

Digital Design Lab

ECE 315

Lab/Project #5A
Counter and 8 bit Register

Group #7

Karysse Hay, Nikeem

Dunkelly-Allen and Dibyanshi Mishra

Zeinab Ramezani, TA

University of Miami

3/24/2022

Overview

The purpose of this lab was to design a 3-bit counter with a Reset and from that, create an 8-bit Register. The 3-Bit counter counts 0, 1, 2, 3, 4 and then returns to the number 0. It should be noted that all invalid states such as 5, 6 and 7 would return to zero. A counter was then designed using this 3-bit counter in combination with four D Flip flops and a Clock input. This allowed for the inputs to follow the sequence with the use of a Clock. The counter also required a Reset input which allowed for the counting sequence to reset back to 0. Counters are specially designed synchronous sequential circuits where the state of the counter is equal to the count held in the circuit by flip flops.

Using the knowledge and logic from the counter, the D Flip Flops were modified to give rise to a register. A register is a storage or memory component in a digital circuit. They are predetermined memory locations. For this lab, the register should be able to both store and output 8-bit binary inputs on the clock's rising edge with 'low' on the Load Input. As mentioned, the Register's output equals the input when the load is set to zero and the clock is on its rising edge. However, when the load is set to 1 and the clock is not on its rising edge, the output equals the previous input.

Equipment

- Quartus Prime Software
- AND gates
- OR gates
- NOT gates
- ModelSim
- Input Pins
- Output Pins
- Orthogonal Node
- Diagonal Node
- DE1-SoC Board

Description

From the lab instructions, the State Diagram for the 3-bit counter was given. This was used to create a truth table which consisted of Present State and Next State. From this table which is shown in Table 1, three boolean equations were found for A^+ , B^+ and C^+ . These were found to be: $A^+ = \overline{A} \overline{C}$, $B^+ = A \overline{C} \overline{B} + \overline{A} \overline{C} B$ and $C^+ = A \overline{C} B$.

After the logic was completed, a project using the Quartus Prime software was created. After

the project was launched, New Project Wizard was selected. The location and name of the file was chosen, an empty project was selected, and the flowing wizard was left blank. For the next selection, the DE1 SoC board was chosen. Under simulation the Tool Name was changed to simulations and the format to VHDL. After the window changed from the file menu, a new Block Diagram/Schematic option was chosen.

The logic diagram for the 3-bit circuit consisted of the logic for A^+ , B^+ and C^+ and can be seen in Figure 1. It consisted of three inputs (A, B and C), three outputs (Out_1, Out_2 and Out_3), three NOT gates, four AND gates and one OR gate. This circuit counts 0, 1, 2, 3, 4 and then returns to the number 0. It also ensured that any invalid input would return to zero as well. This logic diagram was compiled and simulated and a symbol block was created for it to be used in the next part of the lab.

The 3-Bit Up Counter was used to design a Counter with a Clock input, Reset input and four D flip flops. The clock input allowed for the numbers 0, 1, 2, 3 and 4 to be counted on every rising edge and the Reset input controlled whether or not the counter would be stopped and returned/ reset to 0. In each of the D Flip Flops, VCC was given to PRN and Reset was fed into CLRN. The counter was then compiled, simulated and downloaded to the board. Switches were assigned for Reset (PIN_AC12) and Clock (PIN_AB12). This allowed for the LEDs to be lit up and portray the 3-bit binary numbers (0, 1, 2, 3 and 4) in the sequence that was given in the lab.

For the 8-Bit Register, the D flip flops were modified where the Clock input was changed from just taking in the Clock input to taking in the AND of the Clock input and a new input named Load. Since the register is an 8-bit Register, it needed 8 inputs and D flip flops. This gave an output of 8 binary digits which can be read as one decimal number.

The 8-bit Register was compiled and simulated. The simulation showed that when the Reset input is 1 (Register is not clear), the Load is 0 and the Clock is on the rising edge, the output of the Register equals to the current input. Whereas when the Load is 1 and the clock is not on its rising edge, the output is equal to the previous input this can be seen in Figure 9.

Design Synthesis

Truth Tables:

PS	NS
CBA	$C^+B^+A^+$
000	001
001	010
010	011
011	100
100	000
101	000
110	000
111	000

Table 1: Truth Table for 3-bit counter

	CB 00	01	11	10
A 0	1	1	0	0
1	0	0	0	0

Table 2: K-map for A^+

	CB 00	01	11	10
A 0	0	1	0	0
1	1	0	0	0

Table 3 K-map for B^+

	CB 00	01	11	10
A 0	0	0	0	0
1	0	1	0	0

Table 4: K-map for C^+

Boolean Equations:

$$A^+ = \overline{A} \overline{C}$$

$$B^+ = A \overline{C} \overline{B} + \overline{A} \overline{C} B$$

$$C^+ = A \overline{C} B$$

Complete Logic Diagram

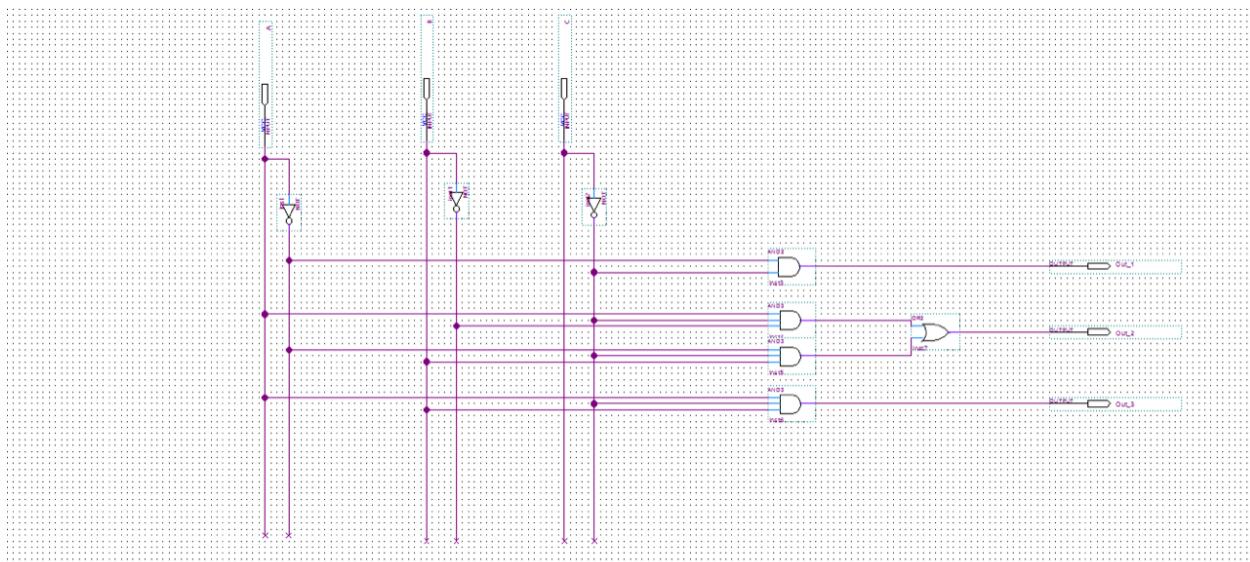


Figure 1: 3- Bit Up Counter Logic Diagram

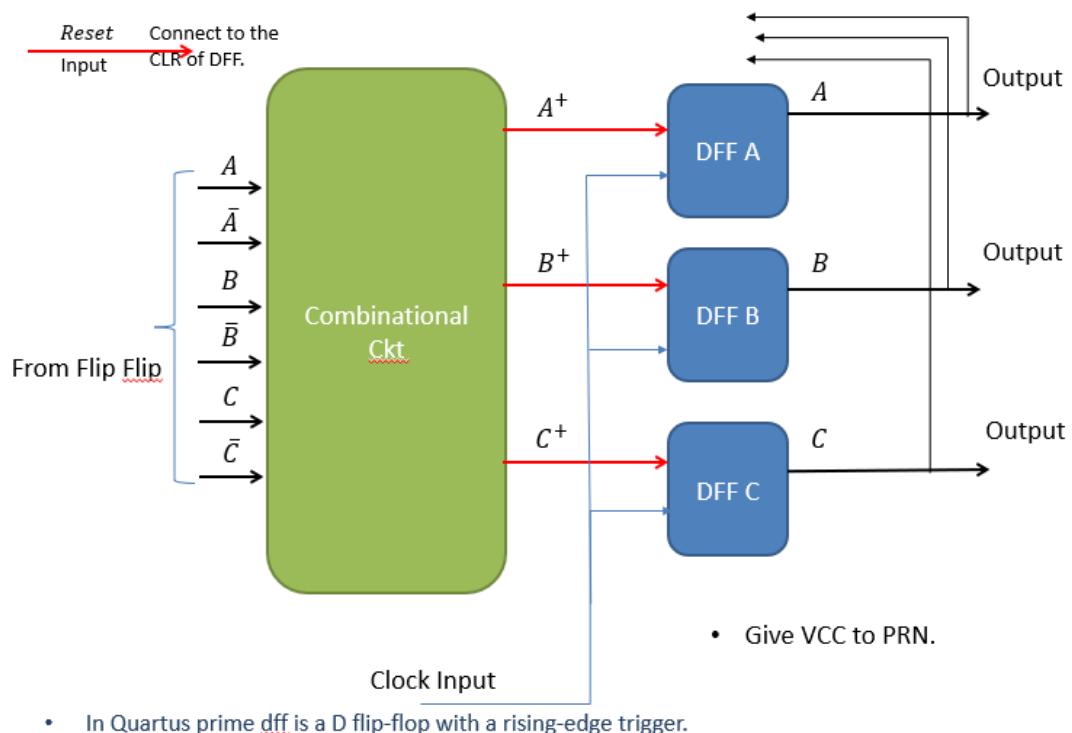


Figure 2: Block Diagram for Counter

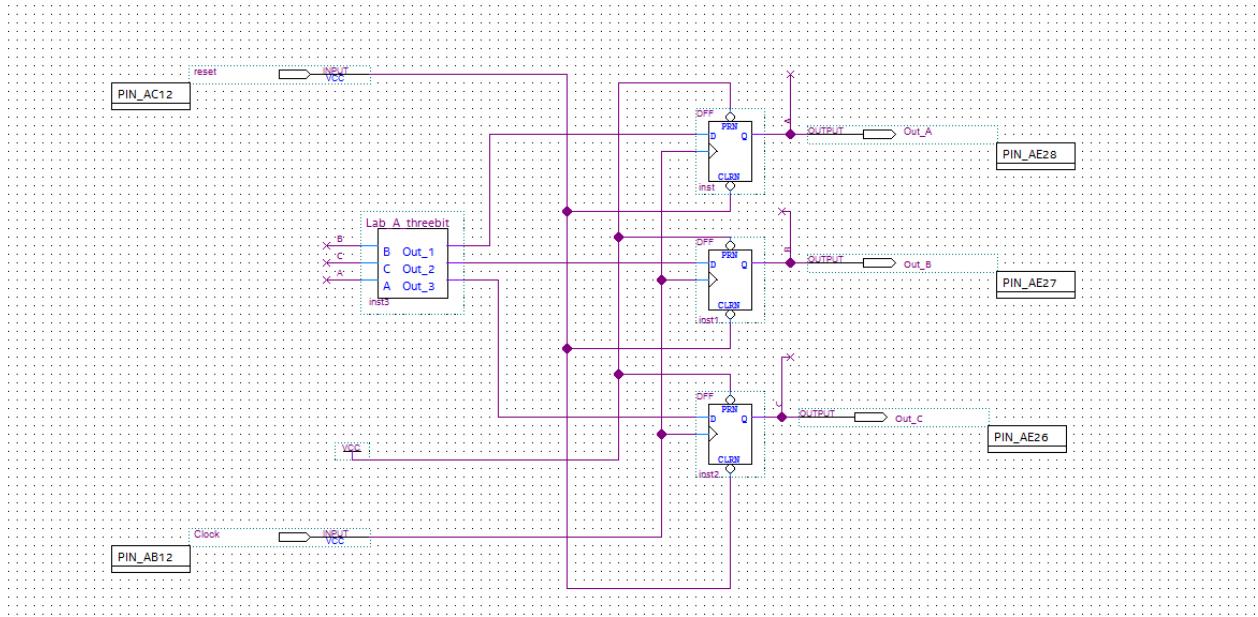


Figure 3: Counter Logic Diagram

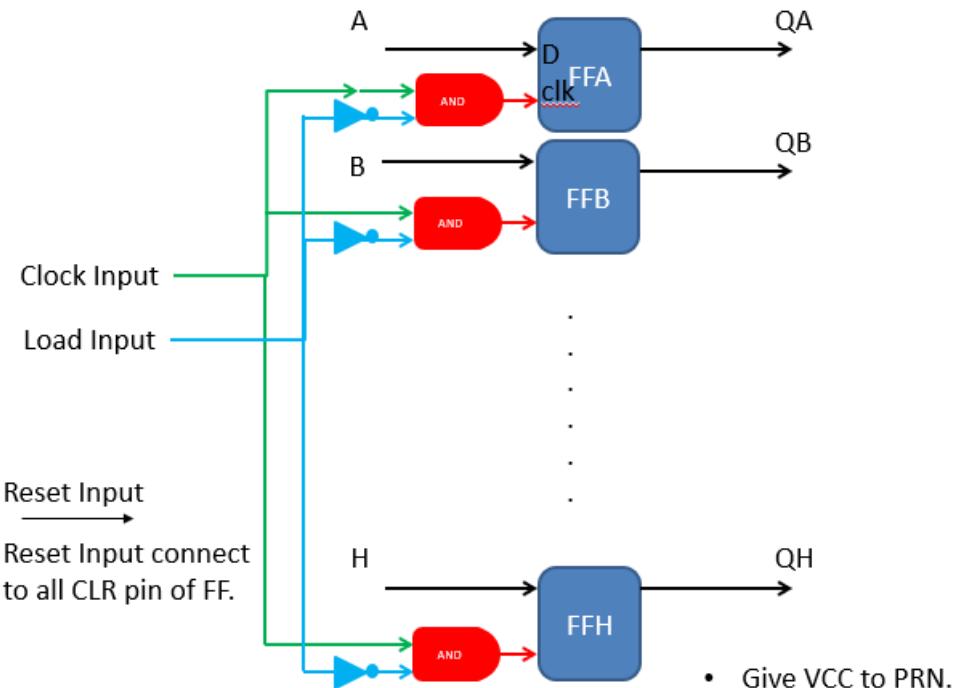


Figure 4: Block Diagram for 8-Bit Register

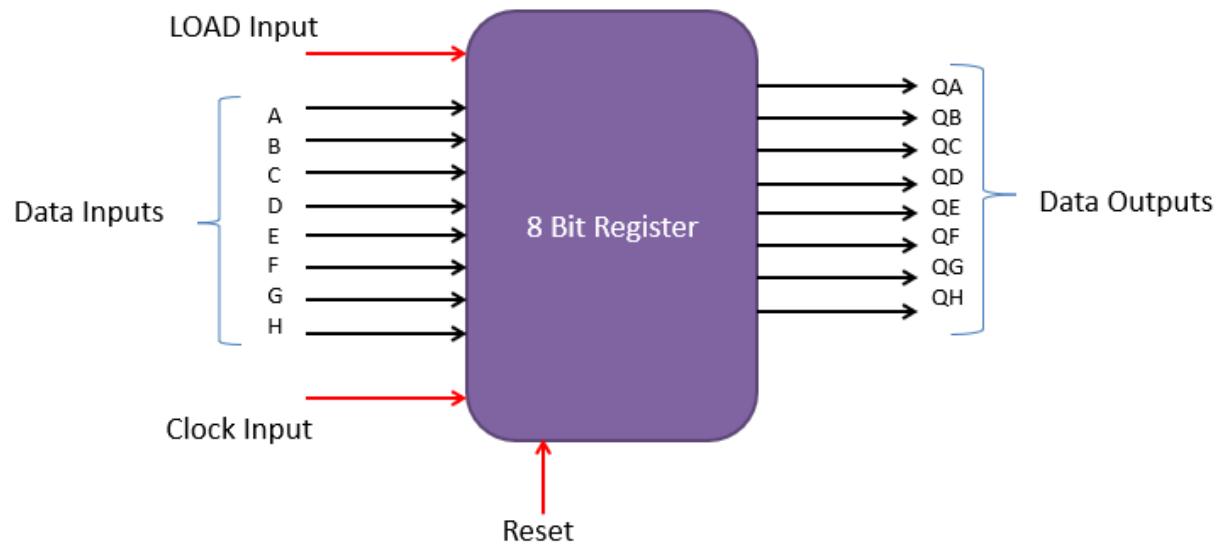


Figure 5: Block Symbol for 8-Bit Register

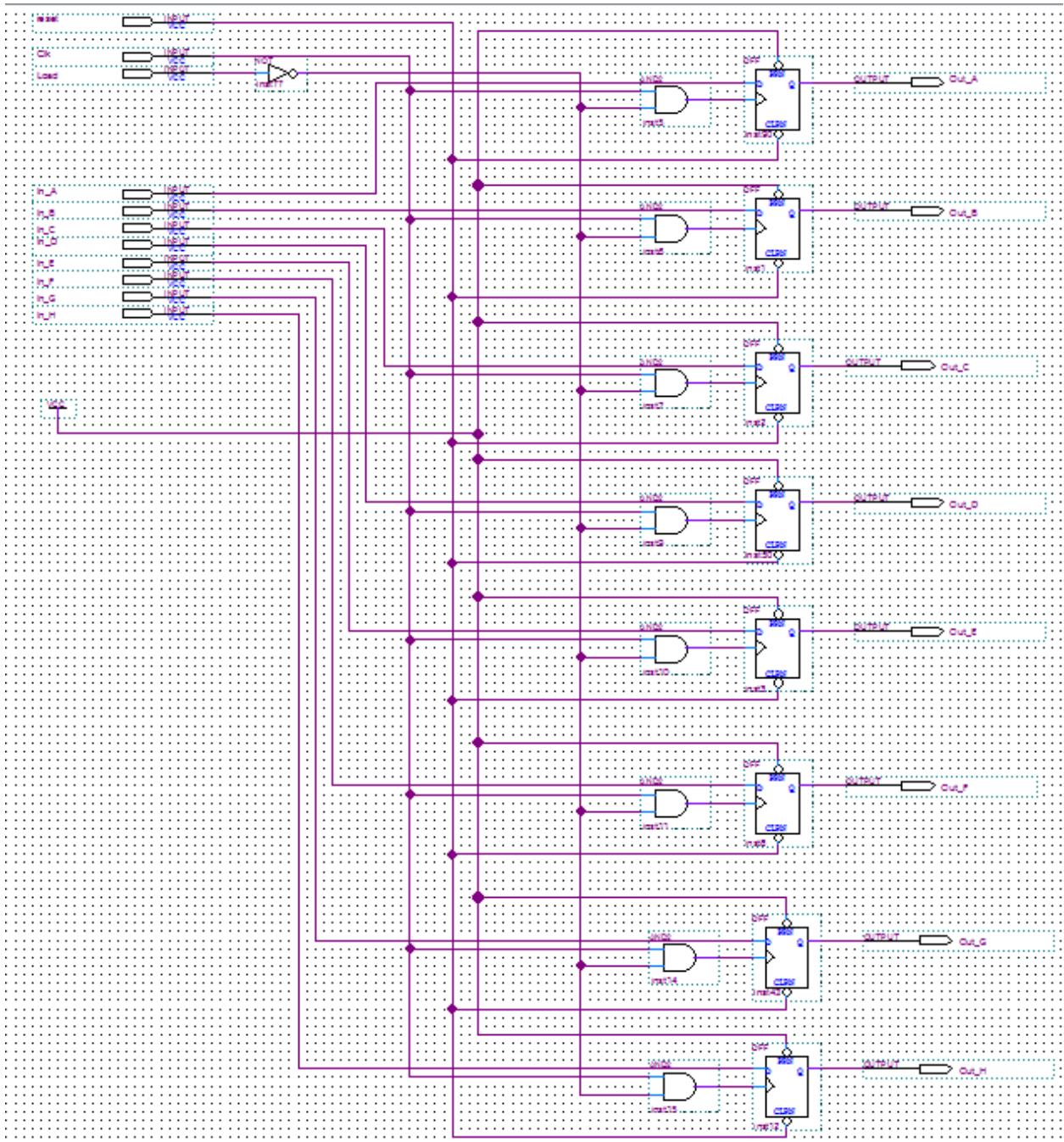


Figure 6: 8-Bit Register Logic Diagram

Results and Simulations

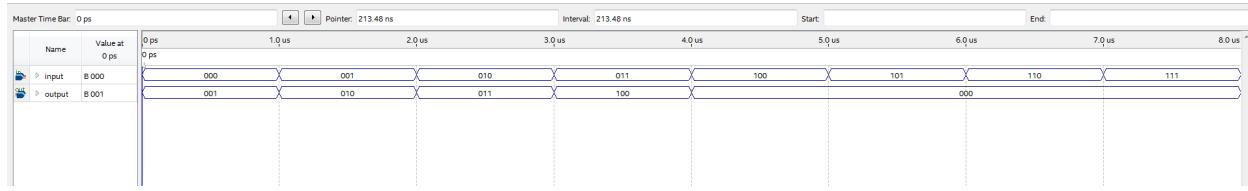


Figure 7: 3-Bit Up Counter Simulation

The 3-bit up counter counts 0, 1, 2, 3, 4 and then returns to the number 0. All states greater than 4 are invalid and would return to 0.

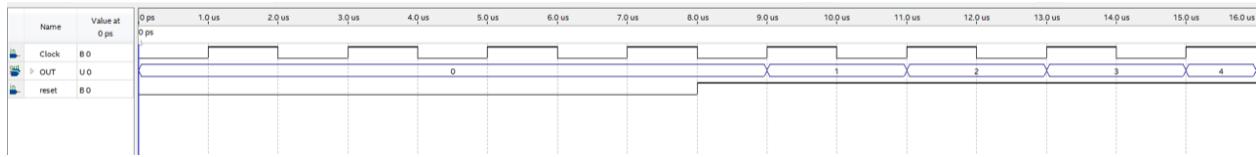


Figure 8: Counter Simulation

The counter was made in combination with the 3-bit up counter and D flip flops, where the counter will continue the sequence synonymous with the clock.

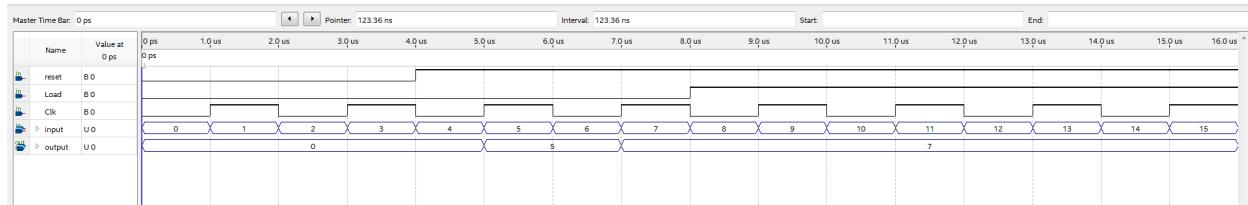


Figure 9: 8-Bit Register Simulation

The Register's output equals the input when the load is set to zero and the clock is on its rising edge which can be seen when the output changes from 0 to 5. However, when load is set to one the output equals previous input which can be seen when the input is 8 and the output is 8 and the output remains 7 (previous input).

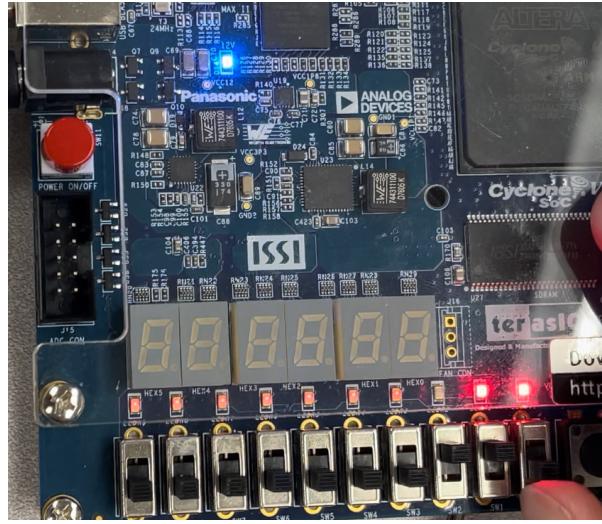


Figure 10: Counter downloaded of DE1-SoC board

The counter was downloaded to the board and the two rightmost switches were assigned to Reset and Clock. When the Clock switch was toggled, the LEDs would light up to represent the binary values for 0, 1, 2, 3 and 4.

Conclusion

To make a 3-bit up counter the truth table needed to be completed, then boolean equations derived from the K-map were implemented to create the logic diagram. The 3-bit up counter counts up from 0-4 and all states greater than 4 would return to 0. In order to create the counter the 3-bit up counter was modified with 4 D flip flops, a clock and a reset. The clock would work synonymous with the inputs 0-4 on the rising edge and the reset would give the user the option to stop the sequence and return to 0 to restart the sequence.

The 8-bit register was made using modified D flip flops that had an input of Load and the Clock. When the Reset input is 1 (Register is not clear), the Load is 0 and the Clock is on the rising edge, the output of the Register equals to the current input. Whereas when the Load is 1 and the clock is not on its rising edge, the output is equal to the previous input.

After downloading the counter program to the DE1-SoC board the counter was not counting up properly. After a few trials we figured out that you had to toggle the *clock* switch four times in

order to get the proper sequence, so every fourth toggle the correct number in the sequence would appear. Sometimes this would work and sometimes it would not. Even though our design and schematic were correct errors still are susceptible to occur. We learned how to create an up counter, and create and use a modified D flip flop. This should prove useful in the next lab since it is building upon this one.