Assignment # 3 - CSI 2111(Solutions)

Q1. Realize, using a suitable PLA, the following functions : [10 marks]

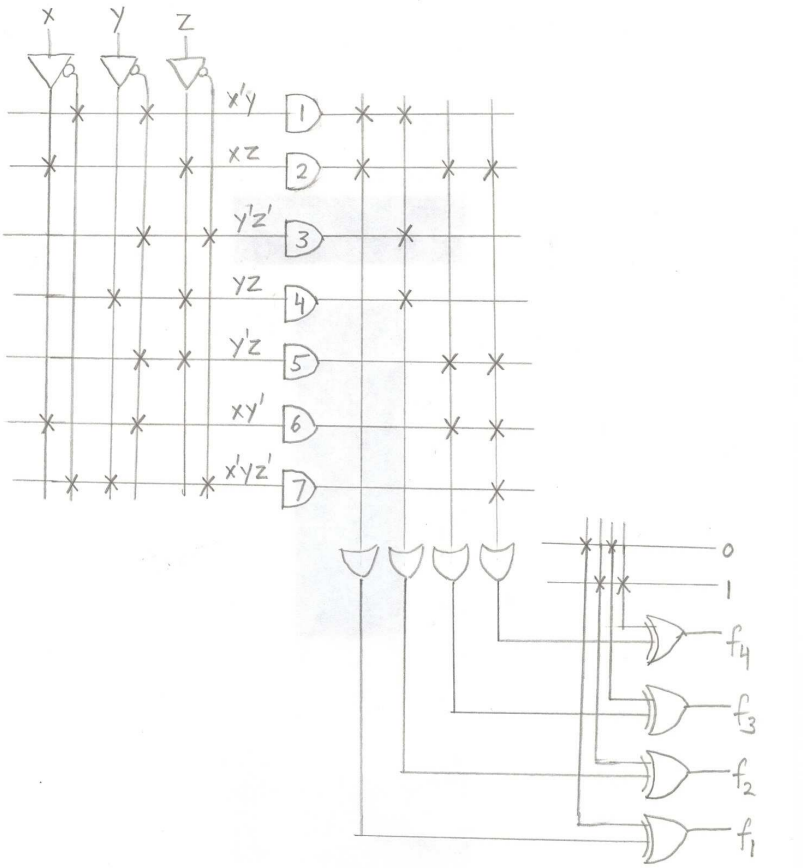
$f_1(x,y,z) = \Sigma m(0,1,5,7)$
$f_2(x,y,z) = \Sigma m(2,5,6)$
$f_3(x,y,z) = \Sigma m(1,4,5,7)$
$f_4(x,y,z) = \Sigma m(0,3,6)$

Minimize the number of product terms and show all connections.

$f_1 = x'y' + xz$ $f_1' = x'y + xz'$
$f_2 = yz' + xy'z$ $f_2' = x'y' + y'z' + yz$ (or $x'z + y'z' + yz$)
$f_3 = y'z + xy' + xz$ $f_3' = x'z' + x'y + yz'$
$f_4 = xyz' + x'yz + x'y'z'$ $f_4' = xy' + xz + y'z + x'yz'$

Two options give minimum (7) product terms: { f_1, f_2, f_3, f_4'} or { f_1, f_2', f_3, f_4'}
PLA programming table for { f_1, f_2', f_3, f_4'}:

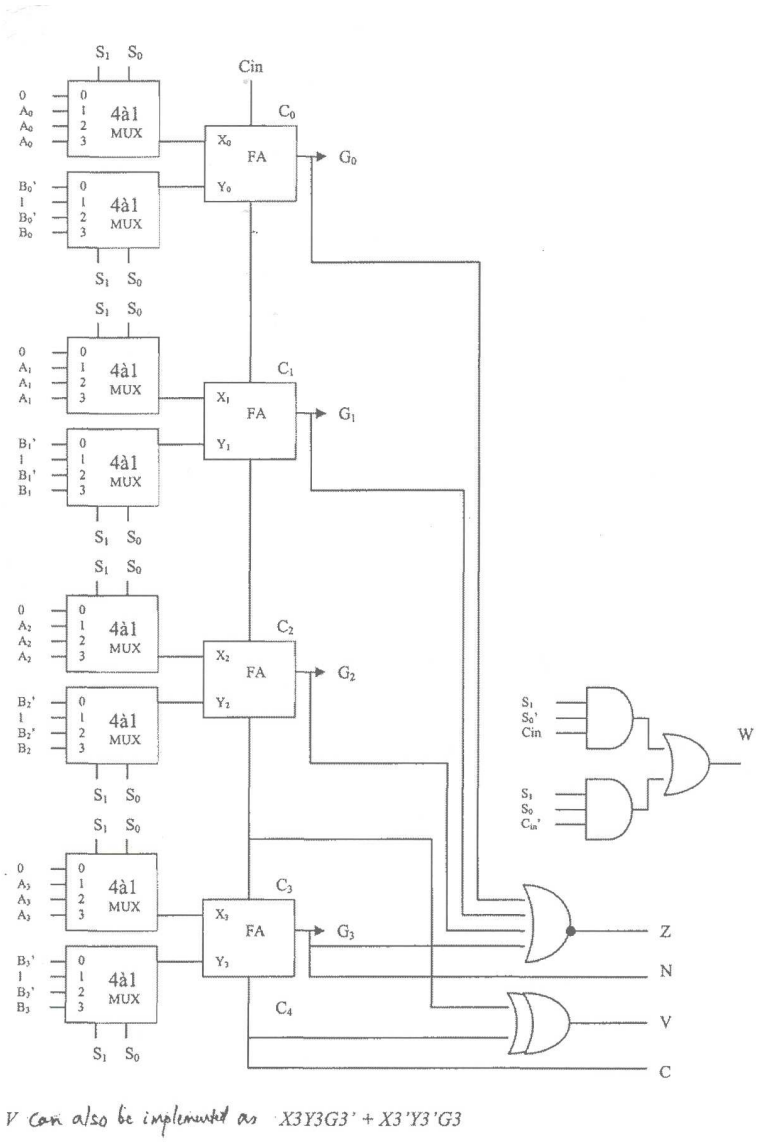
Term	Number of terms	Inputs			Outputs				
		x	y	z	f1	f2	f3	f4	
1	x'y'	0	0	--	1	1	--	--	
2	xz	1	--	1	1	--	1	1	
3	y'z'	--	0	0	--	1	--	--	
4	yz	--	1	1	--	1	--	--	
5	y'z	--	0	1	--	--	1	1	
6	xy'	1	0	--	--	--	1	1	
7	x'yz'	0	1	0	--	--	--	1	
					T	C	T	C	T/C



Q2. Design a 4-bits arithmetic unit with two input selection lines (S_1 and S_0) which generates the following arithmetic operations: [10 marks]

$S_1 S_o$	$\mathbf{C_{in} = 0}$	$\mathbf{C_{in} = 1}$
00	$G = B'$ (complement)	$G = B' + 1$ (negation 2CF)
01	$G = A - 1$ (decrement)	$G = A$ (transfer)
10	$G = bbbb$ (invalid operation)	$G = A + B' + 1$ (subtraction)
11	$G = A + B$ (addition)	$G = bbbb$ (invalid operation)

Suppose $A = A_3A_2A_1A_0$, $B = B_3B_2B_1B_0$, and $G = G_3G_2G_1G_0$. $G = bbbb$ is an unspecified combination of bits which cannot be used as valid result. Use elementary adders and multiplexers in your realization. Add also the logical circuits necessary for the calculation of the bits of state N, Z, V, C, and W (bit indicating *invalid operation* when W=true)



V can also be implemented as X3Y3G3' + X3'Y3'G3

Q3. Consider the following sequential circuit: [10 marks]

$J_A = [(AX')'(A'X)']'$
$J_B = (B'+ X')'$
$K_A = AX'$
$K_B = (A'+ X')'$
$Z = (A + B)'$

a) Construct the transition table for this circuit
b) Construct the transition diagram for this circuit
c) Determine the state equations of the circuit

d) What is the output sequence when one applies the input sequence 01100 to X (the circuit is initially with state AB=00)

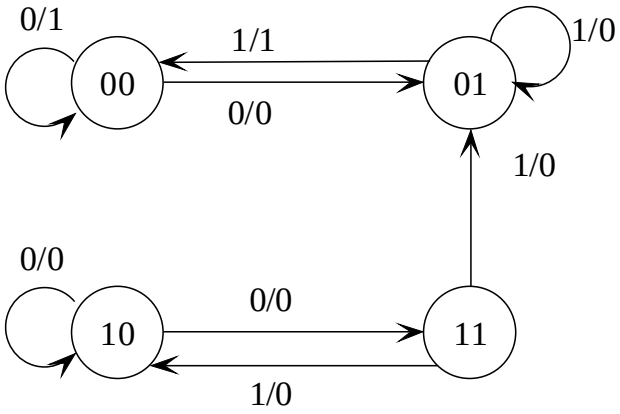
a)

$J_A = AX' + A'X$, $K_A = AX'$

$J_B = BX$, $K_B = AX$

Present State	Next State x=0	Next State x=1	Output x=0	Output x=1
A B	A B	A B	z	z
0 0	0 0	1 0	1	1
0 1	0 1	1 1	0	0
1 0	0 0	1 0	0	0
1 1	0 1	1 0	0	0

b)



c)

A(t+1) = X

B(t+1) = A'B + BX'

d) 11001

Q4. The following transition table describes a sequential circuit: [10 marks]

Present State	Next State x=0	Next State x=1	Output x=0	Output x=1
A B	A B	A B	z	z
0 0	0 0	0 1	1	0
0 1	0 1	1 0	0	1
1 0	1 0	1 1	1	0
1 1	1 1	1 1	0	1

a) Determine the state equations and output of this circuit.

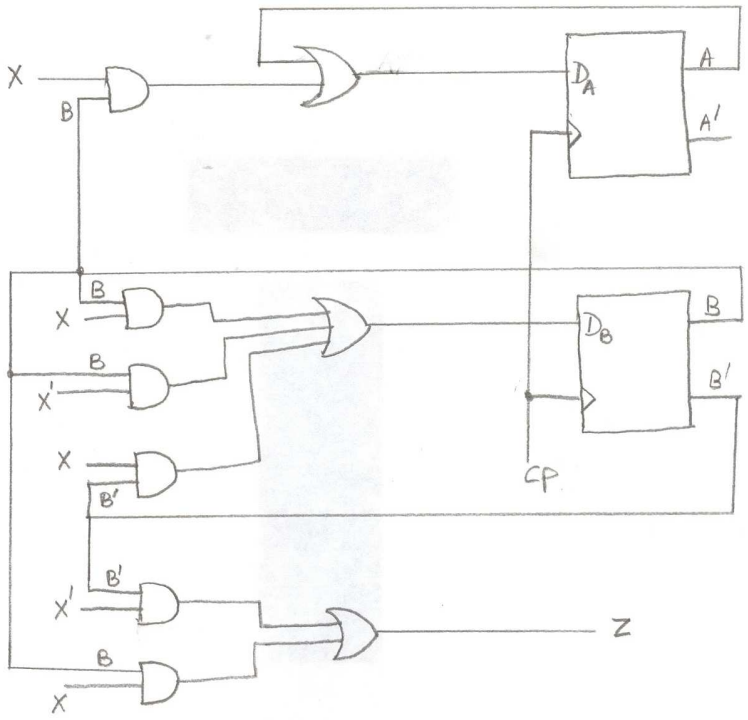
b) Realize this using D flip-flops.
c) Realize this using RS flip-flops.

Present State		Input	Next State		Output	FF Inputs for b)		FF Inputs for c)			
A	B	X	A	B	Z	D_A	D_B	S_A	R_A	S_B	R_B
0	0	0	0	0	1	0	0	0	X	0	X
0	0	1	0	1	0	0	1	0	X	1	X
0	1	0	0	1	0	0	1	0	X	X	0
0	1	1	1	0	1	1	0	1	0	0	1
1	0	0	1	0	1	1	0	X	0	0	X
1	0	1	1	1	0	1	1	X	0	1	0
1	1	0	1	1	0	1	1	X	0	X	0
1	1	1	1	1	1	1	1	X	0	X	0

a)

$A(t+1) = A + BX$
$B(t+1) = B'X + BX' + AB$ or $B'X + BX' + AX$
$Z = B'X' + BX$

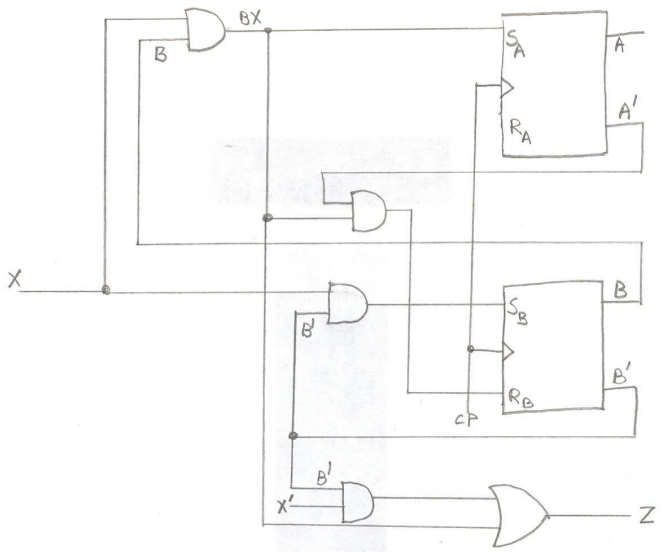
b) $D_A = A + BX$ $D_B = B'X + BX' + AB$



c)

$S_A = BX, \quad R_A = 0$

$S_B = B'X, \quad R_B = A'BX$



Q5. (a) Implement in VHDL, a sequential circuit that takes three signals (Load, ShiftLeft, ShiftRight) and a 8-bit InputValue. Additionally, the circuit has an input Clock signal. The circuit output is a 8-bit OutputValue. The circuit operates on the rising edge of the clock, according to the following rules: [15 marks]

- If Load == 1, then it remembers the 8-bit InputValue. The remembered value is output as a 8-bit value OutputValue.
- ELSE If ShiftLeft == 1, then it updates the remembered value to be its previous value shifted to the left by one bit. The remembered value is output as a 8-bit value OutputValue.
- ELSE If ShiftRight == 1, then it updates the remembered value to be its previous value shifted to the right by one bit. The remembered value is output as a 8-bit value OutputValue.
- ELSE OutputValue is unchanged.

(b) Simulate the circuit that you implemented and test it for the following inputs (in the order as they appear): [5 marks]

1. When LOAD = 1, ShiftLeft = 0, ShiftRight = 0, InputValue = 01100101
2. When LOAD = 0, ShiftLeft = 1, ShiftRight = 0, InputValue = 00000000
3. When LOAD = 0, ShiftLeft = 0, ShiftRight = 1, InputValue = 11111111
4. When LOAD = 1, ShiftLeft = 0, ShiftRight = 1, InputValue = 11110000
5. When LOAD = 0, ShiftLeft = 0, ShiftRight = 1, InputValue = 00000001

Submit the timing diagram showing all the above test cases. (Note that for each test case, you need to show what happens on a rising edge of the Clock.)

(a) VHDL Code:

```
-- CSI2111 Assignment-3 Question-5
-- Load-&-Shift Register
--          If LOAD = 1, loads the InputValue
--      Else if SHIFTLEFT = 1, shifts left the stored value
--      Else if SHIFTRIGHT = 1, shifts right the stored value
-- (operates on rising edge of clock)

LIBRARY ieee;
USE ieee.std_logic_1164.all;


ENTITY circuit_A3_Q5 IS
  PORT(
        Clock, Load             :   IN STD_LOGIC;
        ShiftLeft, ShiftRight   :   IN STD_LOGIC;
        InputValue              :   IN STD_LOGIC_VECTOR( 7 DOWNTO 0 );
        OutputValue             :   OUT STD_LOGIC_VECTOR( 7 DOWNTO 0 )
    );
END circuit_A3_Q5;


ARCHITECTURE behavioral OF circuit_A3_Q5 IS
    SIGNAL SavedValue : STD_LOGIC_VECTOR( 7 DOWNTO 0 );    --saves the
remembered value

BEGIN
    OutputValue <= SavedValue;

    update: PROCESS( Clock )          -- this process is executed whenever CLOCK
changes
    BEGIN
        IF( Clock'EVENT AND Clock= '1' ) THEN
            IF Load = '1' THEN
                SavedValue <= InputValue;        -- loads the input value
            ELSIF ShiftLeft = '1' THEN
                SavedValue(7 downto 1) <= SavedValue(6 downto 0);
-- shift left
                SavedValue(0) <= '0';
            ELSIF ShiftRight = '1' THEN
                SavedValue(6 downto 0) <= SavedValue(7 downto 1);
-- shift right
                SavedValue(7) <= '0';
            ELSE
                SavedValue <= SavedValue;
            -- no change
            END IF;                      -- IF load = '1'
        END IF;                     -- IF( clock'event AND clock = '1' )
    END PROCESS update;
END behavioral;
```

(b) Timing diagram: