ECED 4260 IC design and Fabrication

Assignment #1 Reference Solution

1)

F1(X,Y,Z)=YZ'+X'YZ+XYZ

=YZ'+(X'+X)YZ=YZ'+YZ=Y(Z+Z')=Y(1)=Y

F2(X,Y,Z)=(X+Y)(X'+Y+Z)(X'+Y+Z')

=(X+Y)(X'+Y)=Y

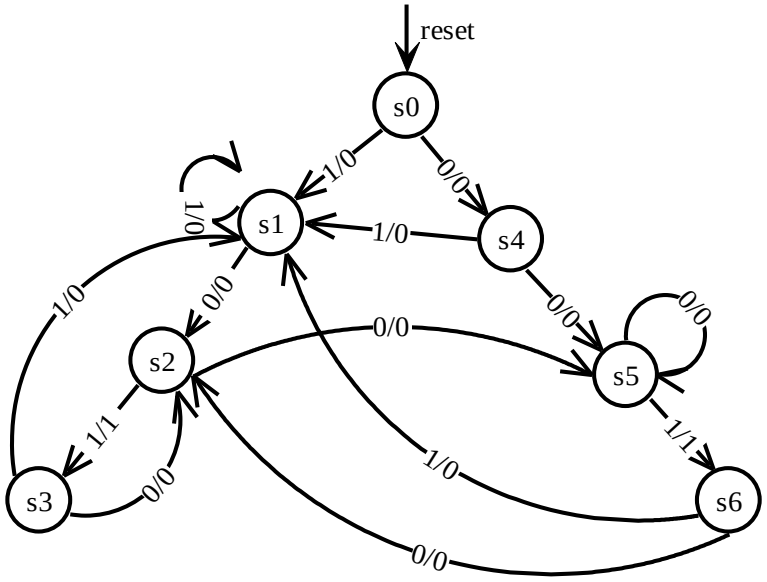
F3(W,X,Y,Z)=X+XYZ+X'YZ+X'Y+WX+W'X

=X+X'Y+(W+W')X=X+X'Y+X=X+X'Y = X+Y

2)

Design a mealy sequential circuit with one input and one output which detects instances of the patterns "101" and "001", including overlapping patterns.

Step1: State diagram:



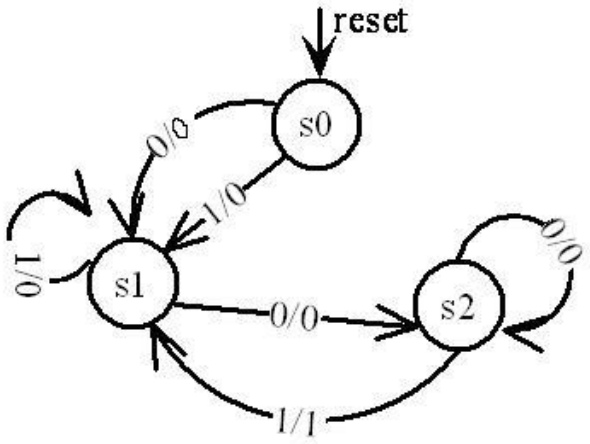
Derive the symbolic table:

Present	Next state		Output	
State	X=0	X=1	X=0	X=1
S0	S4	S1	0	0
S1	S2	S1	0	0
S2	S5	S3	0	1
S3	S2	S1	0	0
S4	S5	S1	0	0
S5	S5	S6	0	1
S6	S2	S1	0	0

Step 2: Reduced table:

S1	2-4					
S2	X	X				
S3s	2-4	Ok	X			
S4	5-4	2-5	X	2-5		
S5	X	X	6-3	X	X	
S6	2-4	Ok	X	ok	2-5	X
	S0	S1	S2	S3	S4	S5

Present	Next state		Output	
State	X=0	X=1	X=0	X=1
S0	S1	S1	0	0
S1	S2	S1	0	0
S2	S2	S1	0	1



step3: state assignment :

1) States that have the same next state for a given input

   {S0, S1, S2}, {S1,S2}
2) States that are the next states of the same state
   {S1, S2}x2
3) States that have the same output for the same input.
   {S0,S1,S2},{S0, S1}

State assignment: S0: 00 , S1:01, s2:11 (not unique)

Step 4: construct the binary state table:

Present	Next state		Output	
State	X=0	X=1	X=0	X=1
00	01	01	0	0
01	11	01	0	0
11	11	01	0	1

Step 5,6: find minimum flip-flop input & output functions

B \ XA	00	01	11	10
0	0	Φ	Φ	0
1	1	1	0	0

A+

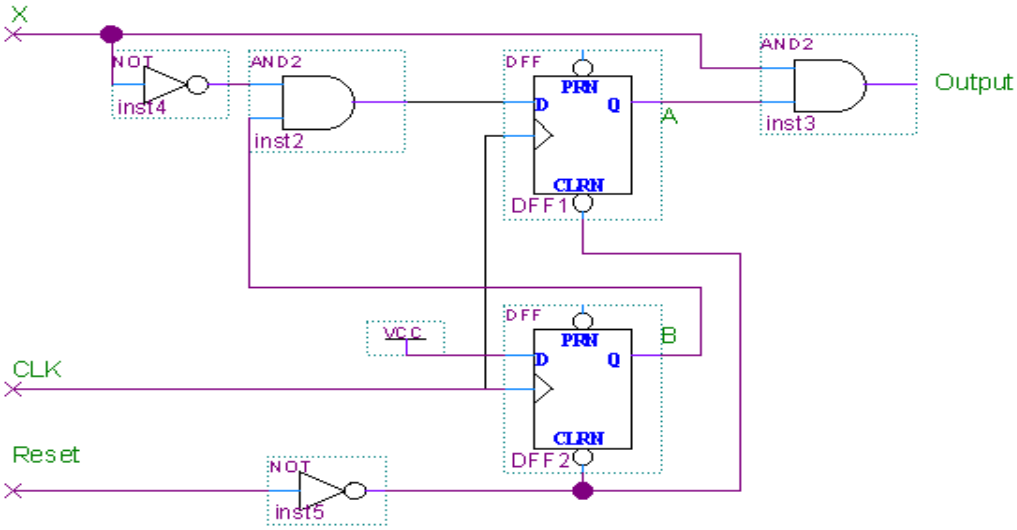
B \ XA	00	01	11	10
0	1	Φ	Φ	1
1	1	1	1	1

B+

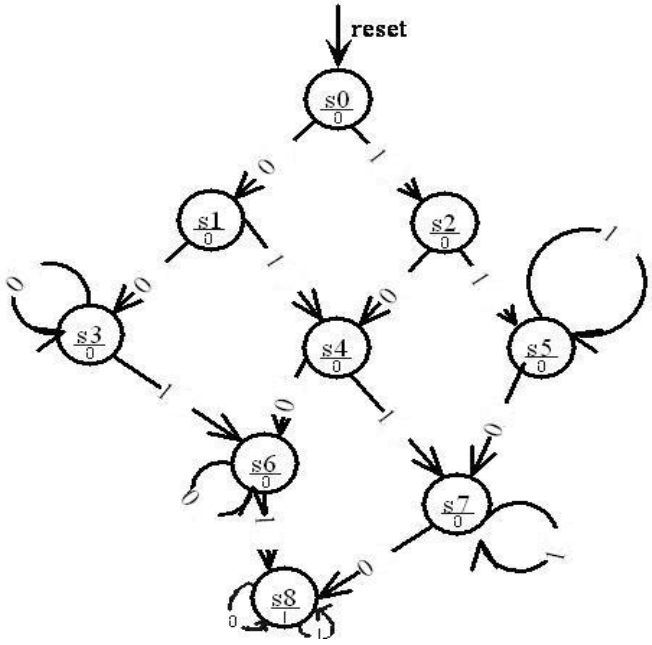
B \ XA	00	01	11	10
0	0	Φ	Φ	0
1	0	0	1	0

output

$\overline{A^+}=\overline{B}+X$ $B^+=1$ output=AX



3)



4) Reduced table:

Present	Next state		Output
	X=0	X=1	
S0	S5	S0	0
S1	S3	S5	0
S3	S1	S4	0
S4	S4	S5	1
S5	S0	S4	0

5) Valid identifiers: Not_bad This_is_one How_about_this_one Last_one
Reserved word: srl

6) For the first one:

```
Entity gates3 is
        Port(a, b, c : in bit;
                d: out bit);
End gates3;

Architecture data_flow of gates3 is
Begin
D<= a and b and c;
End data_flow;
```

For the second one;

```
Entity and2 is
Port( i1, i2: in bit;
        o1: out bit);
End and2;

Architecture data_flow of and2 is
Begin
o1<= i1 and i2;
End data_flow;
-------------------------------------------

Entity gates3 is
 Port(a, b, c : in bit;
        d: out bit);
End gates3;
```

```
Architecture structural1 of gates3 is
Signal e : bit;

Component and2
Port (i1, i2 : in bit; o1 : out bit);
End component;

Begin
First_gate : and2
Port map ( i1 => a, i2 => b, o1 => e);
second_gate : and2
Port map ( i1 => e, i2 => c, o1 => d);
End structural1;
```

Please note: If you use std_logic type, you have to include **library** and **use** statements at the beginning of your vhdl code.