult project while knowing that I had the ability to make it with some time and effort.

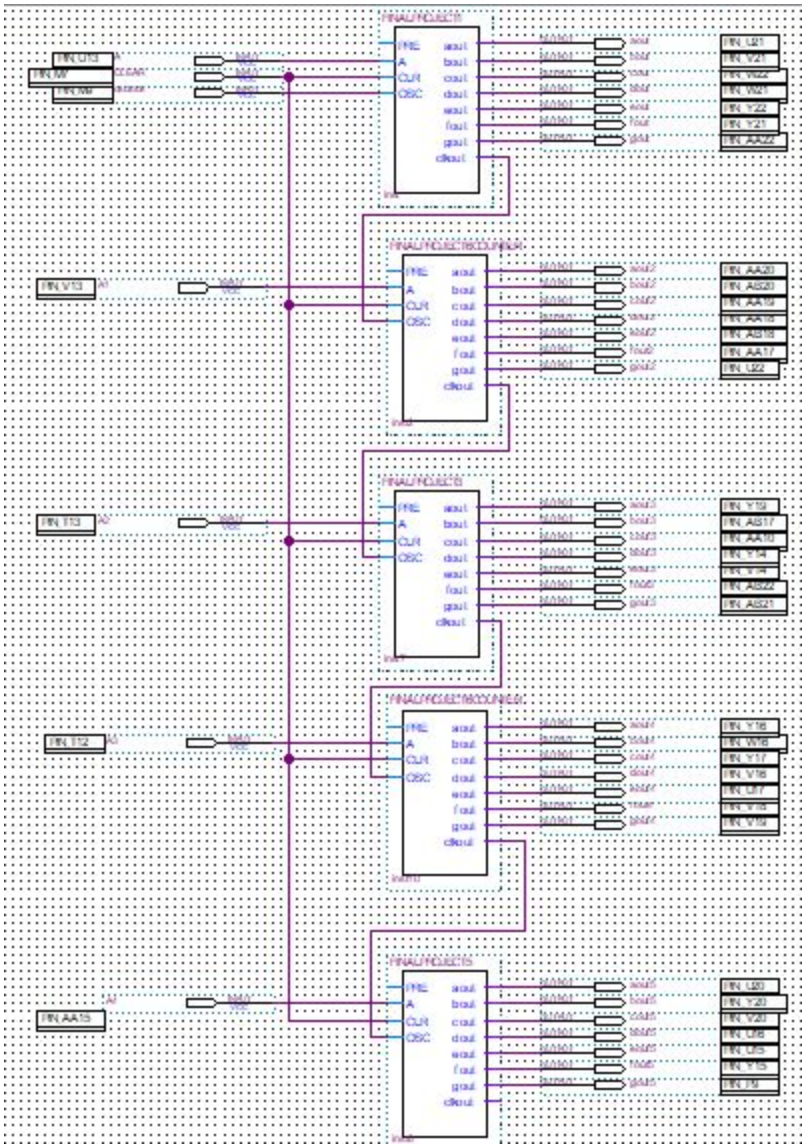
3. Background Theory:

In this project, I am using the concepts of sequential logic, decoders, counters, oscillators, memories, flip-flops, logic gates, and other general ideas to create a digital clock in the format of HMS along with a clear function to reset the clock. The clock will move up automatically with an oscillator of 1 Hz so that it is similar to an actual digital clock. One oscillator is used for the clock and is divided by the counters based on each digit. The full explanation of the project can be seen in part 1 of the final project.

4. Schematics and Diagrams / 5. Circuit Operation

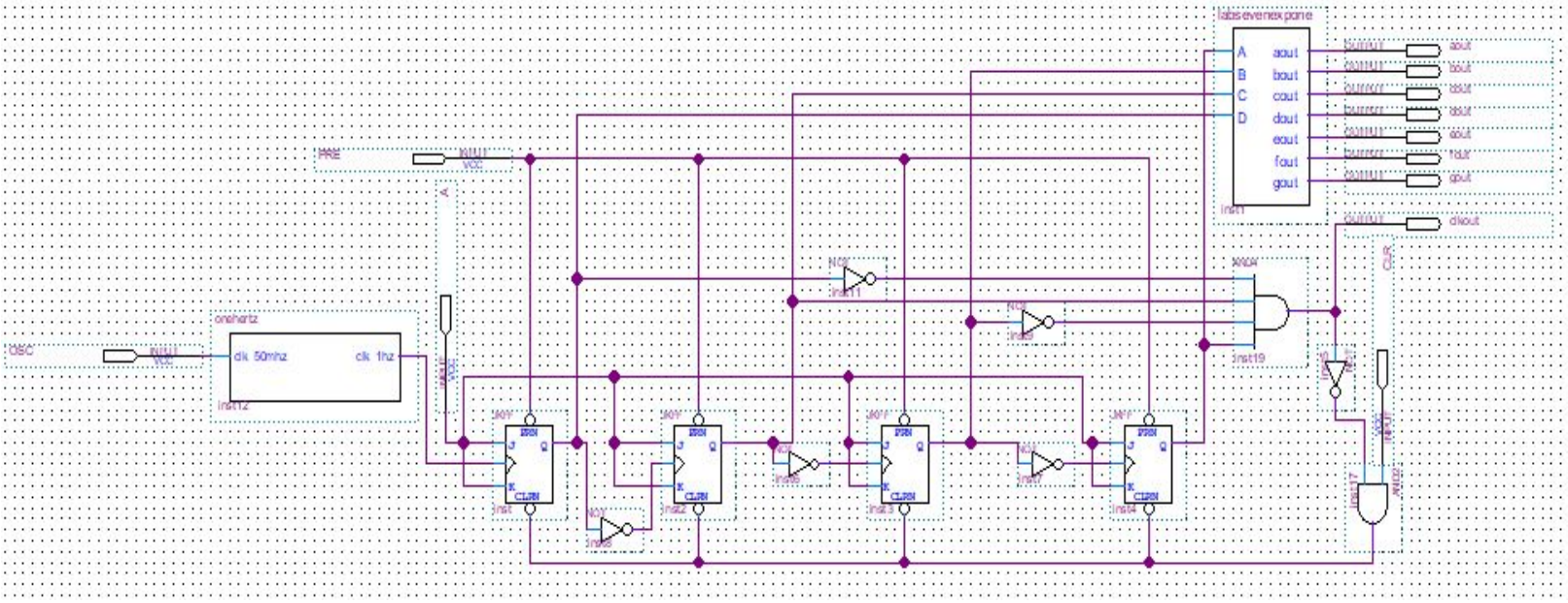
Since the digital clock is fully condensed into many symbol blocks, I will show each design from the inside to the final schematic. For reference, I will show the schematic that is fully condensed first as a reference to all other designs. I will also explain how the schematics operate under the screenshots to show how the digital clock functions altogether.

Screenshot of Final Schematic:



With all of the symbol blocks combined and all pins assigned, a digital clock is made with a range of 0:00:00 - 9:59:59. The clear function can be seen at the top and is connected to all of the symbol blocks, and is assigned to a push button.

Schematic of First Symbol Block:

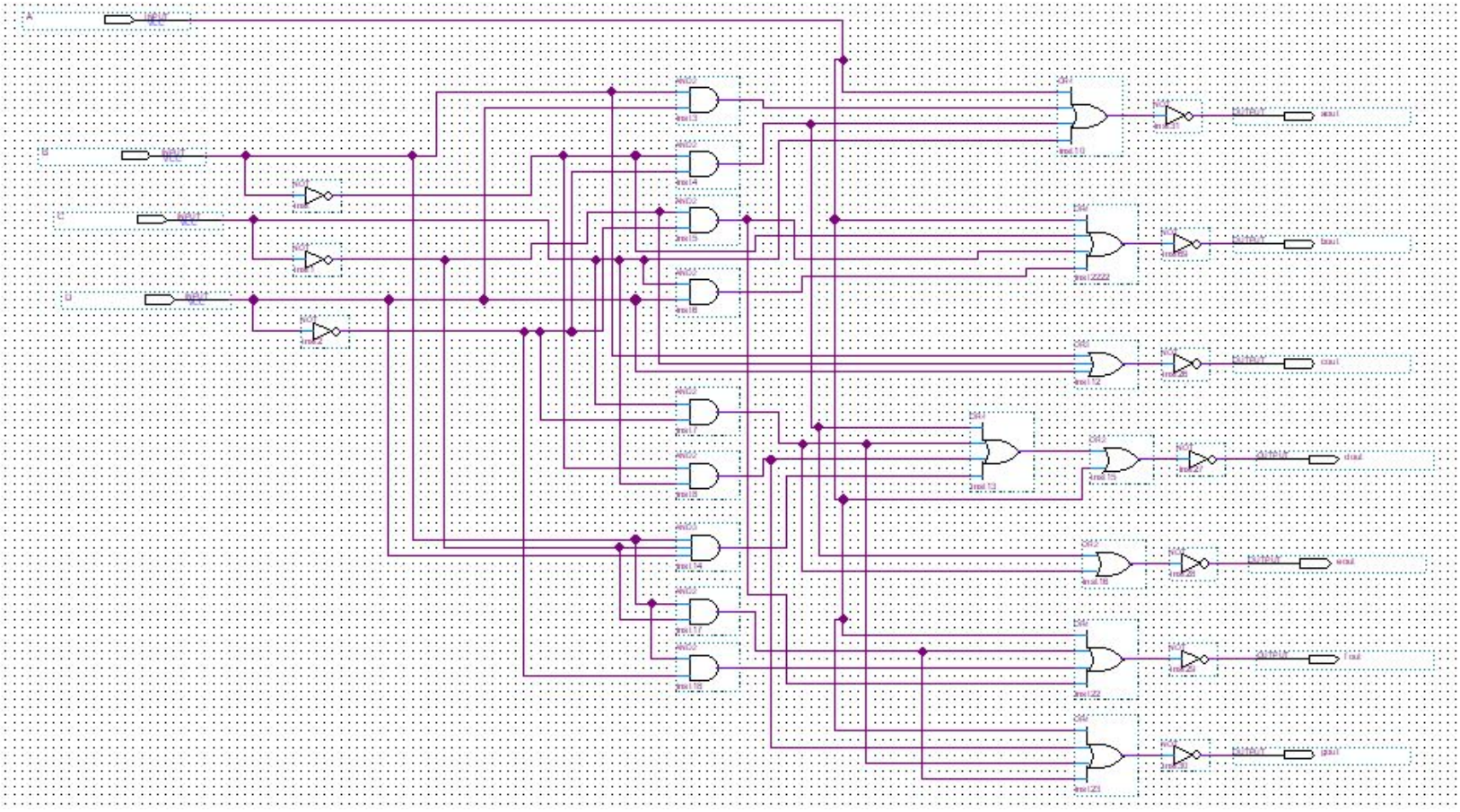


Each symbol block will have the same structure, but with minor differences. In this symbol block, the 4-AND gate uses the outputs of the 4 counters as an input that is true only when the binary value is 1010, or 10. This essentially divides the frequency by 10 for the next clock input because the clock output is connected to the clock input of the second symbol block. It also activates the clear function so that this symbol block has a range from 0-9. The four outputs of the counter are connected to the decoder, where its outputs are assigned to the 7-segment display on the DE0-CV board.

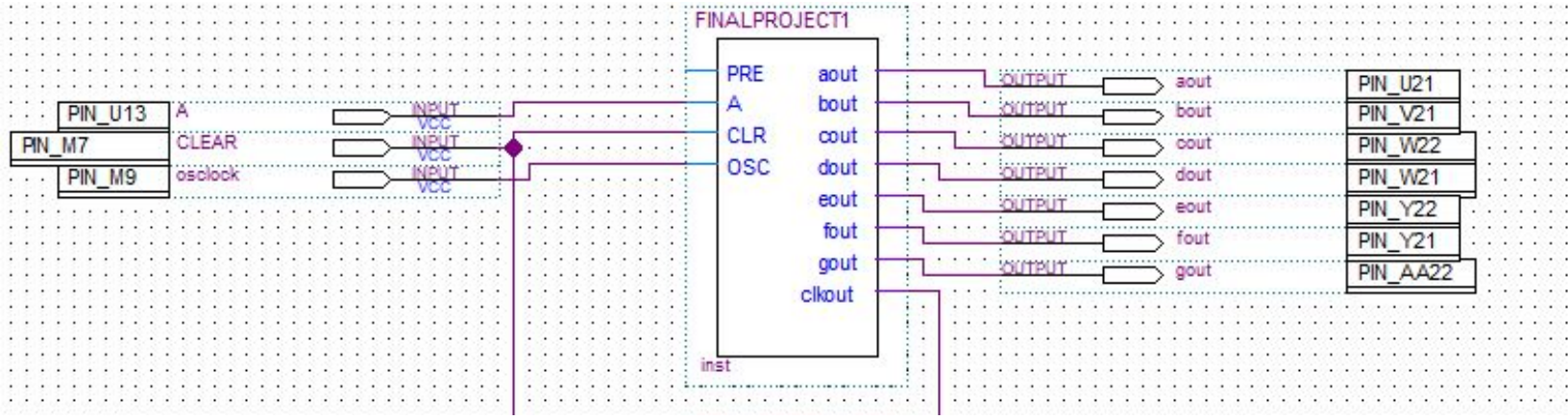
The oscillator and 7-segment display decoder symbol blocks were previously used in other labs, and I have implemented the same ones onto this schematic. The oscillator was made using Verilog code, which is seen below:

```
1   module onehertz(clk_50mhz, clk_1hz);
2   input clk_50mhz;
3   output clk_1hz;
4   reg clk_1hz;
5   reg [24:0] count;
6   always @ (posedge clk_50mhz)
7  begin
8  if(count == 24999999) begin
9   count <= 0;
10   $dumpfile("f.vcd");
11   clk_1hz <= ~clk_1hz;
12  end
13  else begin
14   count <= count + 1;
15  end
16 end
17  endmodule
```

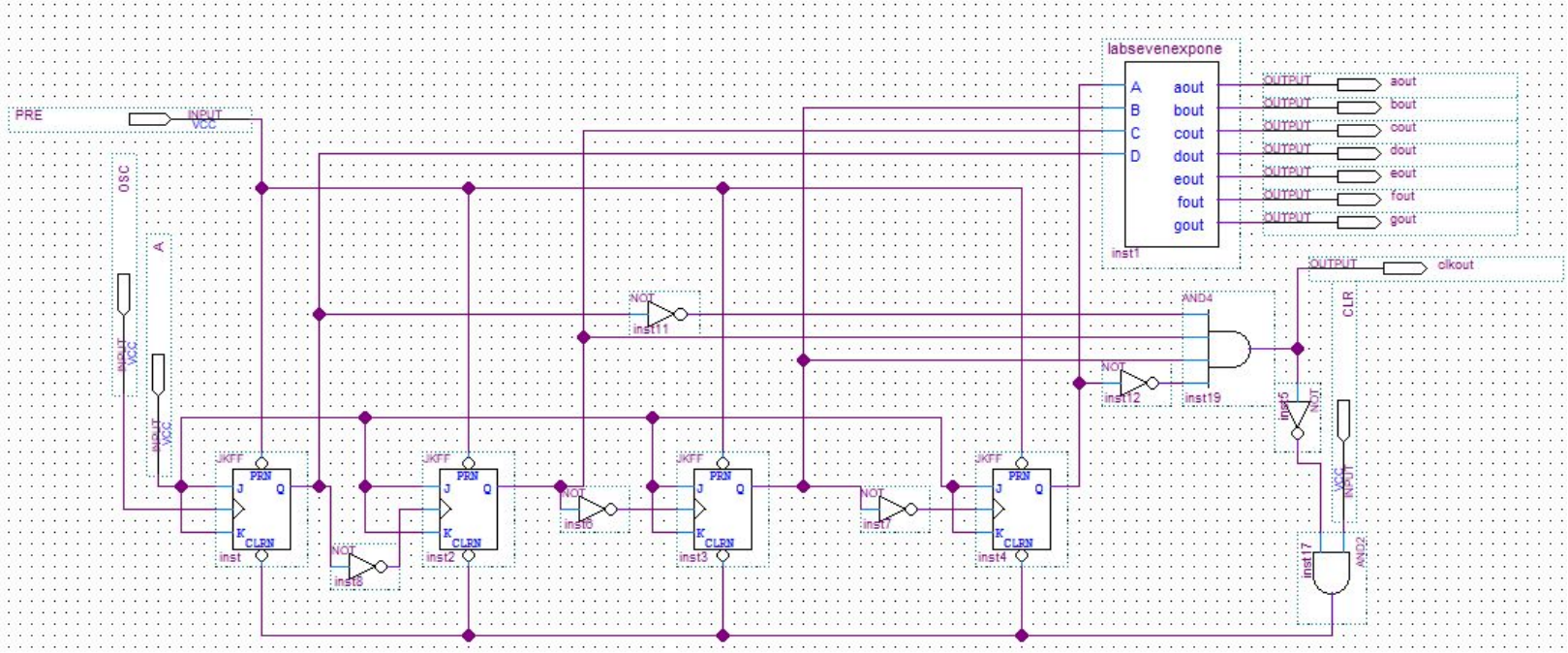
This is the BCD decoder symbol block that was used in the first symbol block (made in previous lab) :



Schematic of First Symbol Block (condensed):

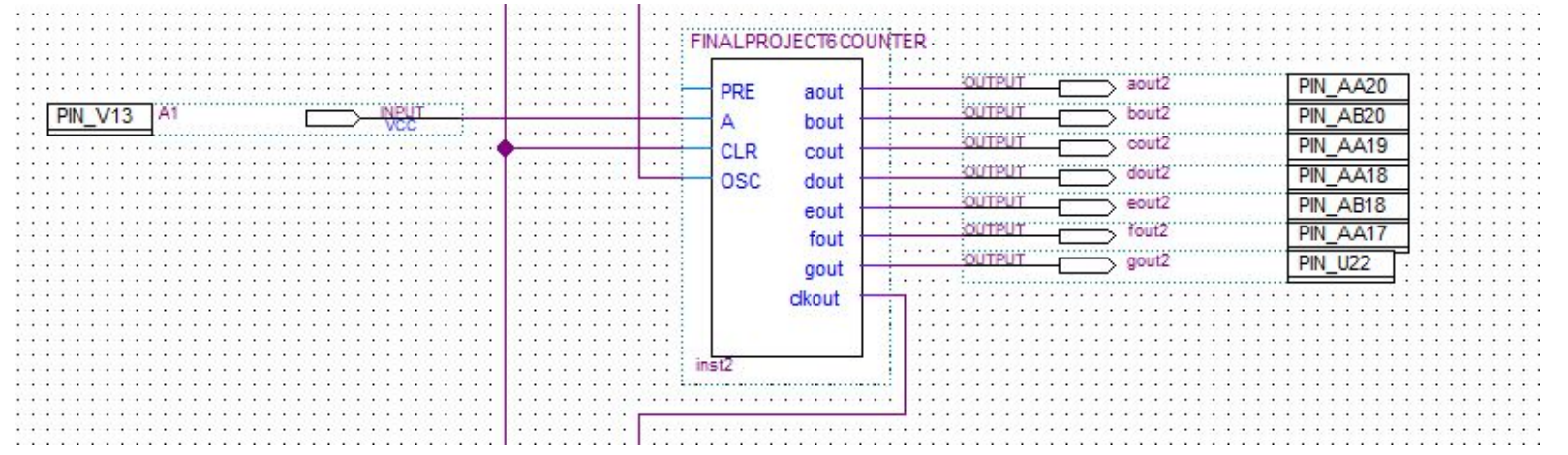


Schematic of Second Symbol Block:

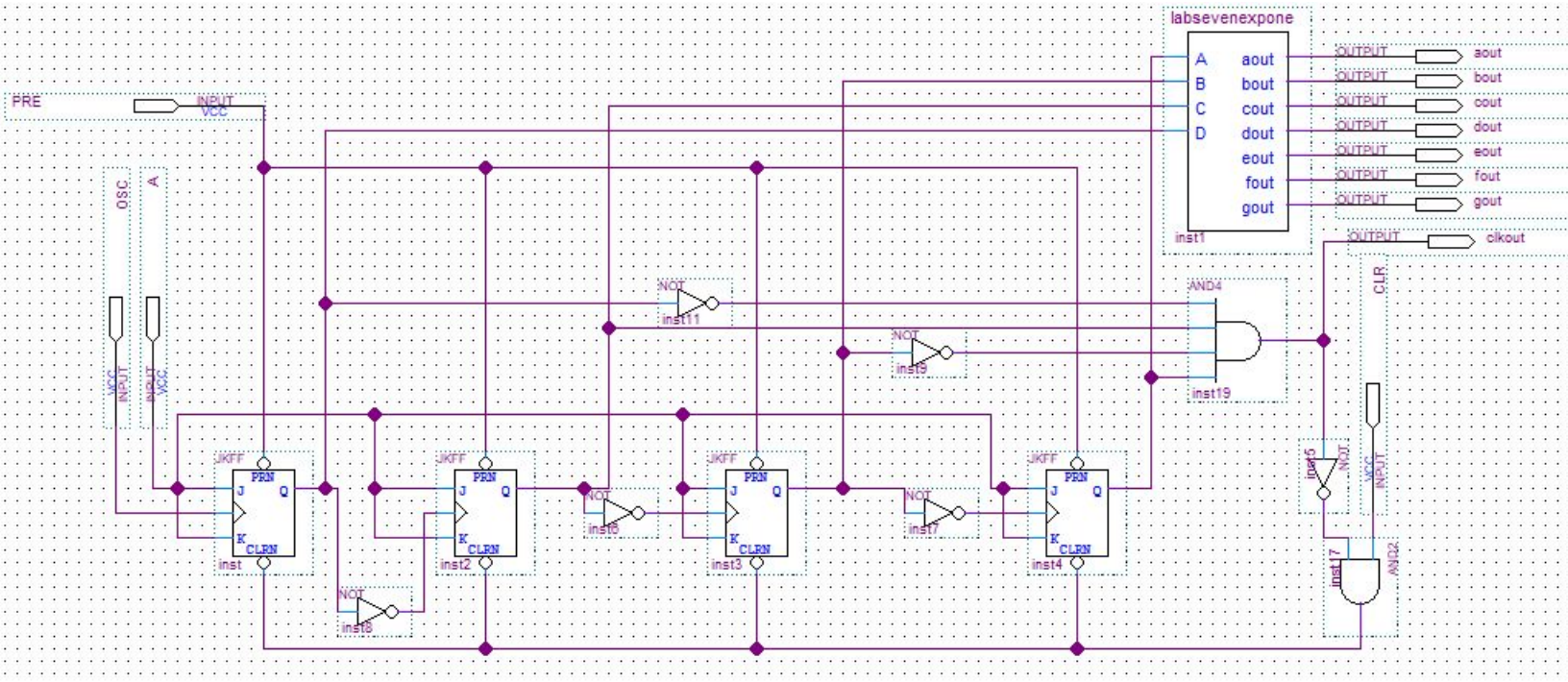


This is similar to the first symbol block, but there is no independent oscillator symbol block in this schematic. This is because the clock input in this schematic is dependent on the output of the first symbol block. This is because the oscillator has already been divided by 10 from the output of the first clock, and we can use this as an input and divide the frequency again by 6. We can see this with the implementation of a 4-AND gate that uses the outputs of the 4 counters as inputs. In order for the AND gate to be true, the value this time is 0110 (6), which can be seen with the inverters for the first and last counter bits. This AND gate divides the frequency by 6 for the clock output, which is now the input for the third symbol block. It also clears the values of the counters once it reaches 6, so that the 4-bit counter has a range of 0-5. Having one oscillator makes the counters sync up together rather than having multiple oscillators with varying frequencies, which makes it harder to sync the digits of the clocks altogether. The BCD decoder at the top right is the same as the first symbol block, so another screenshot is unnecessary to show.

Schematic of Second Symbol Block (condensed):

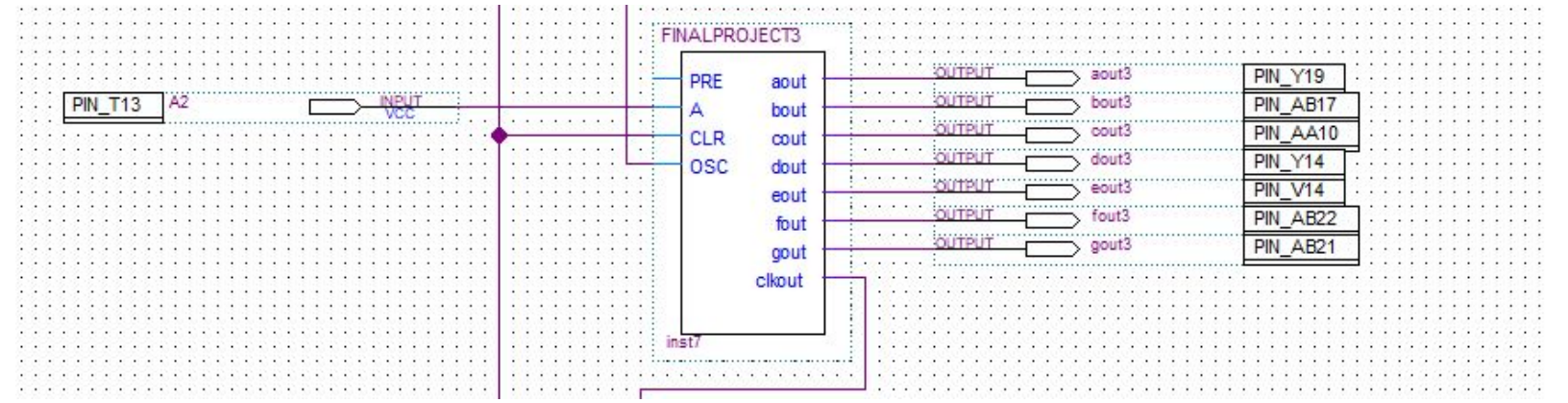


Schematic of Third Symbol Block:

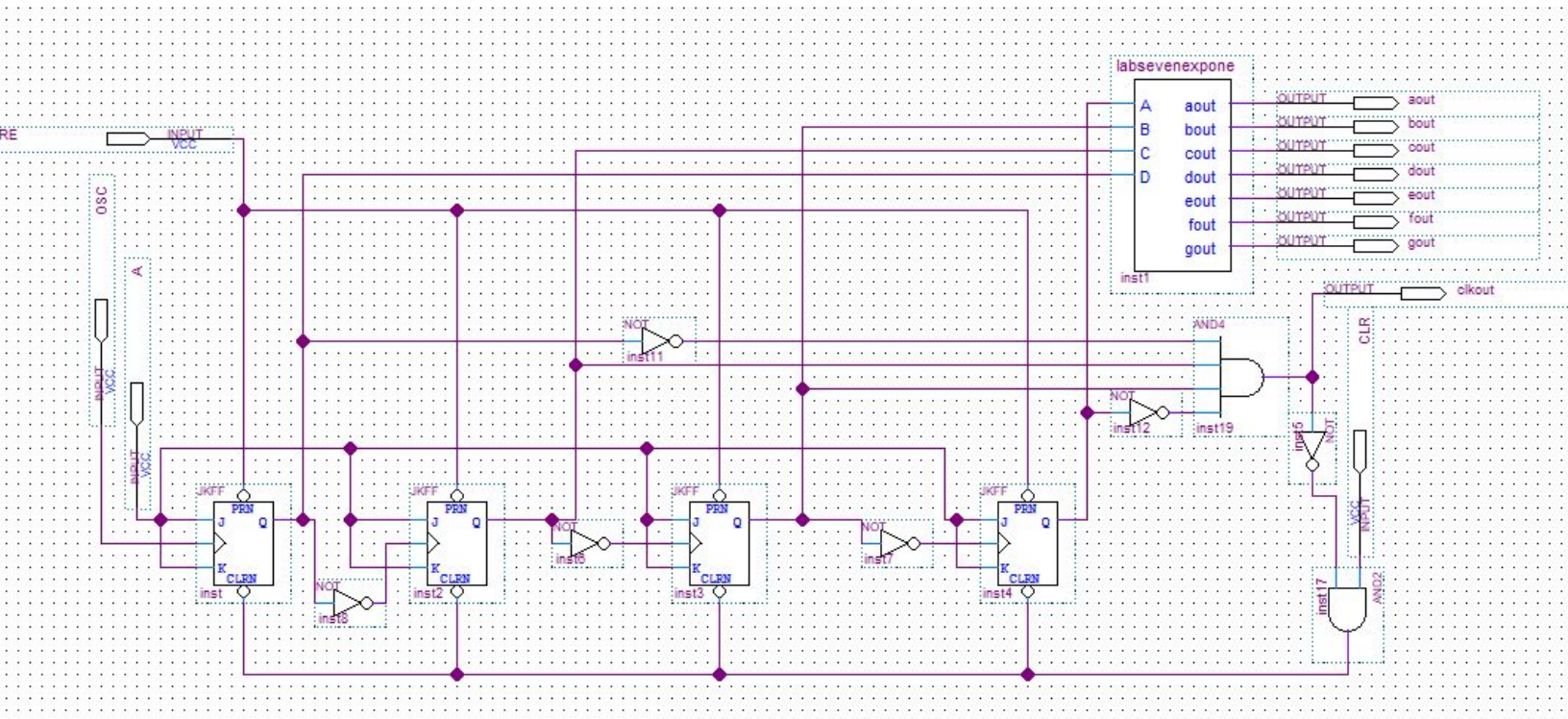


This symbol block is very similar to the first block, since we have to divide the frequency by 10 again in order for the "ones" digit of the minutes. There is no oscillator because it is dependent on the output from the second symbol block. We can see in the 4-AND gate that the value needs to be 1010 in order to be true, which activates the clock for the fourth symbol block and activates the reset of this 4-bit counter so that its range is 0-9. The decoder is used to display the values of the counter on the 7-segment display. Otherwise, the remaining structure is the same as the other symbol blocks.

Schematic of Third Symbol Block (condensed):

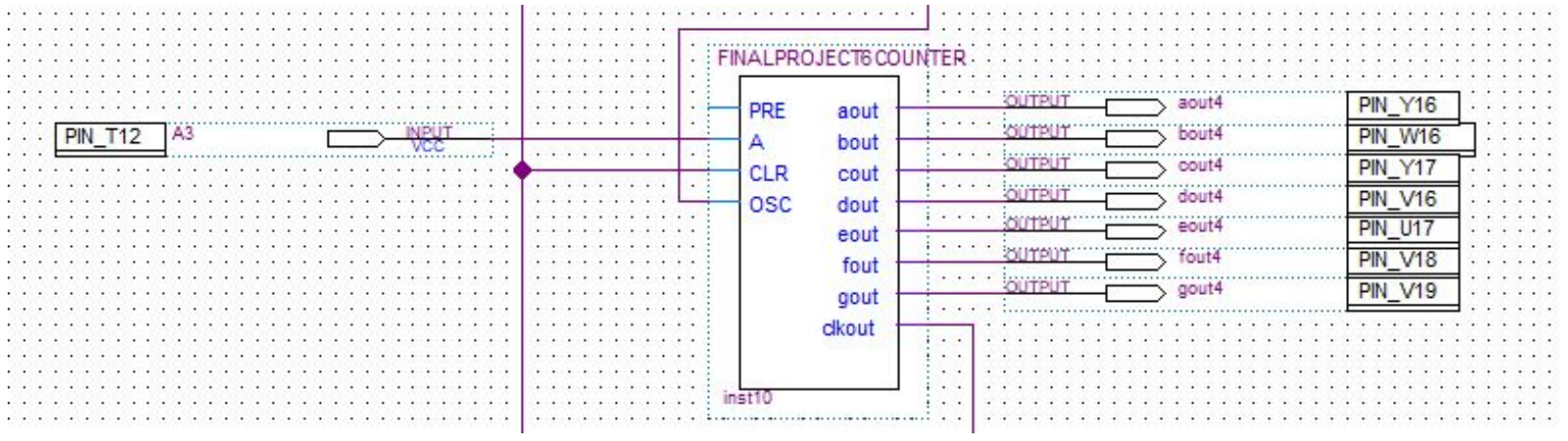


Schematic of Fourth Symbol Block:

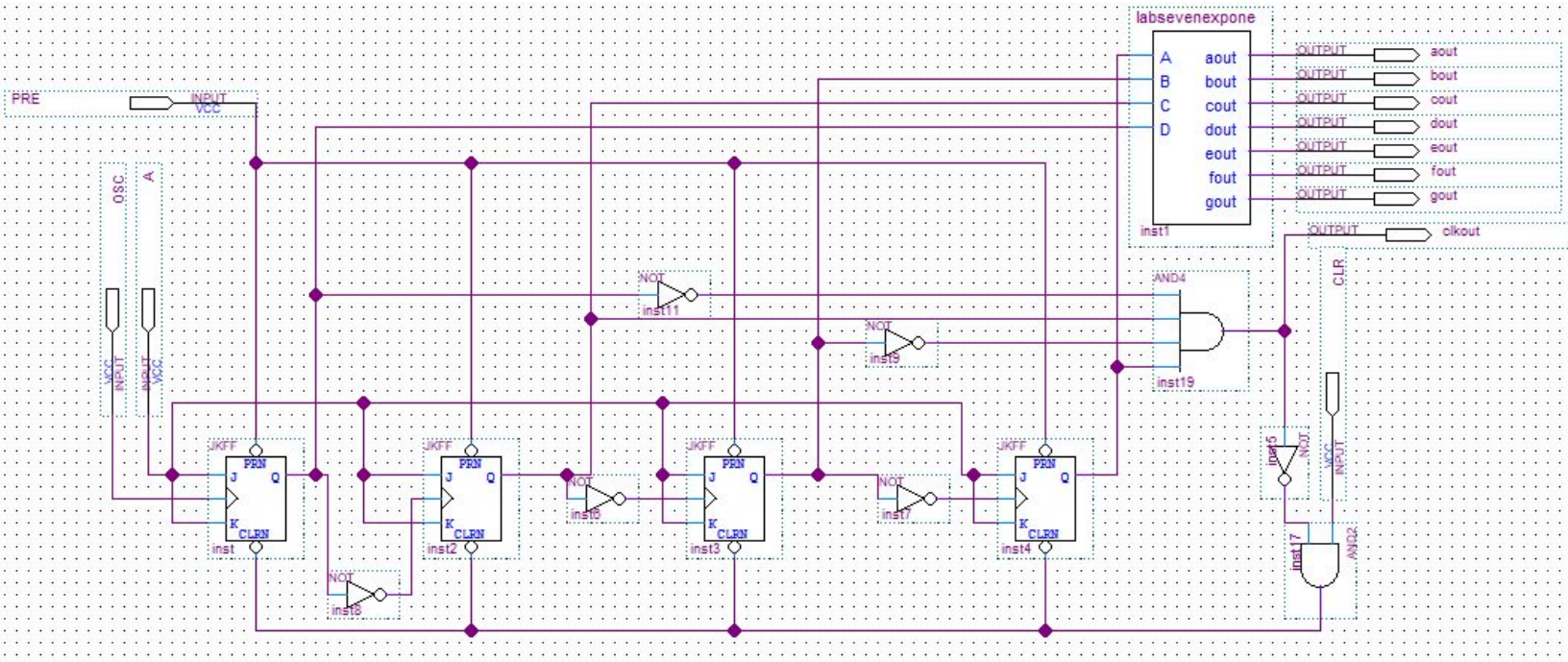


This schematic is exactly the same as the second symbol block because the same functions were required to make the second digit of the minutes. The oscillator input is dependent on the output of the third symbol block.

Schematic of Fourth Symbol Block (condensed):

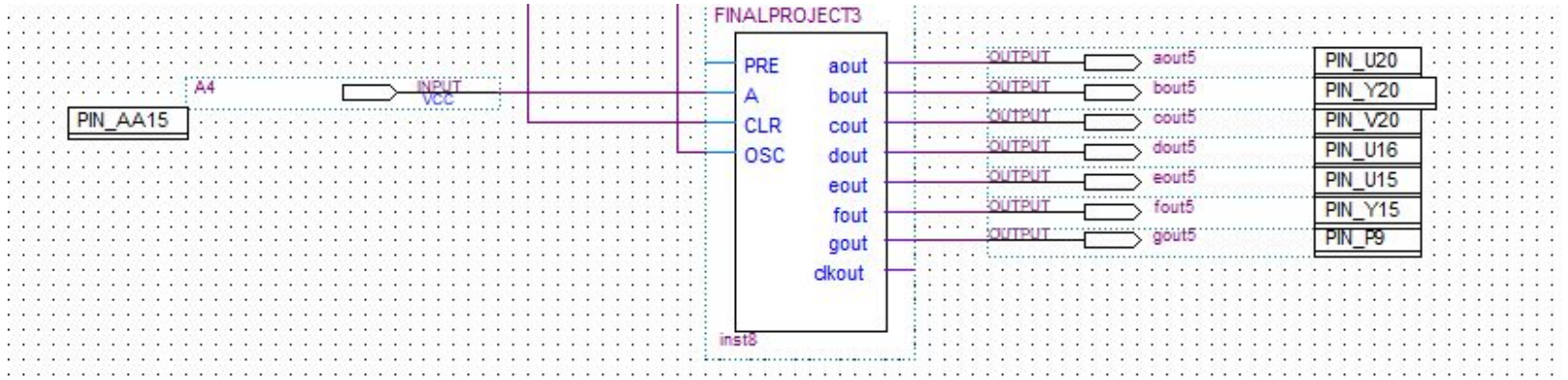


Schematic of Fifth Symbol Block:



This schematic is the same as the third symbol block as the range of my clock for the "hours" portion is only from 0-9. With this range, the frequency can be divided with the 4-AND gate that is true when the binary value is 1010. The decoder is the same as the other symbol blocks, and the values of the counter are displayed on a seven-segment display from the DE0-CV board.

Schematic of Fifth Symbol Block (condensed):



Screenshot of Compilation of Final Project:

Flow Status	Successful - Fri Nov 27 08:42:37 2020
Quartus II 64-Bit Version	14.1.0 Build 186 12/03/2014 SJ Web Edition
Revision Name	elevator
Top-level Entity Name	finalfinalproject
Family	Cyclone V
Device	5CEBA4F23C7
Timing Models	Final
Logic utilization (in ALMs)	51 / 18,480 (< 1 %)
Total registers	46
Total pins	42 / 224 (19 %)
Total virtual pins	0
Total block memory bits	0 / 3,153,920 (0 %)
Total DSP Blocks	0 / 66 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 4 (0 %)
Total DLLs	0 / 4 (0 %)

Screenshot of Pin Planner of Final Project:

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew R
osclock	Input	PIN_M9	3B	B3B_N0	PIN_M9	2.5 V (default)		12mA (default)	
CLEAR	Input	PIN_M7	3A	B3A_N0	PIN_M7	2.5 V (default)		12mA (default)	
A4	Input	PIN_AA15	4A	B4A_N0	PIN_AA15	2.5 V (default)		12mA (default)	
A3	Input	PIN_T12	4A	B4A_N0	PIN_T12	2.5 V (default)		12mA (default)	
A2	Input	PIN_T13	4A	B4A_N0	PIN_T13	2.5 V (default)		12mA (default)	
A1	Input	PIN_V13	4A	B4A_N0	PIN_V13	2.5 V (default)		12mA (default)	
A	Input	PIN_U13	4A	B4A_N0	PIN_U13	2.5 V (default)		12mA (default)	
gout5	Output	PIN_P9	3B	B3B_N0	PIN_P9	2.5 V (default)		12mA (default)	1 (default)
gout4	Output	PIN_V19	4A	B4A_N0	PIN_V19	2.5 V (default)		12mA (default)	1 (default)
gout3	Output	PIN_AB21	4A	B4A_N0	PIN_AB21	2.5 V (default)		12mA (default)	1 (default)
gout2	Output	PIN_U22	4A	B4A_N0	PIN_U22	2.5 V (default)		12mA (default)	1 (default)
gout	Output	PIN_AA22	4A	B4A_N0	PIN_AA22	2.5 V (default)		12mA (default)	1 (default)
fout5	Output	PIN_Y15	4A	B4A_N0	PIN_Y15	2.5 V (default)		12mA (default)	1 (default)
fout4	Output	PIN_V18	4A	B4A_N0	PIN_V18	2.5 V (default)		12mA (default)	1 (default)
fout3	Output	PIN_AB22	4A	B4A_N0	PIN_AB22	2.5 V (default)		12mA (default)	1 (default)
fout2	Output	PIN_AA17	4A	B4A_N0	PIN_AA17	2.5 V (default)		12mA (default)	1 (default)
fout	Output	PIN_Y21	4A	B4A_N0	PIN_Y21	2.5 V (default)		12mA (default)	1 (default)
eout5	Output	PIN_U15	4A	B4A_N0	PIN_U15	2.5 V (default)		12mA (default)	1 (default)
eout4	Output	PIN_U17	4A	B4A_N0	PIN_U17	2.5 V (default)		12mA (default)	1 (default)
eout3	Output	PIN_V14	4A	B4A_N0	PIN_V14	2.5 V (default)		12mA (default)	1 (default)
eout2	Output	PIN_AB18	4A	B4A_N0	PIN_AB18	2.5 V (default)		12mA (default)	1 (default)
eout	Output	PIN_Y22	4A	B4A_N0	PIN_Y22	2.5 V (default)		12mA (default)	1 (default)
dout5	Output	PIN_U16	4A	B4A_N0	PIN_U16	2.5 V (default)		12mA (default)	1 (default)
dout4	Output	PIN_V16	4A	B4A_N0	PIN_V16	2.5 V (default)		12mA (default)	1 (default)
dout3	Output	PIN_Y14	4A	B4A_N0	PIN_Y14	2.5 V (default)		12mA (default)	1 (default)
dout2	Output	PIN_AA18	4A	B4A_N0	PIN_AA18	2.5 V (default)		12mA (default)	1 (default)
dout	Output	PIN_W21	4A	B4A_N0	PIN_W21	2.5 V (default)		12mA (default)	1 (default)
cout5	Output	PIN_V20	4A	B4A_N0	PIN_V20	2.5 V (default)		12mA (default)	1 (default)
cout4	Output	PIN_Y17	4A	B4A_N0	PIN_Y17	2.5 V (default)		12mA (default)	1 (default)
cout3	Output	PIN_AA10	3B	B3B_N0	PIN_AA10	2.5 V (default)		12mA (default)	1 (default)
cout2	Output	PIN_AA19	4A	B4A_N0	PIN_AA19	2.5 V (default)		12mA (default)	1 (default)
cout	Output	PIN_W22	4A	B4A_N0	PIN_W22	2.5 V (default)		12mA (default)	1 (default)
bout5	Output	PIN_Y20	4A	B4A_N0	PIN_Y20	2.5 V (default)		12mA (default)	1 (default)
bout4	Output	PIN_W16	4A	B4A_N0	PIN_W16	2.5 V (default)		12mA (default)	1 (default)
bout3	Output	PIN_AB17	4A	B4A_N0	PIN_AB17	2.5 V (default)		12mA (default)	1 (default)
bout2	Output	PIN_AB20	4A	B4A_N0	PIN_AB20	2.5 V (default)		12mA (default)	1 (default)
bout	Output	PIN_V21	4A	B4A_N0	PIN_V21	2.5 V (default)		12mA (default)	1 (default)
aout5	Output	PIN_U20	4A	B4A_N0	PIN_U20	2.5 V (default)		12mA (default)	1 (default)
aout4	Output	PIN_Y16	4A	B4A_N0	PIN_Y16	2.5 V (default)		12mA (default)	1 (default)
aout3	Output	PIN_Y19	4A	B4A_N0	PIN_Y19	2.5 V (default)		12mA (default)	1 (default)
aout2	Output	PIN_AA20	4A	B4A_N0	PIN_AA20	2.5 V (default)		12mA (default)	1 (default)
aout	Output	PIN_U21	4A	B4A_N0	PIN_U21	2.5 V (default)		12mA (default)	1 (default)

6. Problems during Final Project and how they were solved:

There were some problems that appeared in the duration of creating this project. As it grew more complex, the problems were identifiable, but took some knowledge to solve them. One main problem that I had in the beginning was about the oscillator. At first, I tried putting multiple oscillators with varying frequencies, but it caused many errors while compiling the project. Also, none of the digits synced up perfectly with each other, which is the opposite of how a digital clock would function. Therefore, I realized that there should only be one oscillator input signal for the entire project. Once I solved that problem, more problems arised. With one oscillator signal, it took a lot of time to figure out how to make 09 transition to 10 together for an example. I also realized that I needed to use the clear function to automatically reset each digit once it reaches its maximum range. For example, when the clock is at 09, the "9" must reset to zero while the "0" counts up for the first time. This problem was solved with a 4 input AND gate that uses the outputs of the 4 counters. However, this output was for the next clock input and the clear function of the counters, and there also needs to be a manual clear input for the user. Connecting an input for the manual clear in the same wire as the clock output will cause an error within the compilation, so it took some time for me to figure out the solution. Eventually, this was solved with the use of a 2-input AND gate, so the function clears when the output is true or by the manual input of the user. The last problem I had was figuring out how to make the preset function using the logic I knew, which I could not figure out. This is only because if I plugged an input to the 4 presets of the counters, it messed up the counter for some unknown reason. For example, the "tens" digit for the seconds would count up by 2 everytime the clock input is on instead of one, and I did not know why the preset function would change the way the counter goes up. This problem was weird because although the preset function goes to the maximum range, there were some random side effects that messed up the counter for all the digits. Some digits went past their maximum range, counted up by 2 instead of 1, and began at sometime other than 0:00:00. All of these problems were due to the preset function being active, so I left it alone because this was a problem I could not fix. These problems took the most time to solve and were the most difficult for me during the creation of this project. Even though the solutions are very simple, it is the thought of how to implement these solutions that are difficult.

7. Interesting Information About the Final Project

- Some fun facts about this project:
- There are a total of 7 inputs and 35 outputs for this project
- All of the outputs are assigned to their own segment, as there are 5 seven-segment displays used with 7 segments in each.
- There are 5 4-bit counters that are assigned to the 5 seven-segment displays, with ranges of 0-5 and 0-9.
- Only one oscillator input is used for this whole circuit, where it starts at 1 Hz and gets divided 5 times from each symbol block.
- The part that took the longest to make in this final project was the clear function and the clock output to the next symbol block's clock input.
- I ran the simulation for an hour to verify that the hours counted up after 59:59 had passed.
- This project took the most effort and time compared to the other labs, but it would've been longer without the knowledge that I gained from them.

8. Conclusion of Final Project:

In this final project, I wanted to implement as many concepts that I have learned in this class and apply them to create something that we use everyday. With concepts such as flip flops, counters, memories, decoders, and oscillators, I was able to combine them all to implement a digital clock on the DE0-CV board. The digital clock is displayed onto 7 -segment displays that are provided on the board, and the oscillator creates an automatic clock signal so that the clock ticks up without the assistance of the user. With the oscillator, it creates the experience of an actual digital clock. Although this took a lot of time and effort to make, it was very satisfying to finish the project without errors. There were some problems that appeared during the creation of the project, but all of them were solved in the end. Although this was one of the hardest projects I have done related to computer engineering, it was really fun to simulate the project onto the board. This project turned from something that I wish I could do into reality, and it would not be possible without the knowledge that I had gained from the other labs. I am much more comfortable applying these projects onto the DE0-CV board now and in the future after finishing this final project. I hope that I am able to create more complex projects like this in the future, as this was very enjoyable and fun to make individually.

9. Link to Video with Explanation of Final Project:

https://youtu.be/oM7j1MH27i4