BILKENT UNIVERSITY
Department of Electrical and Electronics Engineering
EEE102 Introduction to Digital Circuit Design
Midterm Exam Solution

6-04-2007

Duration 120 minutes

Surname: _____

Name: _____

ID-Number: _____

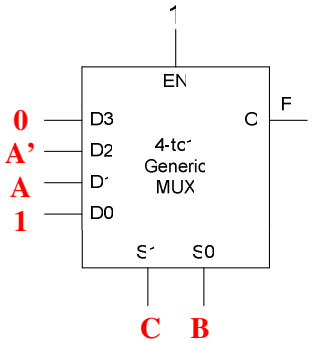
Signature: _____

There are 5 questions of equal weight. Solve all.
Do not detach pages. Show all your work.

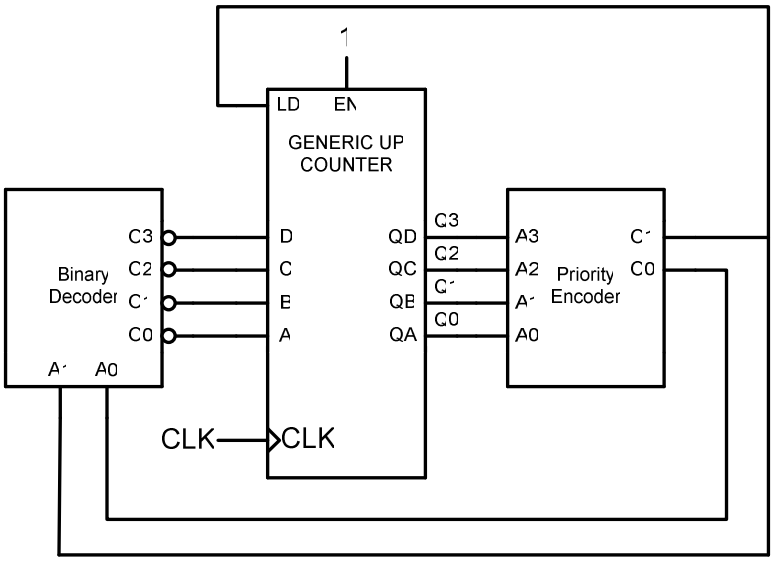
Q1	
Q2	
Q3	
Q4	
Q5	
Total	

Q1. a) (4 points) Implement the function F=A'B+AC by using the following 4-to-1 generic multiplexer. You may assume that the inverted inputs are available. No other logic gates are allowed.

One possible solution is as the following. There are other ways too.



b) (16 points) Determine the counting sequence of the following configuration and fill in the following table. Assume that the counter is initially 0. Priority order is A3, A2, A1, and A0 from high to low in the priority encoder.



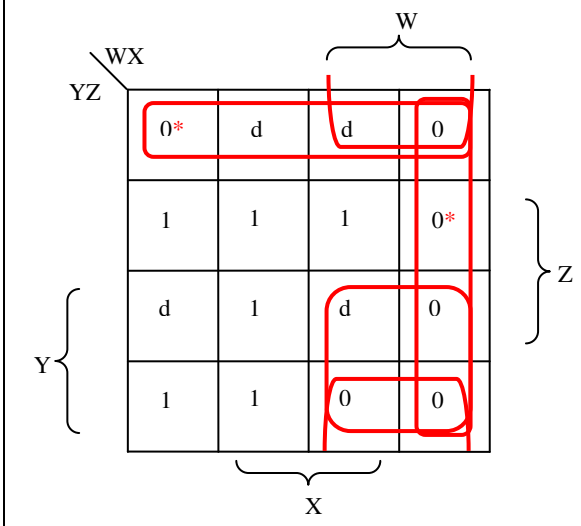
Next State Table to determine the counting sequence:

Q3	Q2	Q1	Q0	LD	Q3*	Q2*	Q1*	Q0*
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	0	0
0	1	0	0	1	1	0	1	1
1	0	1	1	1	0	1	1	1
0	1	1	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1

Counting sequence: 7 – 11 – 7 – 11 – 7 – 11…..

Q2. a) (5 points) Find all minimized POS expressions for the following TT.

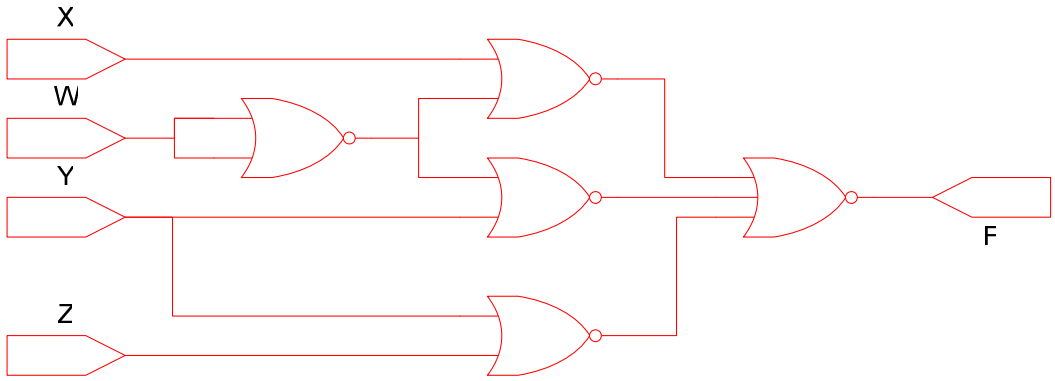
Z	Y	X	W	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	d
0	1	0	0	d
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	d
1	1	0	1	1
1	1	1	0	0
1	1	1	1	d



F=(Y+Z)(W'+X)(W'+Z), F=(Y+Z)(W'+X)(W'+Y')

b) (5 points) Draw the schematics of one of the above minimized functions by using NOR gates only.

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



c) (5 points) Expand F=(X+Y').Z + (X'.Y.Z') using Shannon's expansion theorem with respect to Y.

Shannon's expansion theorem is as follows:

F(X1, X2, …. Xn) = X1.F(1,X2,,…..Xn)+ X1'.F(0,X2,,…..Xn)

F=Y(Z+X'Z')+ Y'Z

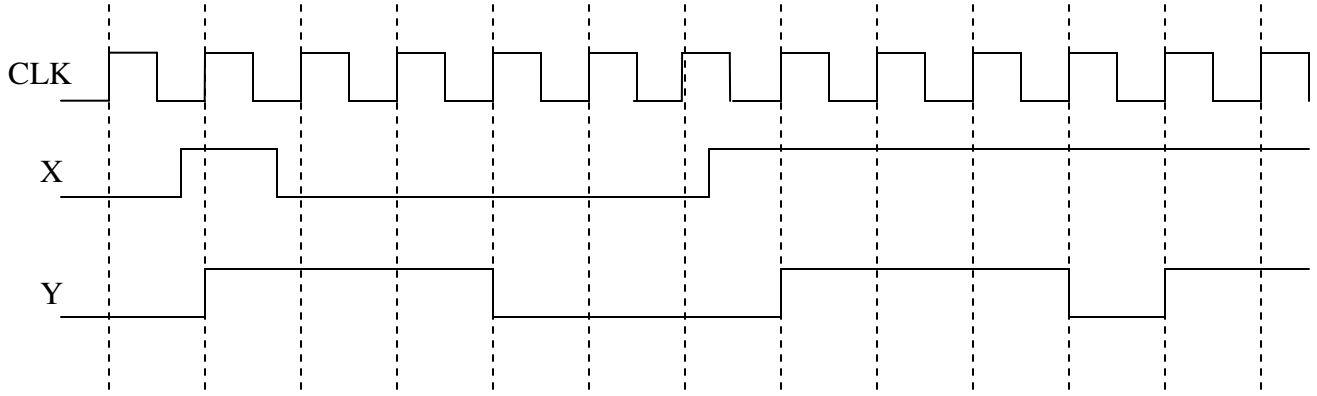
d) (5 points) Prove using algebraic manipulation that

(X+Y').Z + (X'.Y.Z') = (X+Y'+Z').(X'+Z).(Y+Z)

Duality: (XY'+Z)(X'+Y+Z')=(XY'Z')+(X'Z)+(YZ)

Multiply Out Left Side: XY'Z'+X'Z+YZ (Note that XX'=YY'=ZZ'=0)

Q3. A synchronous FSM has one input X and one output Y. Y is 0 initially (at Power ON). If at a clock +ve tick X = 1 then Y becomes 1 for three clock periods and then returns to 0. If X is 1 during this time then it is not effective. Write VHDL code to describe this circuit. The timing diagram below is given for you to understand the problem better.



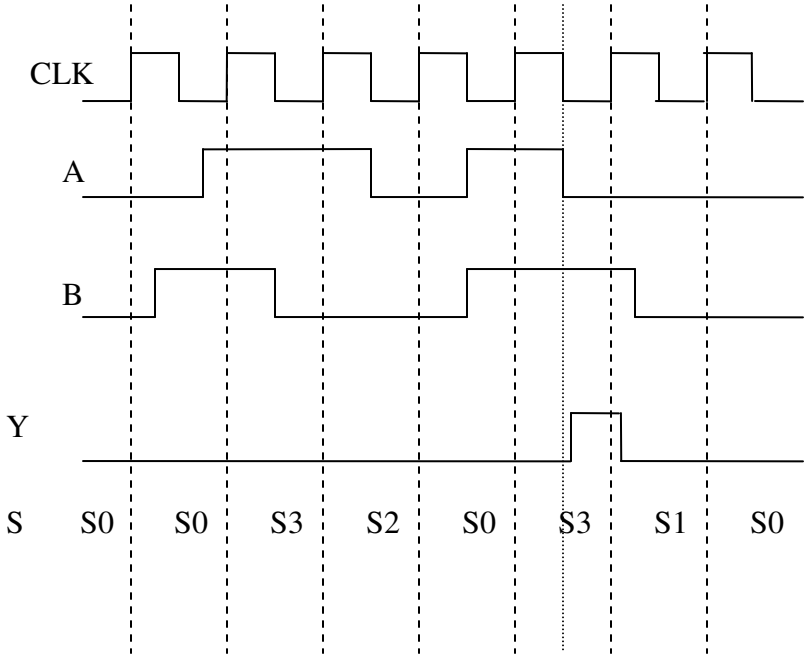
Solution:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity OneShot is
   Port ( clock : in std_logic;
          X : in std_logic;
          Y : inout std_logic);
end OneShot;

architecture Behavioral of OneShot is
signal flag:std_logic :='0';
begin
Y<=flag;
process(clock)
variable count:std_logic_vector (1 downto 0) :="00";
begin
        if clock'event and clock='1' then
                if flag='0' then
                        if X='1'then flag<='1';end if;
                elsif flag ='1' then
                        count:=count+1;
                        if count="11" then flag<='0';count:="00";end if;
                end if;
        end if;
end process;

end Behavioral;
```

Q4. The VHDL code of an FSM is given below.

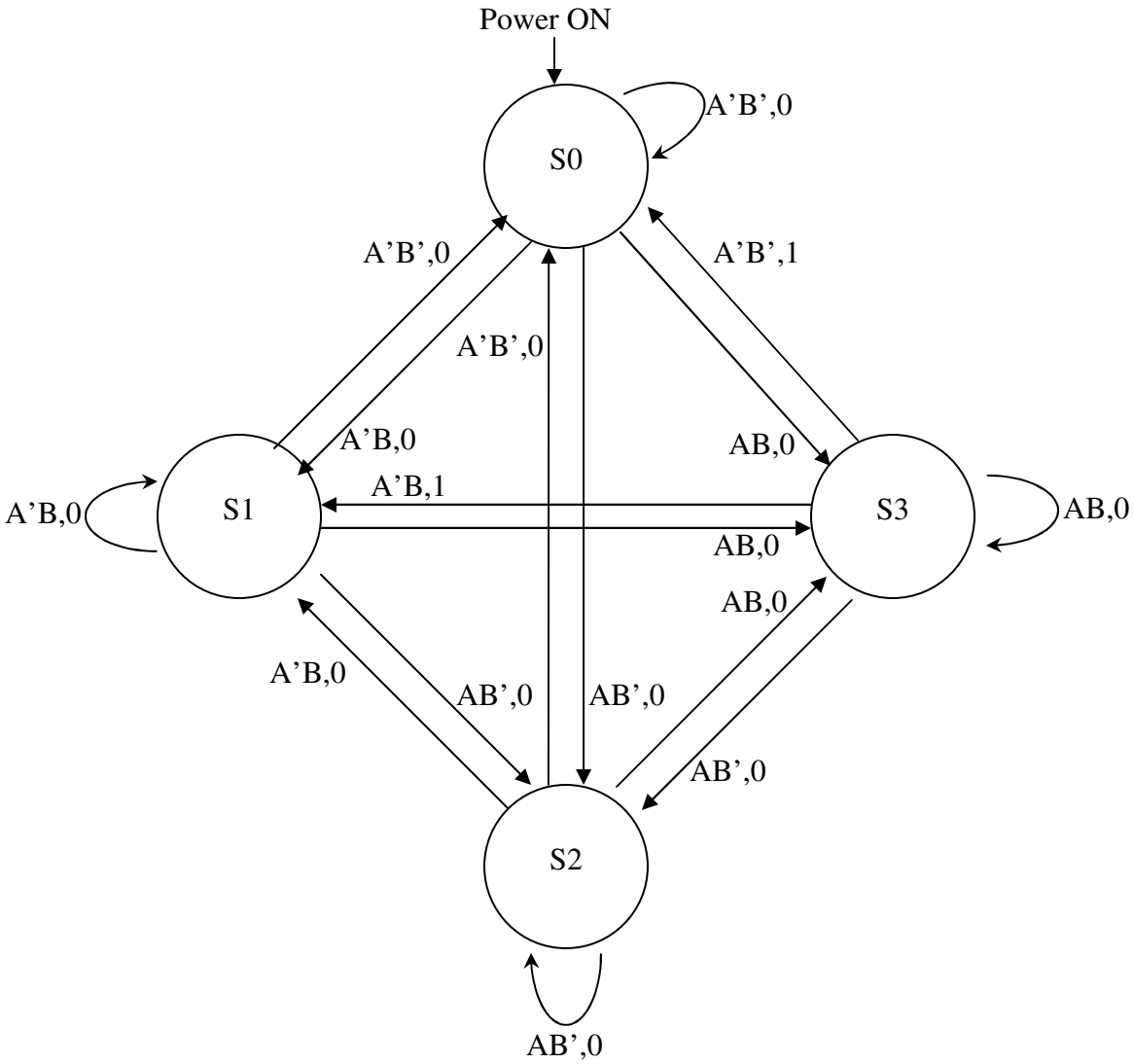
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity TwoInputFSM is
   Port ( CLK : in std_logic;
          A : in std_logic;
          B : in std_logic;
          Y : out std_logic);
end TwoInputFSM;

architecture Behavioral of TwoInputFSM is
type newtype is (S0,S1,S2,S3);
signal S:newtype :=S0;
signal N:newtype;

begin
S<=N when CLK'event and CLK='1' else S;
N<=S1 when (not A and B)='1' else S2 when (A and not B)='1' else S3 when (A and
B)='1' else S0 when (not A and not B)='1';
Y<='1' when S=S3 and A='0' else '0';
end Behavioral;
```

a) Draw the waveform of Y for the given waveforms for A and B.
Solution:

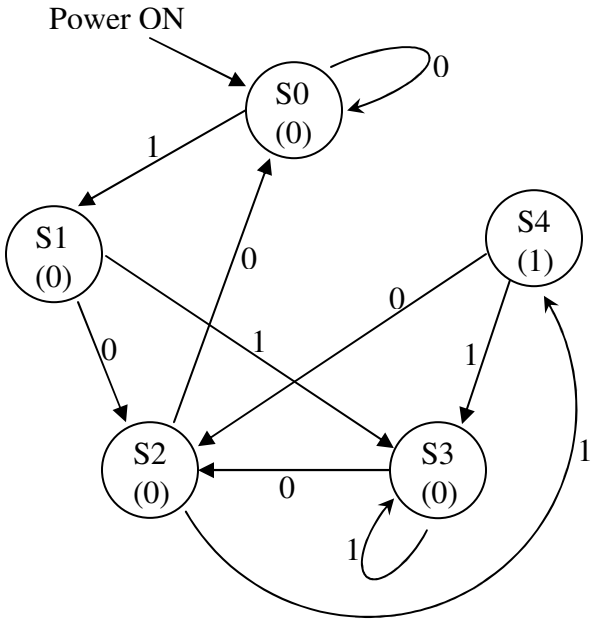


b) **Draw the state/output diagram of this FSM**

Solution:



Q5. Design and draw the FSM which has the state/output diagram shown below. Show all your design steps. What does this FSM do?



Considering that S1 and S3 are equivalent states,

State Encoding:

State	Q1	Q0
S0	0	0
S1	0	1
S2	1	0
S4	1	1

Next State Table:

Q1	Q0	X	Q1*	Q0*
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	1	0
1	1	1	0	1

Excitation Table: Same as NS table.

Next State Logic:

Q1*=D1=Q0X'+Q1Q0X

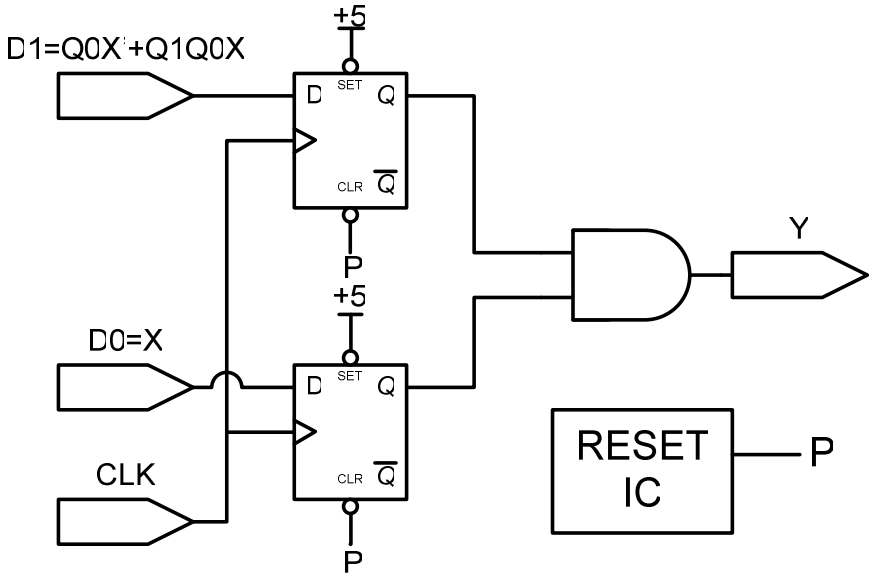
Q0*=D0=X

Output Table:

Q1	Q0	Y
0	0	0
0	1	0
1	0	0
1	1	1

Y=Q1.Q0

Circuit:



Function:

Detects "101" sequence from the input.