CSC327
Shyang Kao

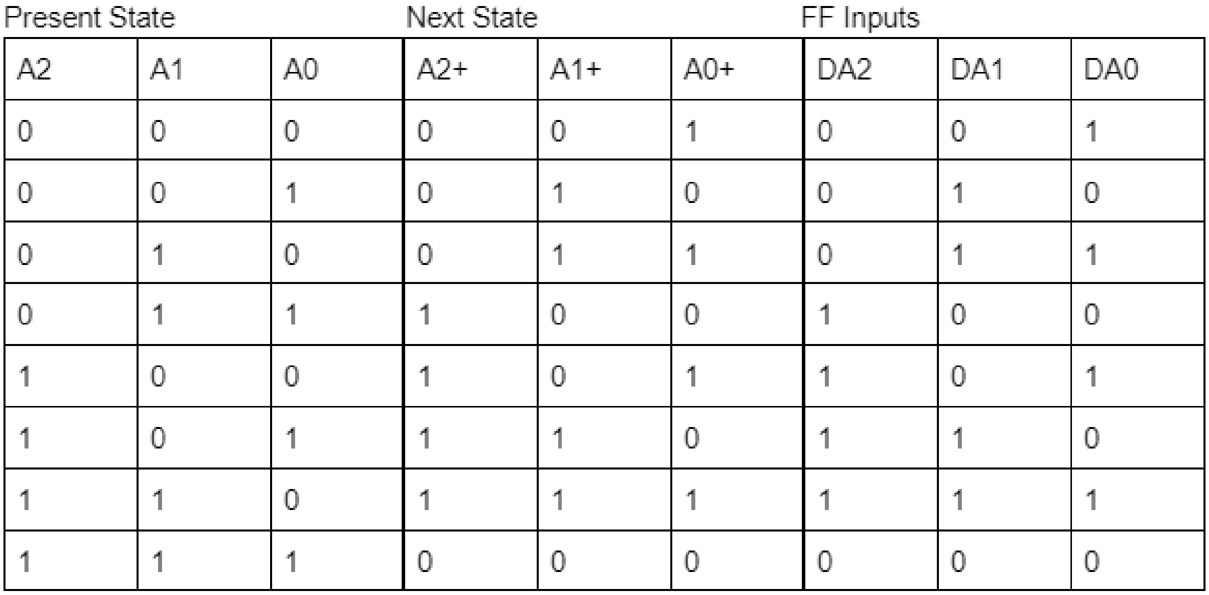
Lab8

Experiment Description

The goal of this lab is to create and test various binary counters. This lab has two purposes. The first step in this experiment is to create a Binary Counter that counts from 0 to 7 and then repeats. Designing the UP Logic and DOWN Logic for a BCD counter circuit is the second component of this lab.

Experiment Procedure

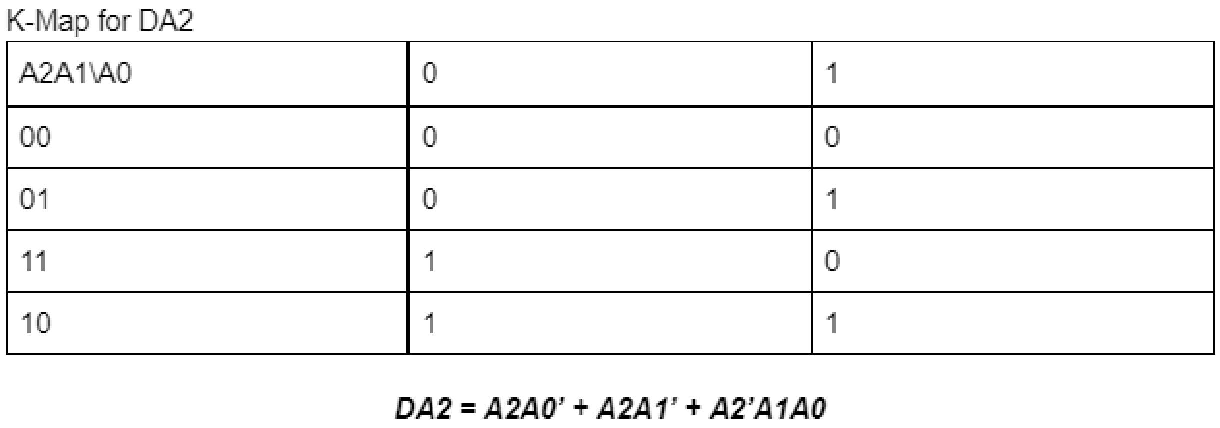
It is necessary to first establish a transition table before beginning to develop the binary counter that counts from 0 to 7 and repeats in part 1 of the lab. The current state, next state, and flip flop binary values are shown in figure 1's transition table. A DFF and the DFF equation (Qn+1 = D) are utilized for the flip-flop.



Present State			Next State			FF Inputs		
A2	A1	A0	A2+	A1+	A0+	DA2	DA1	DA0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

Figure 1 - Transition Table for Binary Counter

With the transition table, K-maps for DA2, DA1, and DA0 can be created. These K-maps are shown in figures 2, 3, and 4, as well as the equations derived from the K-maps.

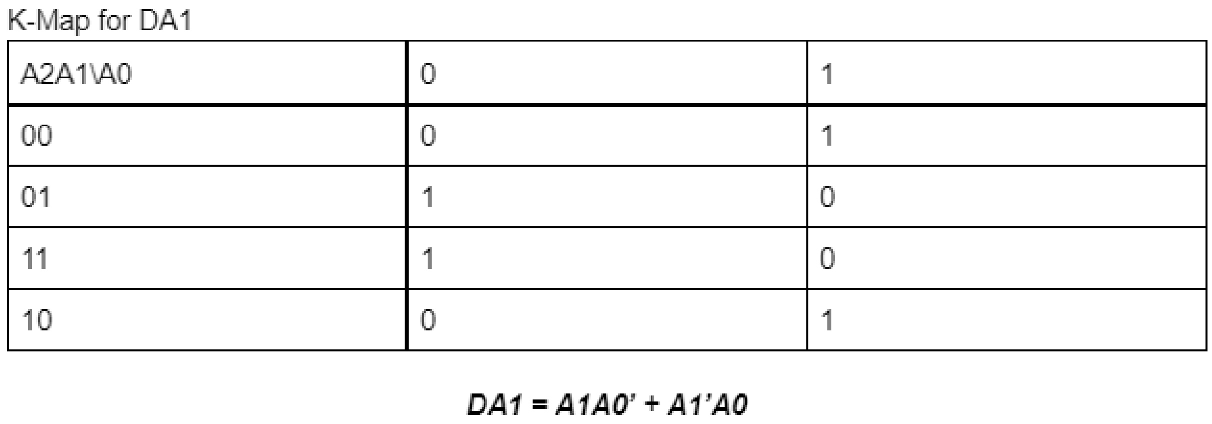


K-Map for DA2

A2A1\A0	0	1
00	0	0
01	0	1
11	1	0
10	1	1

$DA2 = A2A0' + A2A1' + A2'A1A0$

,Figure 2 - K-map and Equation for DA2

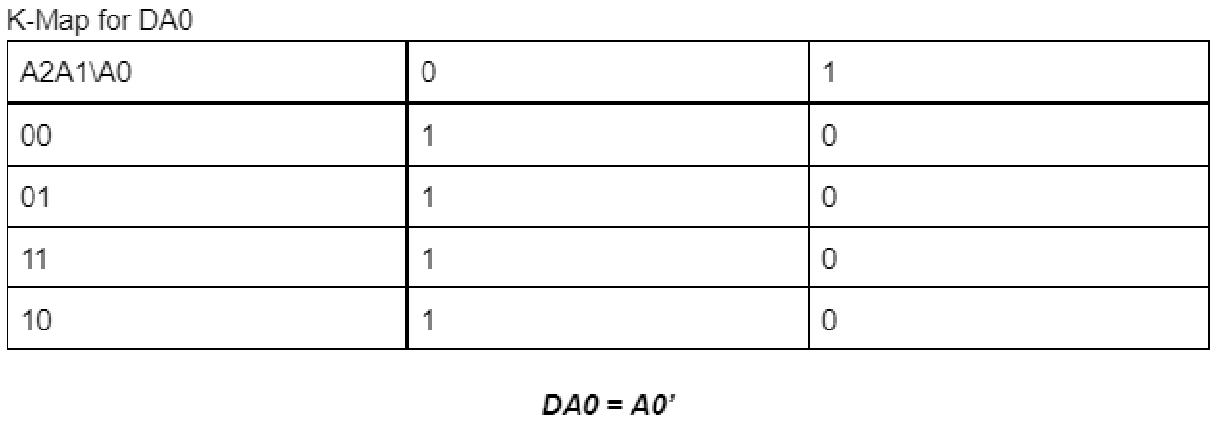


K-Map for DA1

A2A1\A0	0	1
00	0	1
01	1	0
11	1	0
10	0	1

$DA1 = A1A0' + A1'A0$

Figure 3 - K-map and Equation for DA1



K-Map for DA0

A2A1\A0	0	1
00	1	0
01	1	0
11	1	0
10	1	0

$DA0 = A0'$

Figure 4 - K-map and Equation for DA0

With the equations derived from the K-maps, a digital circuit can be constructed for the counter. Figure 5 is the circuit resulting from the equations.

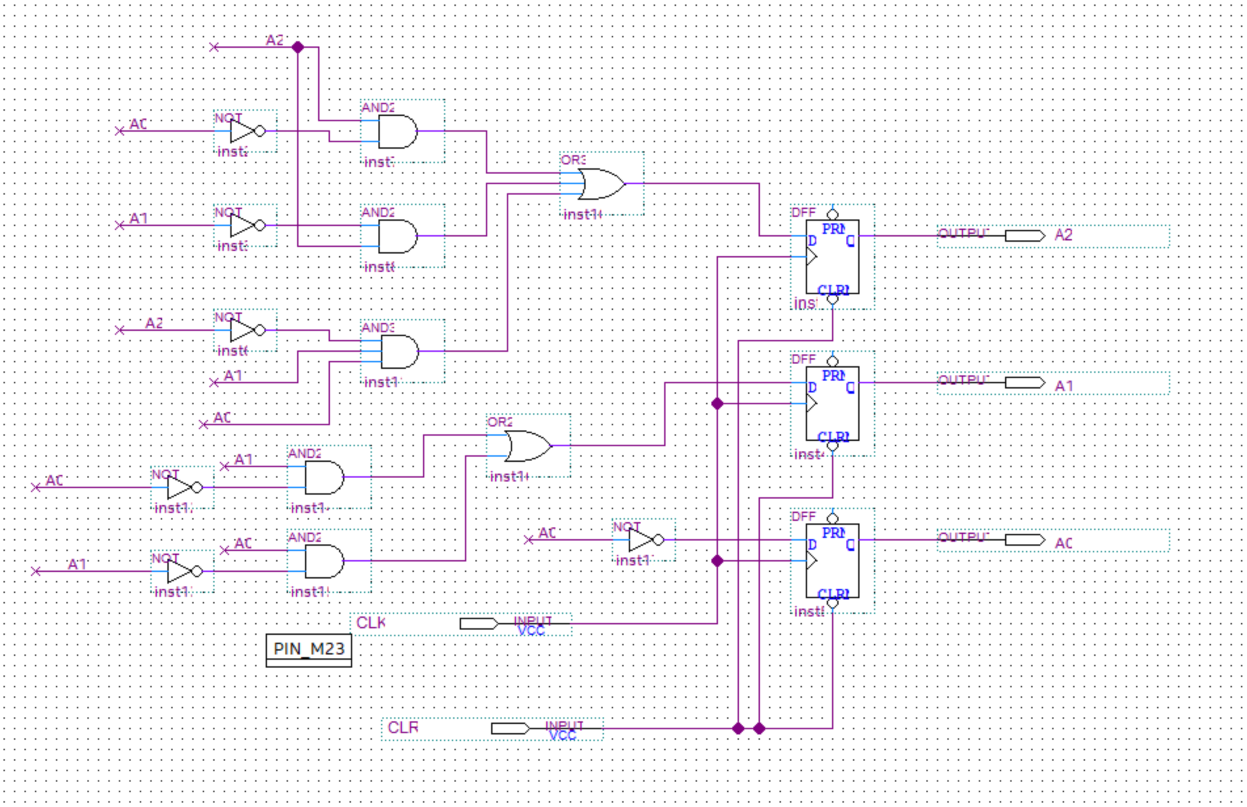


Figure 5 - Circuit for 0-7 Binary Counter

With the circuit constructed, a ModelSim simulation can be done. To run the ModelSim simulation, the circuit is converted to VHDL code. Figure 6 shows the resulting ModelSim simulation, and below is the ModelSim code that was used to generate the simulation.

CODE:

```
add wave CLR CLK A2 A1 A0
force CLR '0' 0 ns, '1' 5ns
force -freeze CLK 0 1, 1 {20 ns} -r 40ns
run 300 ns
```

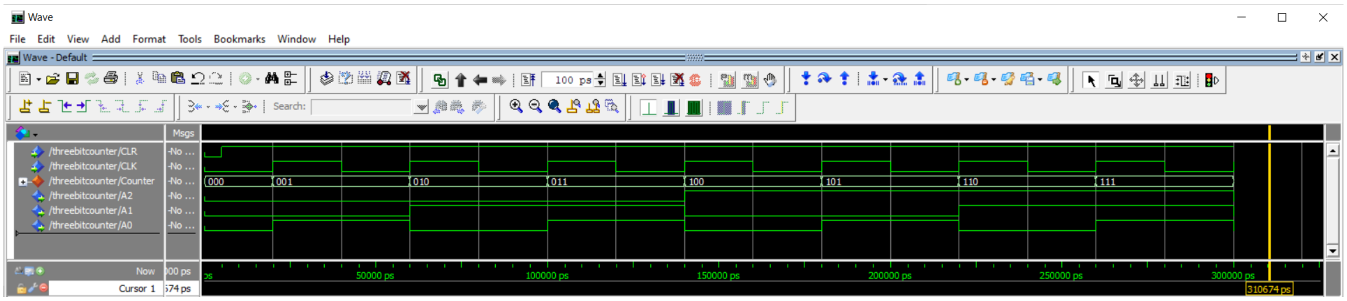
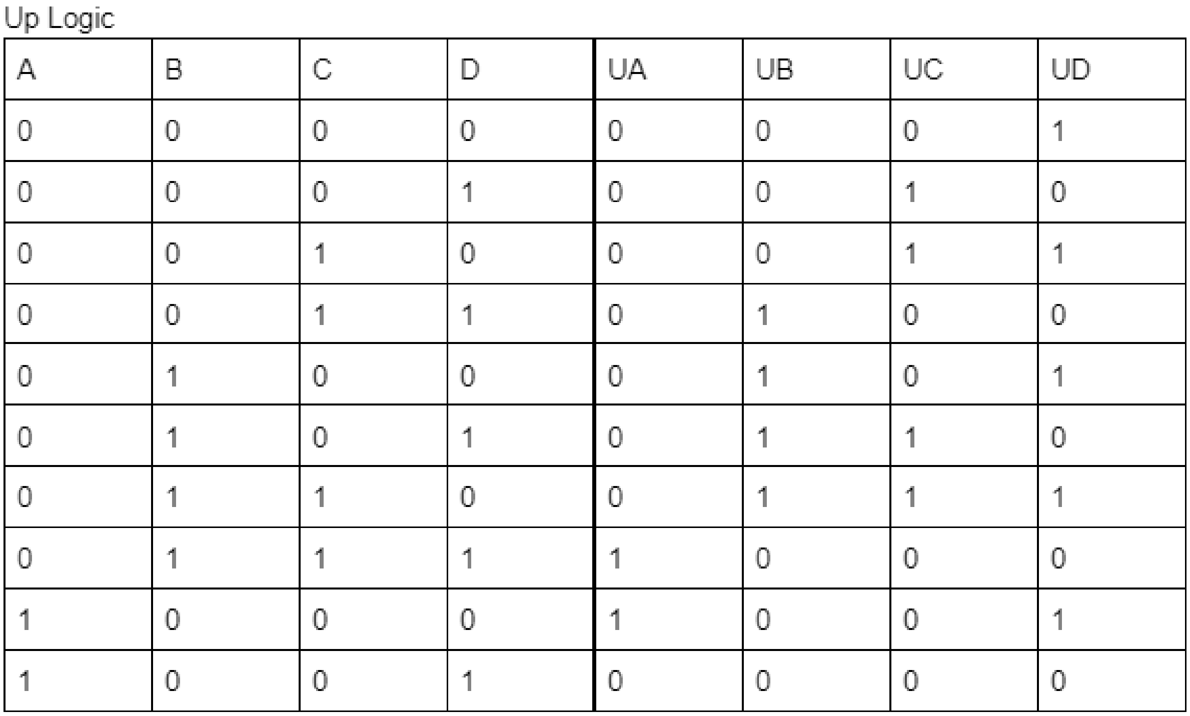


Figure 6 - ModelSim Simulation for 0-7 Binary Counter

With part 1 of the lab completed, part 2 requires us to design and implement the UP and DOWN logic for a synchronous BCD counter circuit. To begin, the UP logic is designed first. A truth table for the UP logic is shown in figure 7.

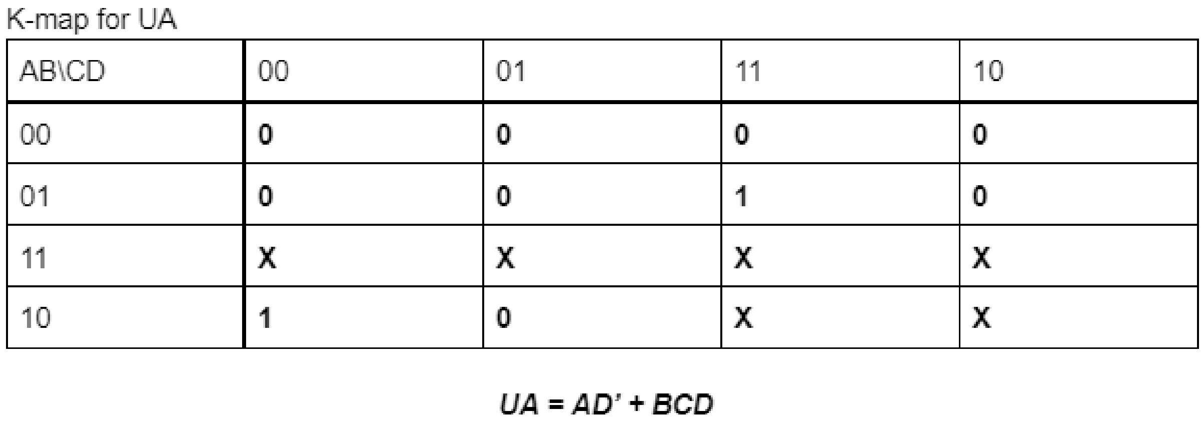


Up Logic

A	B	C	D	UA	UB	UC	UD
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

Figure 7 - Truth Table for UP Logic

With the truth table, K-maps for UA, UB, UC, and UD can be created. These K-maps are shown in figures 8, 9, 10 and 11, as well as the equations derived from the K-maps.

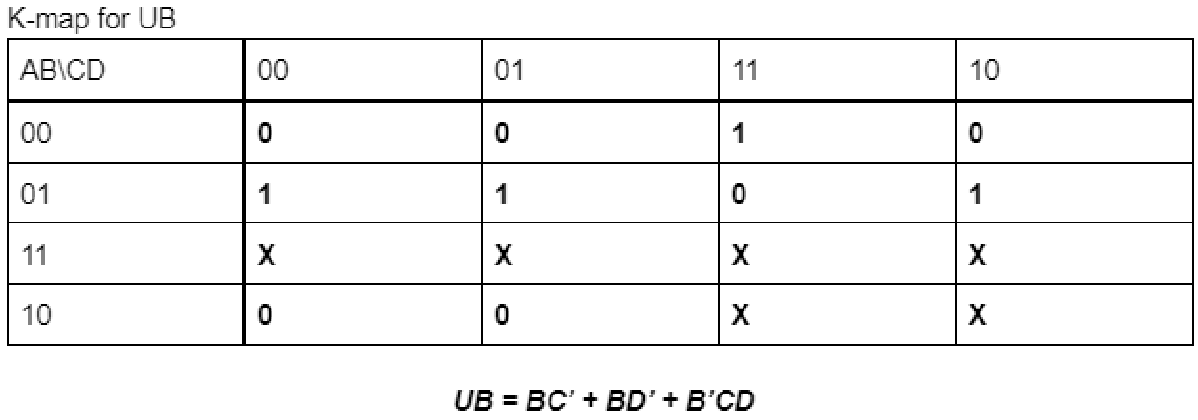


K-map for UA

AB\CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	1	0	X	X

$UA = AD' + BCD$

Figure 8 - K-map and Equation for UA

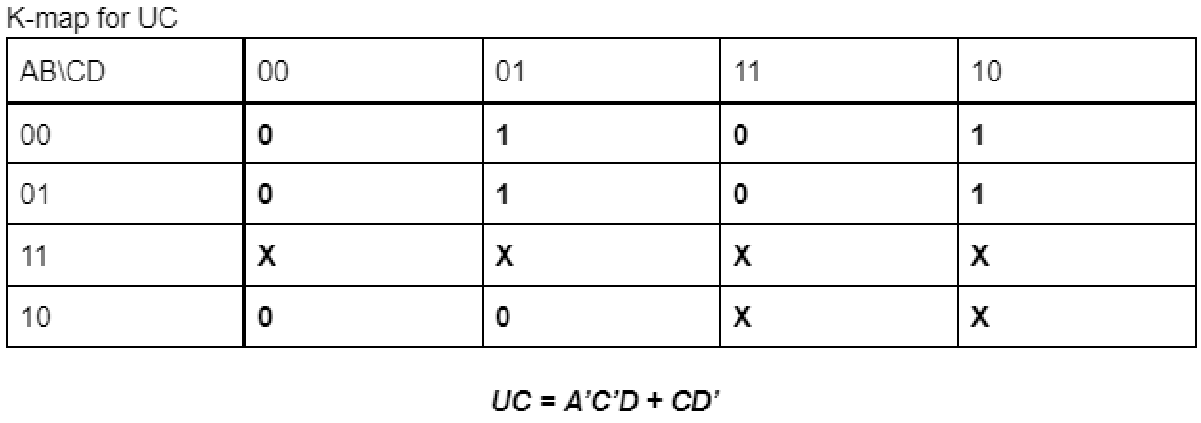


K-map for UB

AB\CD	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	X	X	X	X
10	0	0	X	X

$UB = BC' + BD' + B'CD$

Figure 9 - K-map and Equation for UB

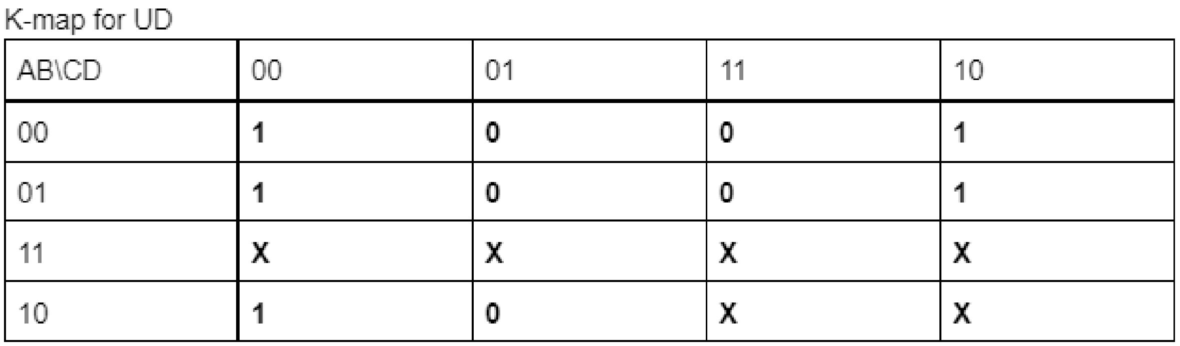


K-map for UC

AB\CD	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	X	X	X	X
10	0	0	X	X

$UC = A'C'D + CD'$

Figure 10 - K-map and Equation for UC



K-map for UD

AB\CD	00	01	11	10
00	**1**	**0**	**0**	**1**
01	**1**	**0**	**0**	**1**
11	**X**	**X**	**X**	**X**
10	**1**	**0**	**X**	**X**

UD= D'

Figure 11 - K-map and Equation for UD

With the equations derived from the K-maps, a digital circuit can be constructed for the UP logic. Figure 12 is the resulting circuit.

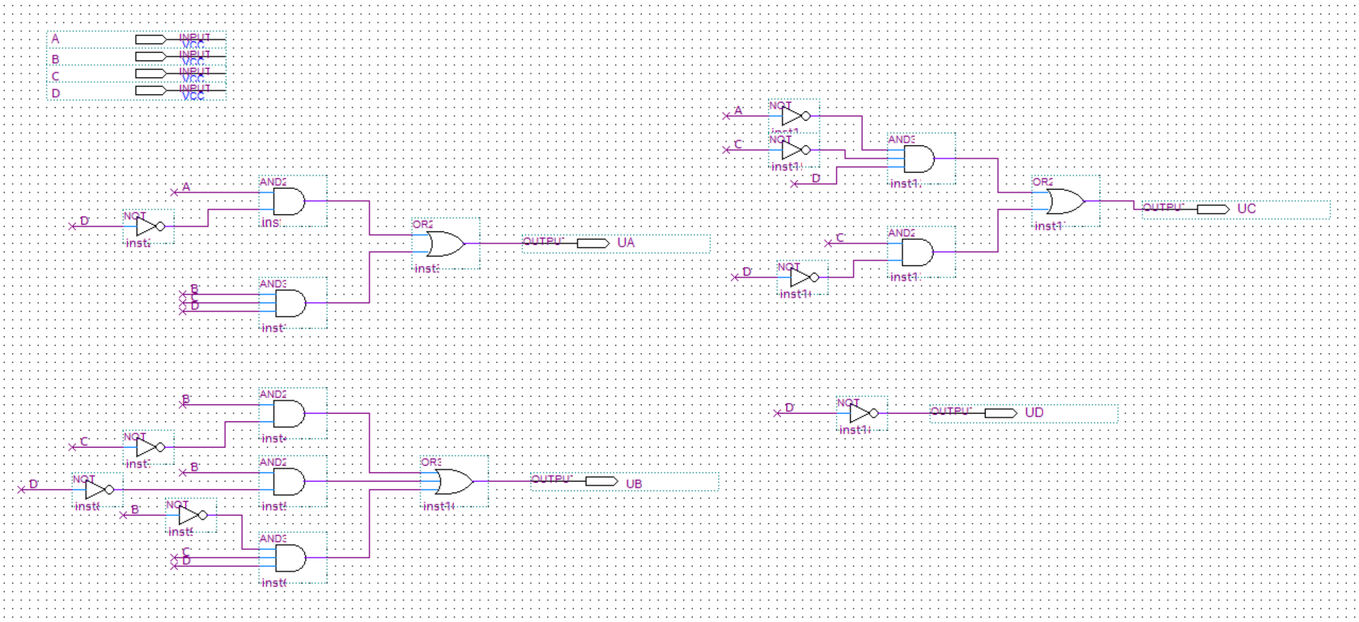
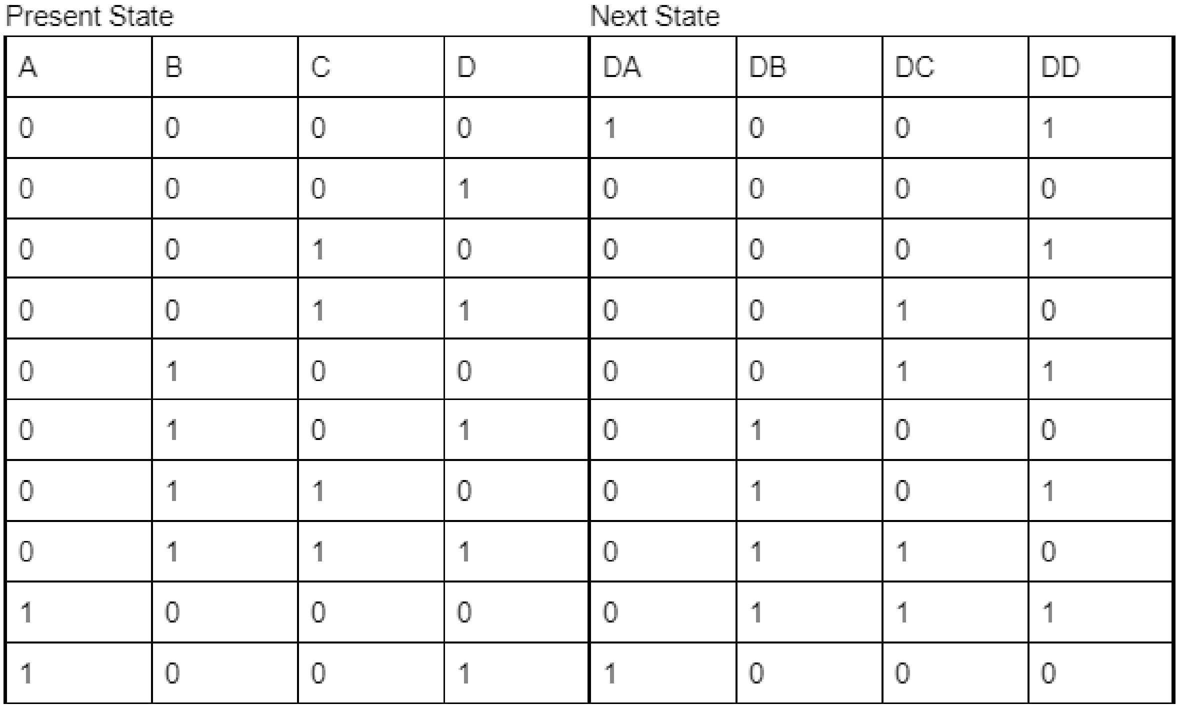


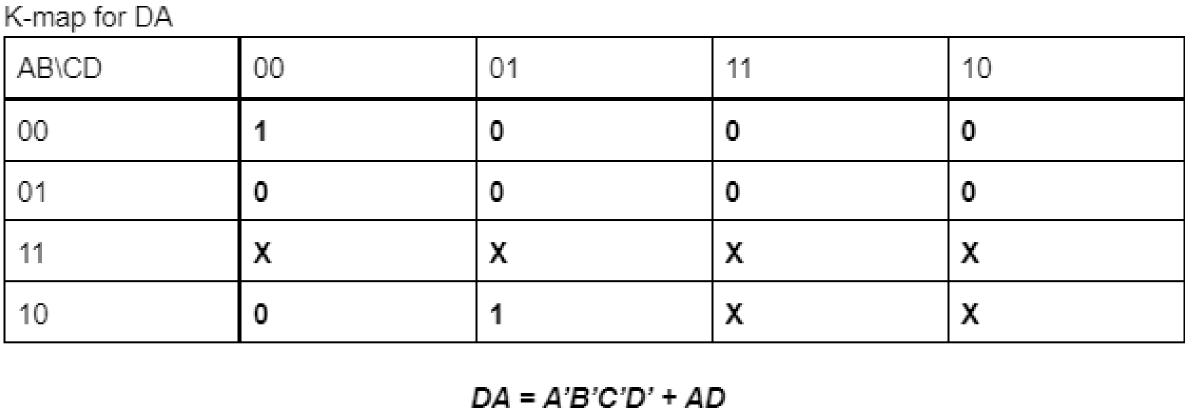
Figure 12 - Circuit for UP Logic

The same steps that are used to create the UP logic are also used to create the DOWN logic. Figure 13 is the truth table for the DOWN logic, and figures 14-17 are the K-maps and equations for DA, DB, DC, and DD.



Present State				Next State			
A	B	C	D	DA	DB	DC	DD
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	1
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	1
0	1	1	1	0	1	1	0
1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	0

Figure 13 - Truth Table for DOWN Logic

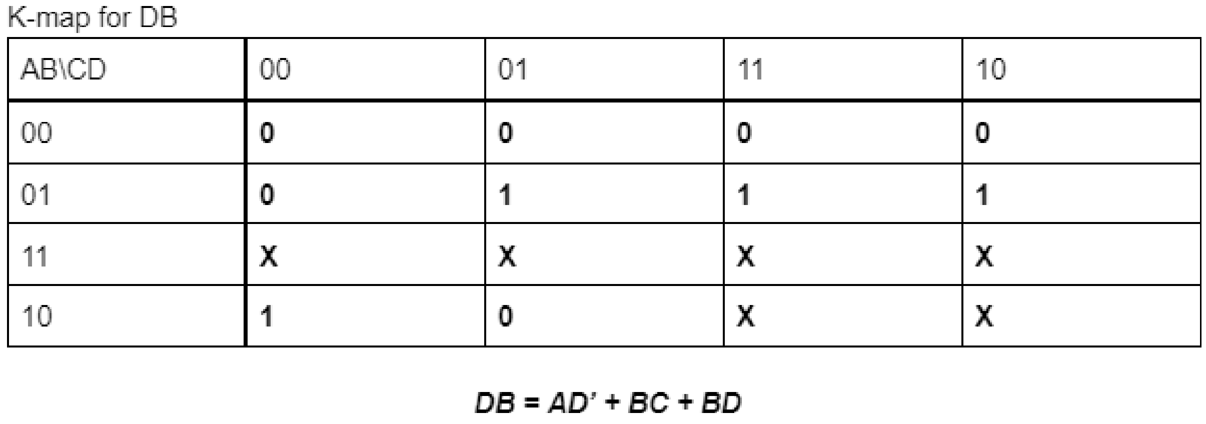


K-map for DA

AB\CD	00	01	11	10
00	**1**	**0**	**0**	**0**
01	**0**	**0**	**0**	**0**
11	**X**	**X**	**X**	**X**
10	**0**	**1**	**X**	**X**

$$DA = A'B'C'D' + AD$$

Figure 14 - K-map and Equation for DA

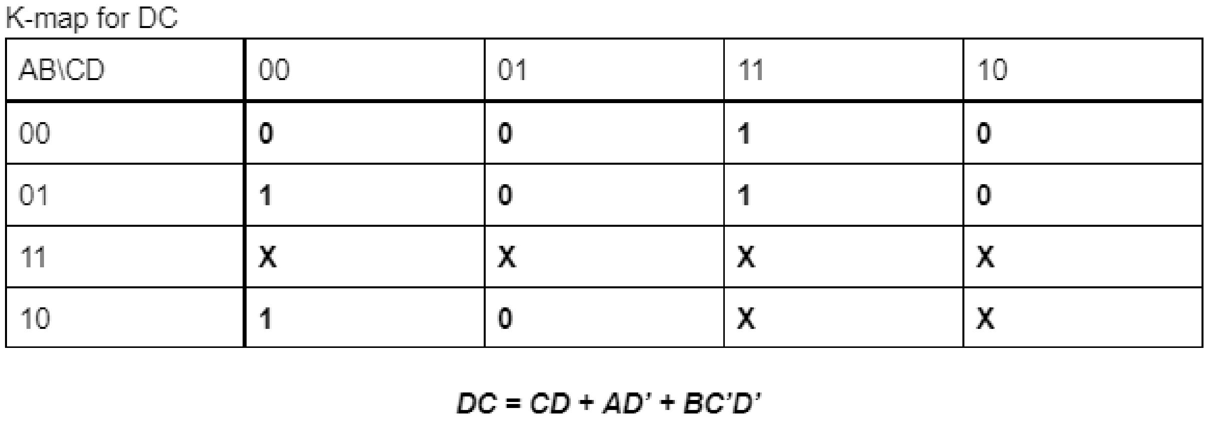


K-map for DB

AB\CD	00	01	11	10
00	**0**	**0**	**0**	**0**
01	**0**	**1**	**1**	**1**
11	**X**	**X**	**X**	**X**
10	**1**	**0**	**X**	**X**

DB = AD' + BC + BD

Figure 15 - K-map and Equation for DB

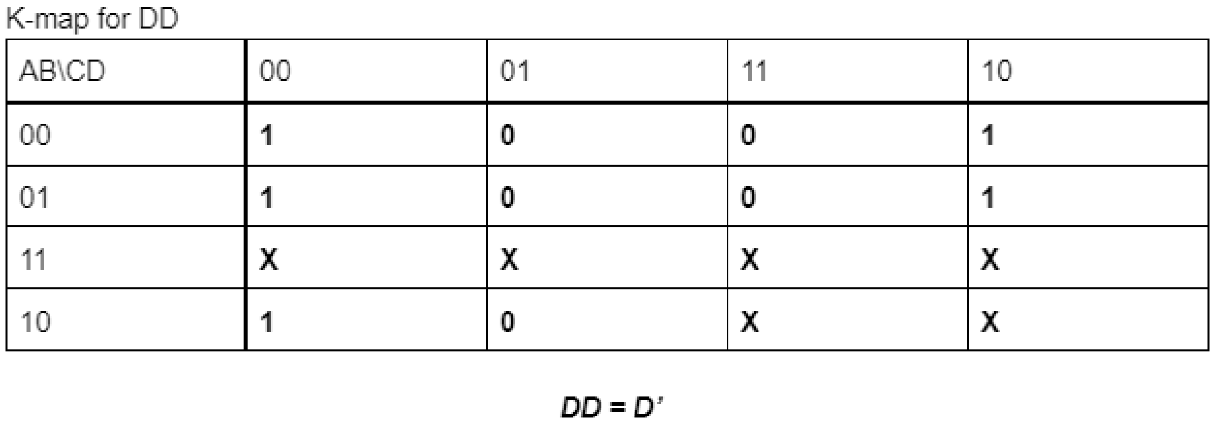


K-map for DC

AB\CD	00	01	11	10
00	**0**	**0**	**1**	**0**
01	**1**	**0**	**1**	**0**
11	**X**	**X**	**X**	**X**
10	**1**	**0**	**X**	**X**

DC = CD + AD' + BC'D'

Figure 16 - K-map and Equation for DC



K-map for DD

AB\CD	00	01	11	10
00	**1**	**0**	**0**	**1**
01	**1**	**0**	**0**	**1**
11	**X**	**X**	**X**	**X**
10	**1**	**0**	**X**	**X**

DD = D'

Figure 17 - K-map and Equation for DD

With the equations derived from the K-maps, a digital circuit can be constructed for the DOWN logic. Figure 18 is the resulting circuit.

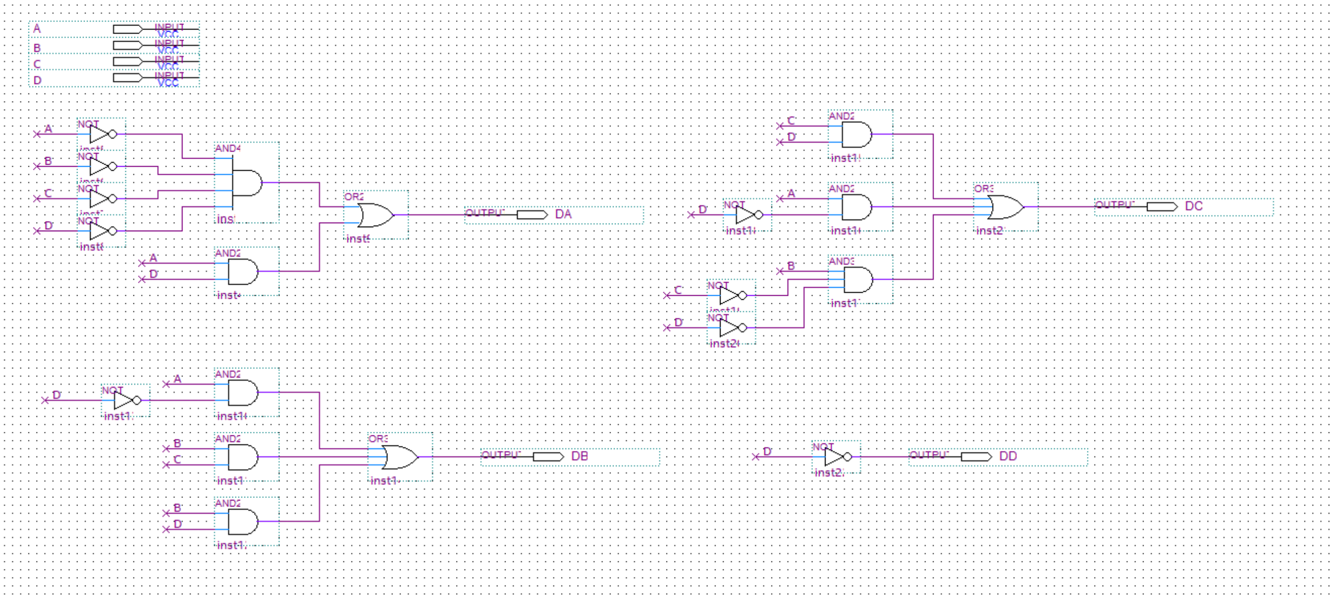


Figure 18 - Circuit for DOWN Logic

With both the UP and DOWN logic complete, symbol files can be made for the logics and they can be implemented in the provided counter circuit. The counter circuit with the UP and DOWN logic symbol files is shown in figure 19.

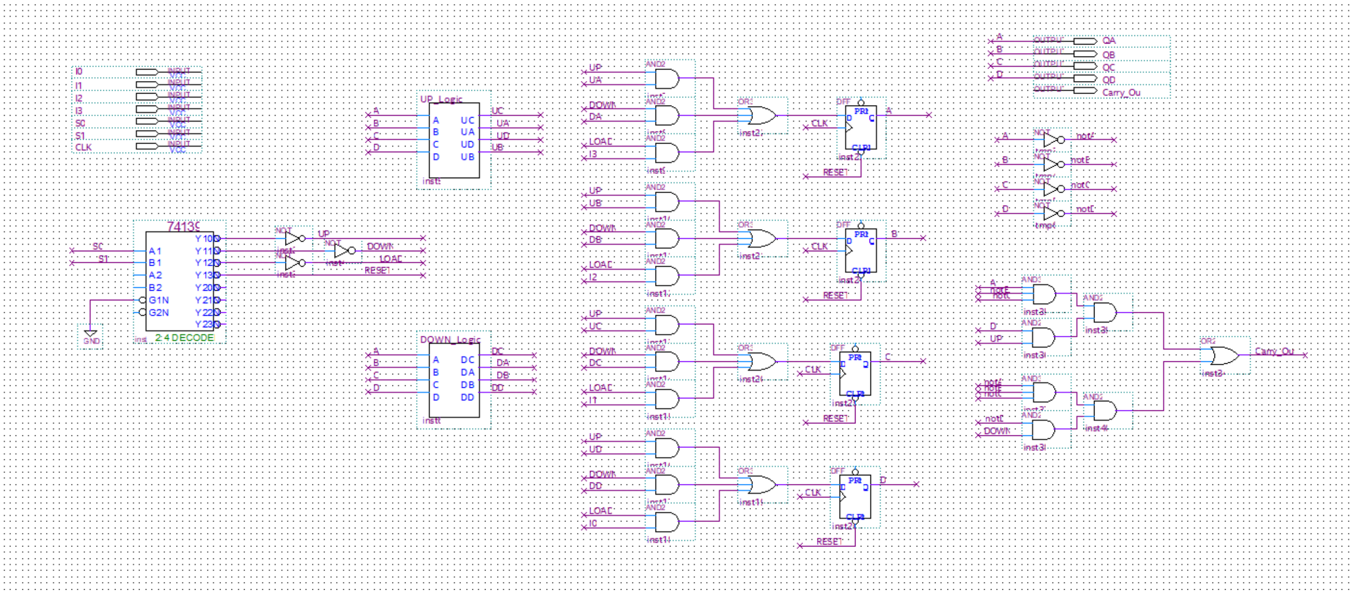
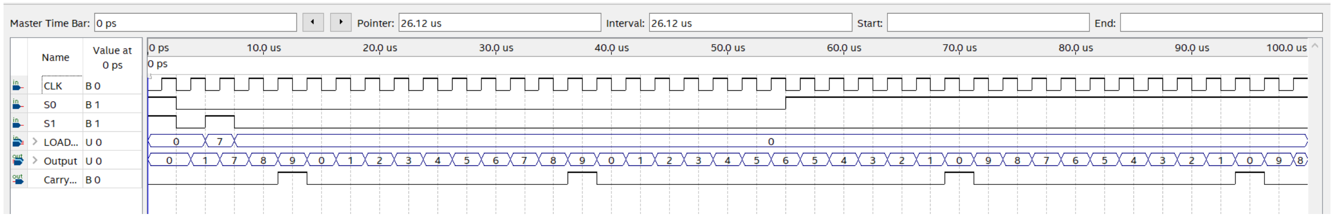


Figure 19 - Counter Circuit with UP and DOWN Logic

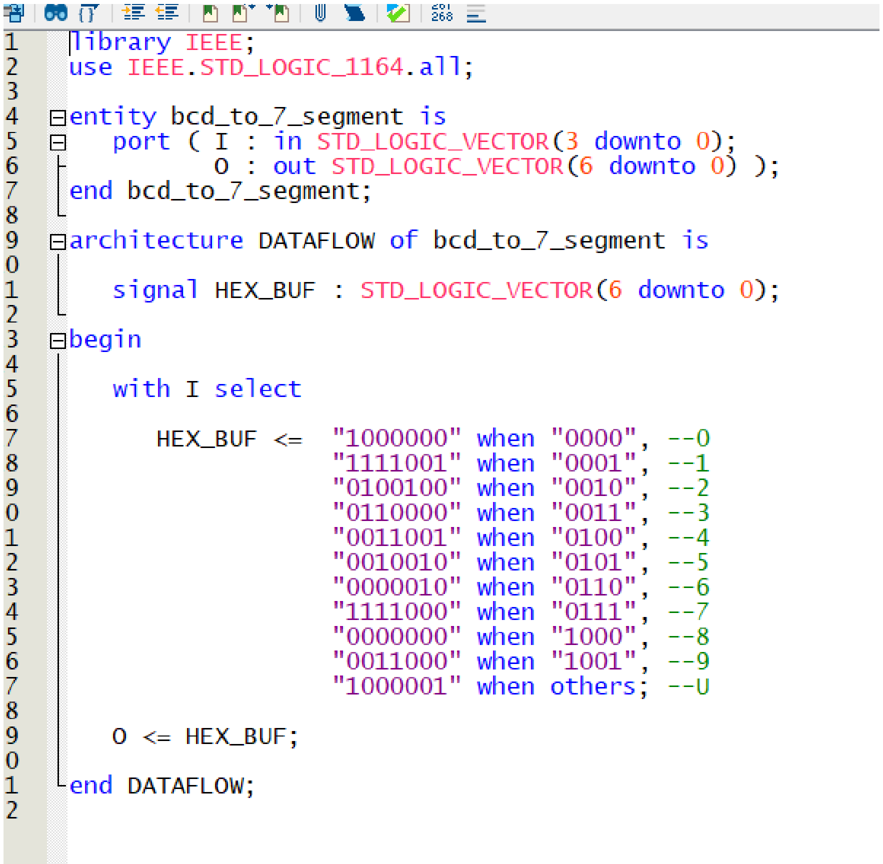
With the counter circuit complete, a simulation can be done to ensure that the circuit is working as intended. Figure 20 is the simulation of the counter circuit in figure 19.


Master Time Bar: 0 ps
Pointer: 26.12 us
Interval: 26.12 us
Start:
End:

Name	Value at 0 ps
CLK	B 0
S0	B 1
S1	B 1
LOAD...	U 0
Output	U 0
Carry...	B 0

Figure 20 - Simulation of Counter Circuit

The DE2 board is requested to convert the BCD values so that they may be shown on a seven-segment display in order to implement the final design. Figure 21 shows a BCD-to-seven-segment converter that is included in the lab file. The finished circuit, shown in Figure 22, incorporates the counter circuit from Figure 19 as well as the BCD-to-seven-segment converter.



```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity bcd_to_7_segment is
    port ( I : in STD_LOGIC_VECTOR(3 downto 0);
           O : out STD_LOGIC_VECTOR(6 downto 0) );
end bcd_to_7_segment;

architecture DATAFLOW of bcd_to_7_segment is

    signal HEX_BUF : STD_LOGIC_VECTOR(6 downto 0);

begin

    with I select

        HEX_BUF <=   "1000000" when "0000", --0
                     "1111001" when "0001", --1
                     "0100100" when "0010", --2
                     "0110000" when "0011", --3
                     "0011001" when "0100", --4
                     "0010010" when "0101", --5
                     "0000010" when "0110", --6
                     "1111000" when "0111", --7
                     "0000000" when "1000", --8
                     "0011000" when "1001", --9
                     "1000001" when others; --U

    O <= HEX_BUF;

end DATAFLOW;
```

Figure 21 - BCD-to-Seven Segment Converter

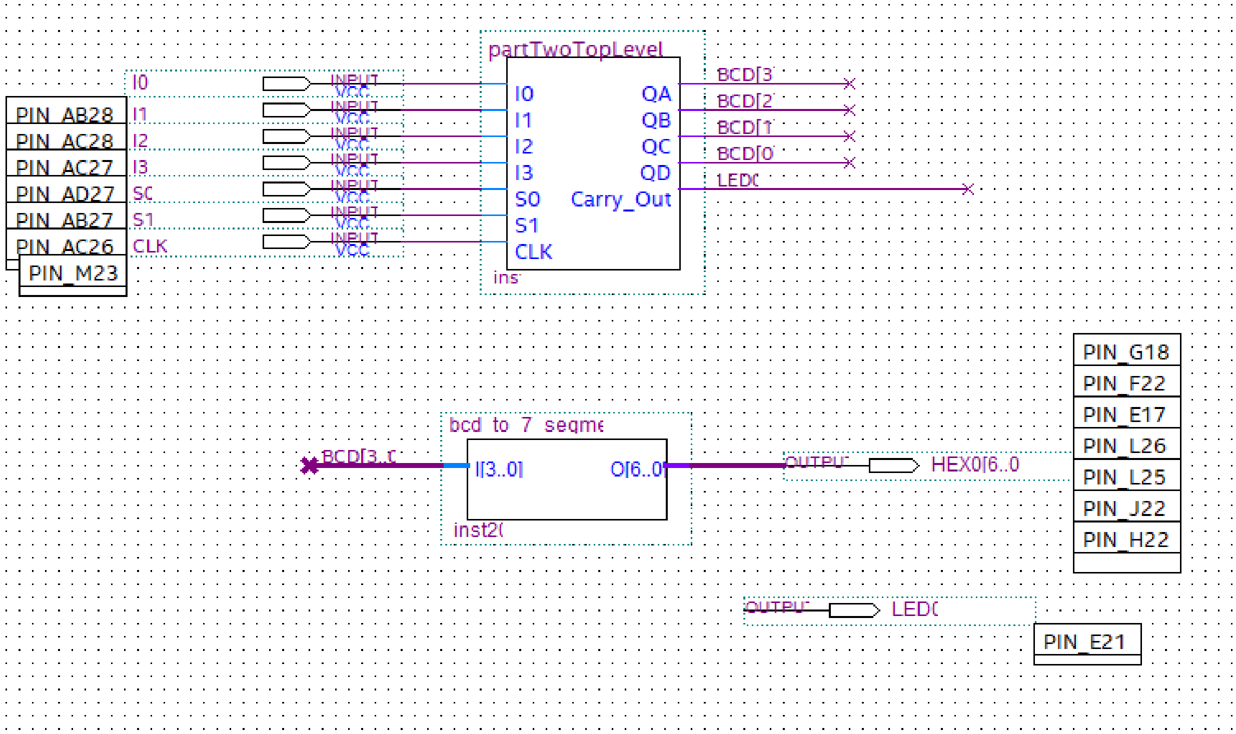


Figure 22 - Final Circuit

Experiment Results

We created a binary counter in this lab that counts from 0 to 7 and repeats in part 1. We built the UP and DOWN logic for a counter circuit in part 2 and applied it into the counter circuit. We used transition and truth tables, as well as K-maps, to accomplish this. The relevant equations were also obtained using the K-maps.

Conclusion

We gained more experience creating counter circuits in this lab, and we learned when to utilize flip flops and when not to. We were also able to emphasize the significance of using a hierarchical design method to simplify complicated circuits.