ECE 273: Introduction to Digital Logic

Lab <8> : <Sequence Detector>

By

<Hussein El-Souri>

Lab Instructor: < Zheming Liang >

Sunday, April 9, 2017
Winter 2017

Honor Code:

I have neither given nor received unauthorized assistance on this graded report.

X___Hussein El-Souri

Department of Electrical and Computer Engineering
College of Engineering and Computer Science
The University of Michigan-Dearborn

Objective

Design a system to detect a sequence of inputs. The system will monitor one input and it will control one output. The system shall be designed to detect the sequence 1101. If the system detects this sequence, it should toggle the state of its output. I must design the sequence detector two different ways; one using Mealy (using JK flip flops) and the other using Moore (using T flip flops) type state machines with minimal number of states.

Theory

The design of this lab requires the use of state diagrams, state tables, excitation tables and K-maps.

Equipment Used

- Software: Xilinx, PlanAhead, Logic Friday
- Hardware: BASYS 2 FPGA board

Procedure

First off, I start with the mealy state diagram.

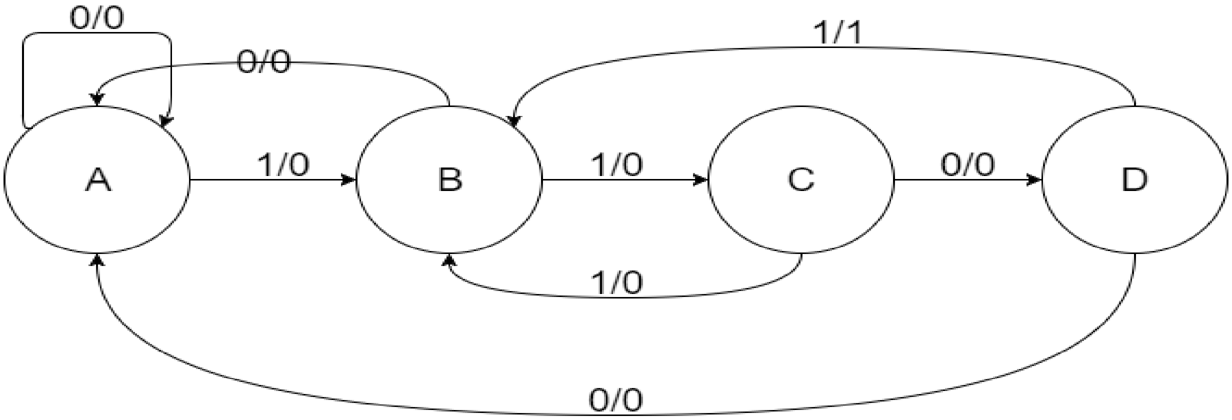


Figure 1: Mealy State Diagram

PS	NS		Z	
	X=0	X=1	X=0	X=1
A	A	B	0	0
B	A	C	0	0
C	D	B	0	0
D	A	B	0	1

Table 1: State Table

Q1Q2	X=0		X=1		Z	
	Q1+	Q2+	Q1+	Q2+	X=0	X=1
00	0	0	0	1	0	0
01	0	0	1	1	0	0
11	1	0	0	1	0	0
10	0	0	0	1	0	1

Table 2: State Assignment Table

Q1Q2	X=0		X=1		X=0		X=1		Z	
	J1	K1	J1	K1	J2	K2	J2	K2	X=0	X=1
00	0	X	0	X	0	X	1	X	0	0
01	0	X	1	X	X	1	X	0	0	0
11	X	0	X	1	X	1	X	0	0	0
10	X	1	X	1	0	X	1	X	0	1

Table 3: Excitation Table

x \ Q1Q2	00	01	11	10
0			x	x
1		1	x	x

Table 4: J1 K-maps

J1= Q2X

K2= **Q2' + X**

x \ Q1Q2	00	01	11	10
0	x	x		1
1	x	x	1	1

Table 5: K1 K-map

x \ Q1Q2	00	01	11	10
0		x	x	
1	1	x	x	1

Table 6: J2 K-map

J2 = X

K2= X'

x \ Q1Q2	00	01	11	10
0	x	1	1	x
1	x			x

Table 7: K2 K-map

x \ Q1Q2	00	01	11	10
0				
1				1

Table 8: Z k-map

Z = Q1Q2'X

Now I design a VHDL Module that follows these equations through the following code:

```
begin

J1 <= Q2 AND X;
K1 <= (NOT Q2) OR X;
J2 <= X;
K2 <= NOT X;
Z <= Q1 AND (NOT Q2) AND X;

end Behavioral;
```

I use this module to create a schematic symbol as such:

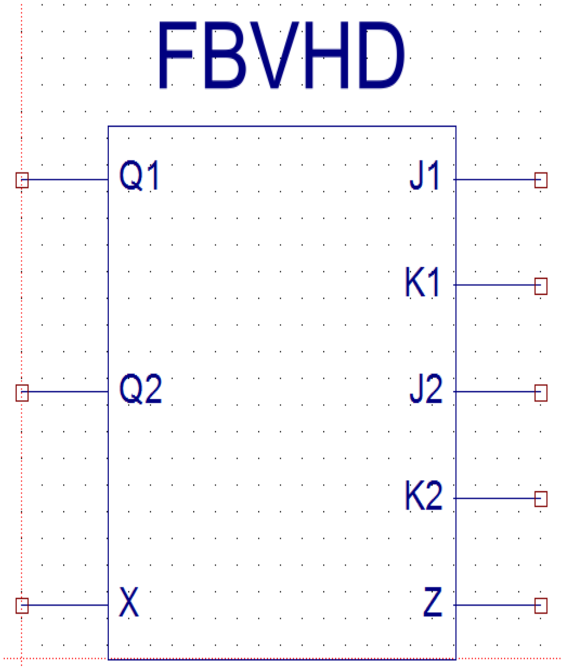


Figure 2: FBVHD symbol

Now I use this symbol and two J/k flip flops to design our mealy machine I also need a T flip flop to toggle the output and hold it at 1. I also need a clock divider and to make appropriate connections and I/O assignments resulting in the following main schematic:

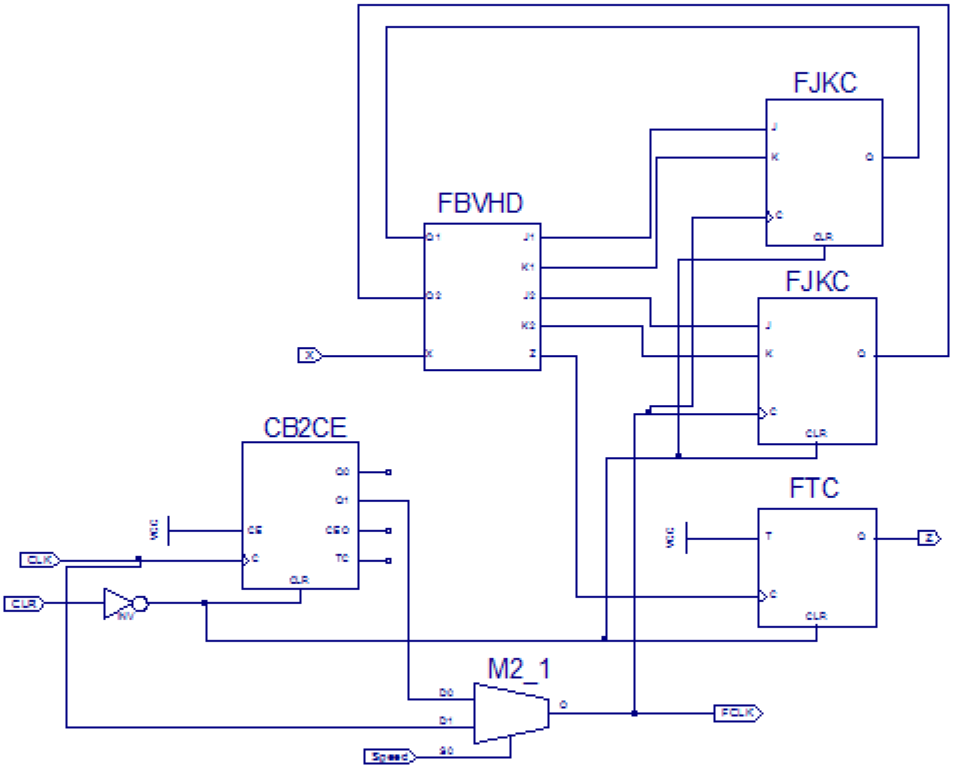
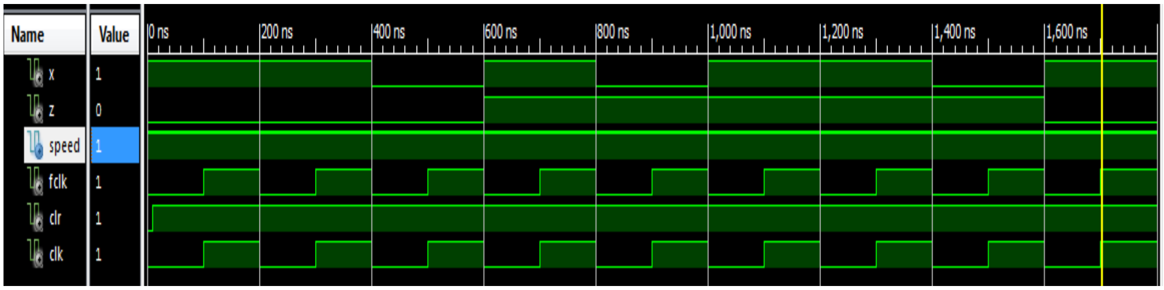


Figure 3: Main Schematic

Saving the schematic properly will allow us to create a test bench that will simulate this circuit through the following code:

```
-- *** Test Bench - User Defined Section ***
RESET : PROCESS
   BEGIN
      CLR <= '0';
             WAIT FOR 10ns;
             CLR <= '1';
             WAIT;
   END PROCESS;
CLOCK :PROCESS
       BEGIN
             CLK <= '0';
             WAIT FOR 100ns;
             CLK <= '1';
             WAIT FOR 100ns;
       END PROCESS;
SPD :PROCESS
       BEGIN
             SPEED <= '1';
             WAIT;
       END PROCESS;
tb : PROCESS
   BEGIN
      X<='1';
             WAIT FOR 200ns;
             X<='1';
             WAIT FOR 200ns;
             X<='0';
             WAIT FOR 200ns;
             X<='1';
             WAIT FOR 200ns;
             X<='0';
             WAIT FOR 200ns;
             X<='1';
             WAIT FOR 200ns;
             X<='1';
             WAIT FOR 200ns;
             X<='0';
             WAIT FOR 200ns;
             X<='1';
             WAIT FOR 200ns;
   END PROCESS;
-- *** End Test Bench - User Defined Section ***
```

Simulating this code will allow us to generate the waveforms.



Name	Value
x	1
z	0
speed	1
fclk	1
clr	1
clk	1

Figure 4: Waveforms

Now for the Moore machine it will follow this state diagram:

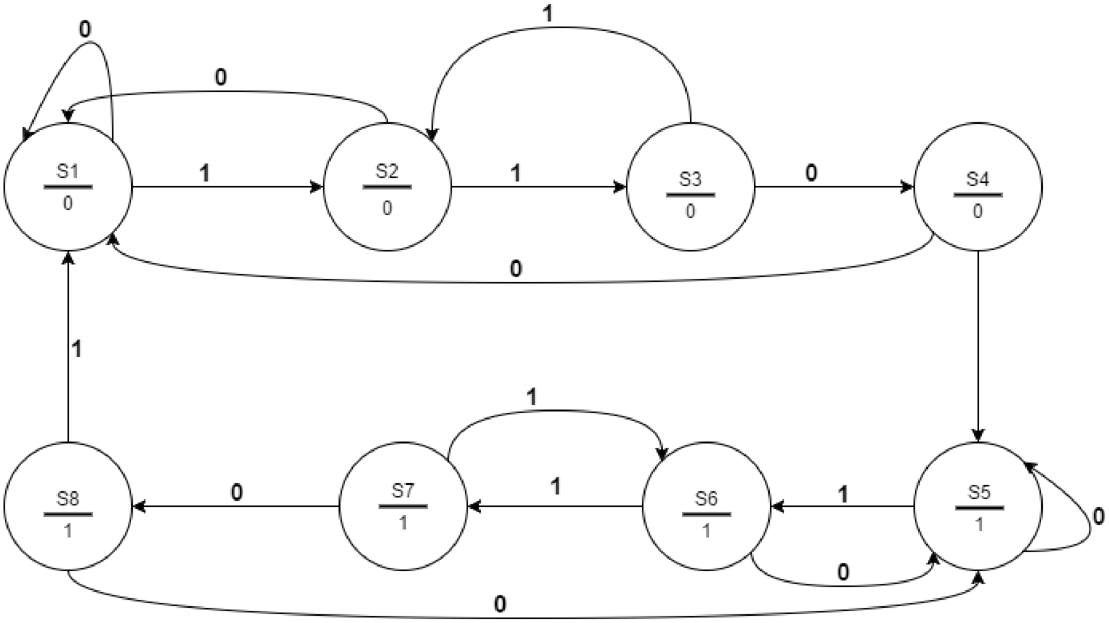
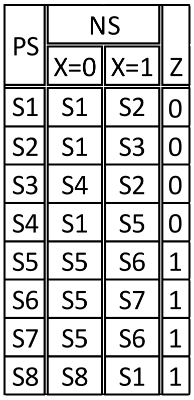


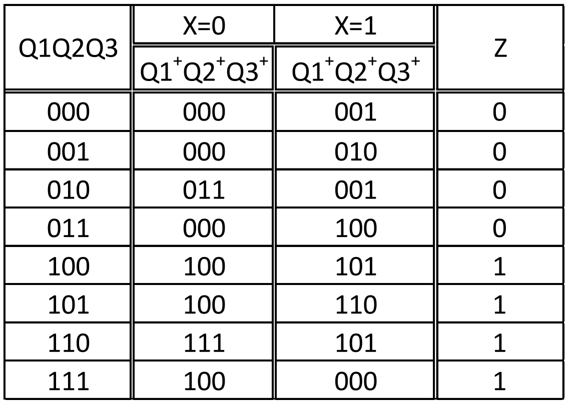
Figure 5: Moore state diagram

This diagram gives us the following state table:



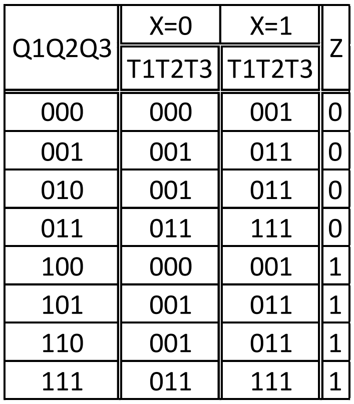
PS	NS		Z
	X=0	X=1	
S1	S1	S2	0
S2	S1	S3	0
S3	S4	S2	0
S4	S1	S5	0
S5	S5	S6	1
S6	S5	S7	1
S7	S5	S6	1
S8	S8	S1	1

Table 9: Moore state table



Q1Q2Q3	X=0	X=1	Z
	$Q1^+Q2^+Q3^+$	$Q1^+Q2^+Q3^+$	
000	000	001	0
001	000	010	0
010	011	001	0
011	000	100	0
100	100	101	1
101	100	110	1
110	111	101	1
111	100	000	1

Table 10: State assignment table



Q1Q2Q3	X=0	X=1	Z
	T1T2T3	T1T2T3	
000	000	001	0
001	001	011	0
010	001	011	0
011	011	111	0
100	000	001	1
101	001	011	1
110	001	011	1
111	011	111	1

Table 11: Excitation Table

Now I use k-maps to find minimized equations for T1,T2,T3 but for speed and accuracy I will use logic Friday.

T1 = Q1Q3X;

T2 = Q2Q3 + Q2X + Q3X;

T3= Q2 + Q3+ X’

Z= Q1;

Using these equations, I create a module:

```
begin

T1 <= Q1 AND Q3 AND X;
T2 <= (Q2 AND Q3) OR (Q2 AND
X) OR (Q3 AND X);
T3 <= Q2 OR Q3 OR X;
Z <= Q1;

end Behavioral;
```

And from the module I create a symbol and from a symbol a main schematic for which I run a test bench and compare waveforms.

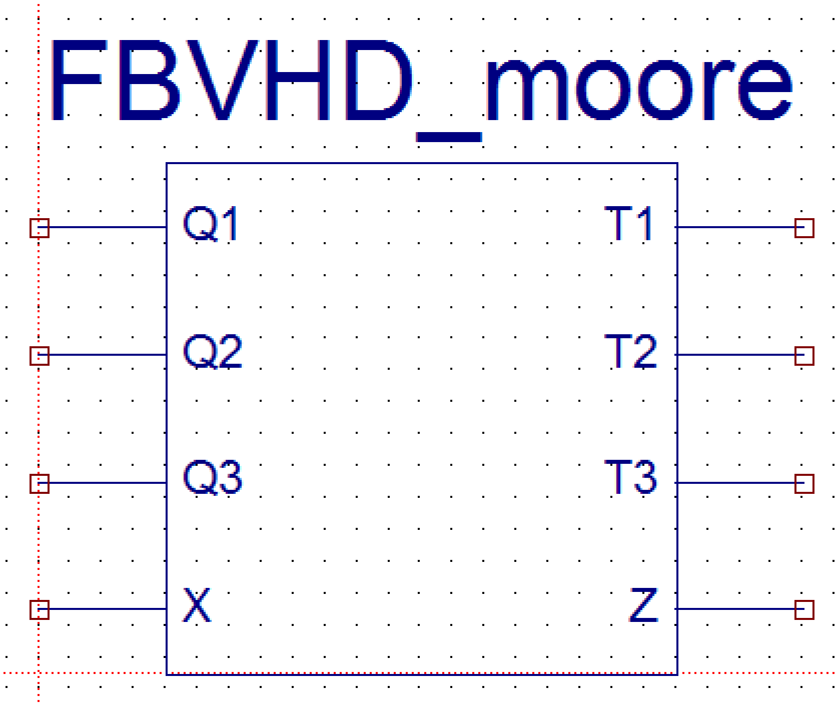


Figure 6: FBVHD_moore

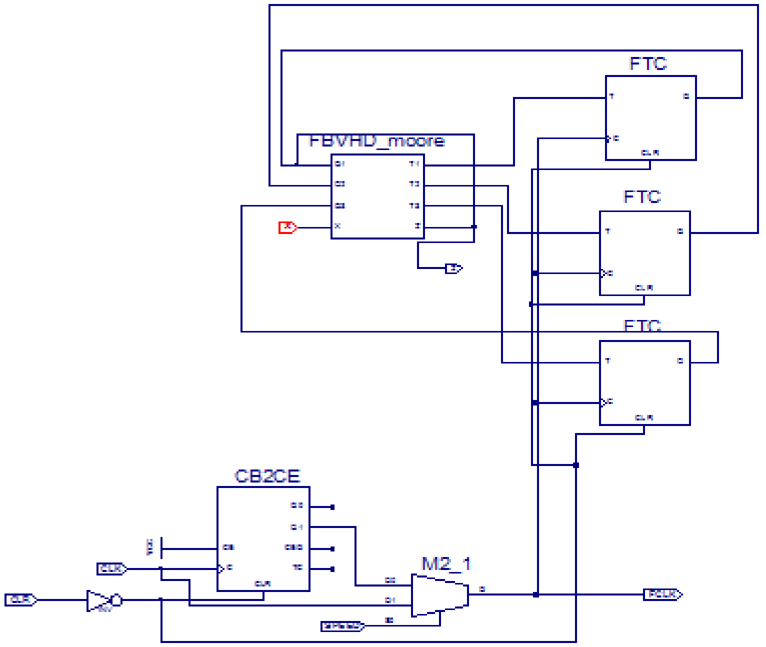
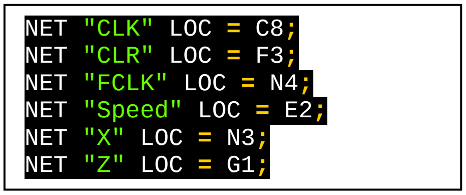


Figure 7: moore main schematic

Results

Comparing waveforms, I deduce both designs are the same also after implementing mealy on the BASYS board using these constraints and planAhead



```
NET "CLK" LOC = C8;
NET "CLR" LOC = F3;
NET "FCLK" LOC = N4;
NET "Speed" LOC = E2;
NET "X" LOC = N3;
NET "Z" LOC = G1;
```

I can simulate the design on the board such that I switch inputs along with the clock only. Meaning every time, the FCLK turns on I can change the input and try to create a sequence.

Conclusion

I realized that sequence detection does not occur instantly meaning that the input can only be read and considered part of the sequence when the clock is active .