

=====

Question 1

Using **Boolean algebra**, minimize the following function:

a) $F(A,B,C,D) = ABCD + (ABD)' + ABC'D$

b) Given $f(x,y,z) = xy + xz' + yz$

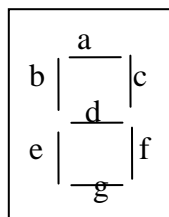
i) Implement f in **NOR-NOR** format

ii) Implement f in **AND-OR-INVERT** format

Obtain optimum implementation.

Question 2

Design a combinational circuit decoder that examines a BCD digit and displays a letter "L" if the digit was less than or equal 5. Use the Display unit shown below. Implement the circuit using minimum 2*1 MUXes.



Question 3

- a) Design a Half Subtractor.
- b) Design a Full Subtractor using two Half Subtractors.
- c) Using two 4 * 1 multiplexers implement the Full Subtractor

Question 4

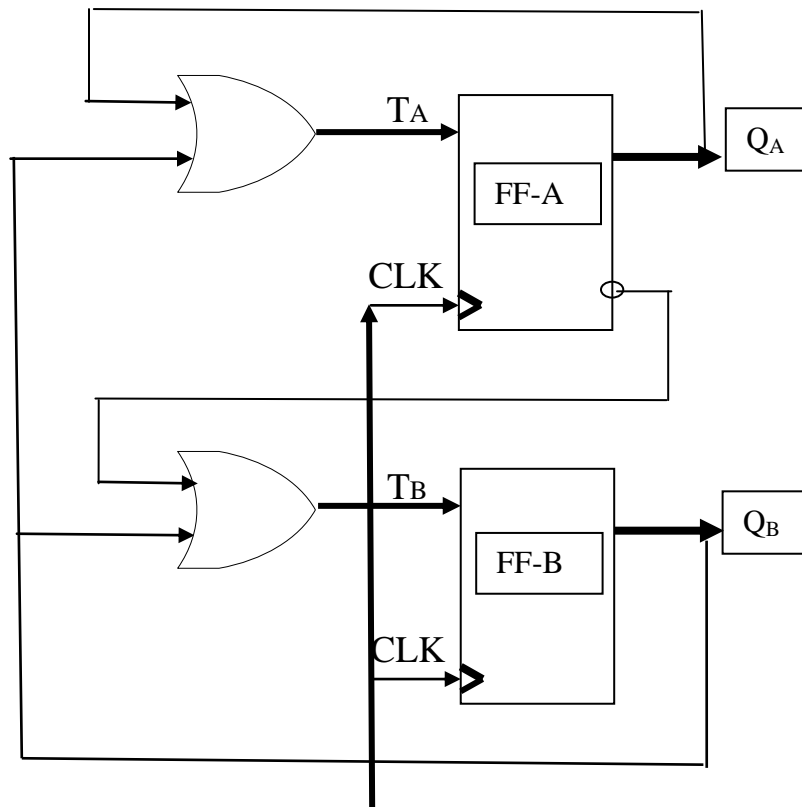
Design a BCD adder that adds two BCD digits and produces a sum digit in BCD. You may use 4-bit binary adders for your design. Give the circuit diagram.

Question 5

Design a sequential circuit with two JK flip flops A & B and two inputs E & F. If E=0, the circuit remains in the same state regardless of the value of F. When E=1 and F=1, the circuit goes through the state transition from 00 to 01 to 10 to 11, back to 00 and repeats. When E=1 and F=0, the circuit goes through the state transitions from 00 to 11, to 10 to 01, back to 00 and repeats.

Question 6

Analyze the circuit below fully. Derive the Transition Table, Excitation Table, State Diagram and the Output. Explain the function of the circuit.



SOLUTION

COEN312, DEC4 2008

Q1.

a.

$$\begin{aligned} F(A,B,C,D) &= ABCD + (ABD)' + ABC'D \\ &= ABD(C+C') + (ABD)' = ABD + (ABD)' = 1 \end{aligned}$$

OR

$$\begin{aligned} &= ABCD + (ABD)' + ABC'D \\ &= BCD + A' + B' + D' + BC'D \\ &= CD + A' + B' + D' + C'D \\ &= C + A' + B' + D' + C'D \\ &= A' + B' + D' + C + D \\ &= 1 \end{aligned}$$

b.

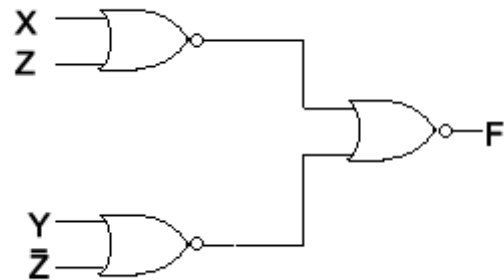
$$\begin{aligned} f(x,y,z) &= xy + xz' + yz \\ &= xy(z+z') + xz' + yz \\ &= \cancel{xyz} + \cancel{xyz'} + xz' + yz \\ &= xz' + yz \end{aligned}$$

i)

$$f =$$

Z \ XY	00	01	11	10
0	0	0	1	1
1	0	1	1	0

$$f = (x+z).(y+z')$$

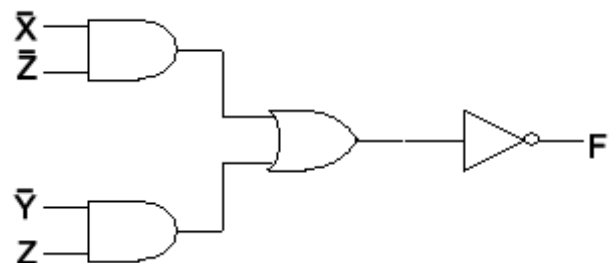


ii)

$$f =$$

Z \ XY	00	01	11	10
0	1	1	0	0
1	1	0	0	1

$$f = (x'z') + (y'z)$$



Q2.

Segments 'beg' has to be on in order to have 'L' on the seven-segments display.

	A	B	C	D		a	b	c	d	e	f	g
0	0	0	0	0		0	1	0	0	1	0	1
1	0	0	0	1		0	1	0	0	1	0	1
2	0	0	1	0		0	1	0	0	1	0	1
3	0	0	1	1		0	1	0	0	1	0	1
4	0	1	0	0		0	1	0	0	1	0	1
5	0	1	0	1		0	1	0	0	1	0	1
6	0	1	1	0		0	0	0	0	0	0	0
7	0	1	1	1		0	0	0	0	0	0	0
8	1	0	0	0		0	0	0	0	0	0	0
9	1	0	0	1		0	0	0	0	0	0	0

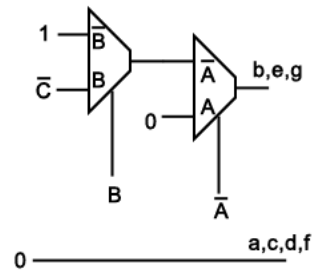
$$a=c=d=f=0$$

$$b=e=g = A'B' + A'C'$$

$$= A'(B' + C')$$

f=

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



Q3.

For the half-subtractor,					For the full-subtractor,																																																																																				
a.						b.																																																																																			
<table><tr><td>a</td><td>b</td><td></td><td>D</td><td>B</td></tr><tr><td>0</td><td>0</td><td></td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td></td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td></td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td></td><td>0</td><td>0</td></tr></table>					a	b		D	B	0	0		0	0	0	1		1	1	1	0		1	0	1	1		0	0	<table><tr><td>a</td><td>b</td><td>B₋</td><td></td><td>D</td><td>B</td></tr><tr><td>0</td><td>0</td><td>0</td><td></td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td></td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td></td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td></td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td></td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td></td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td></td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td></td><td>1</td><td>1</td></tr></table>						a	b	B ₋		D	B	0	0	0		0	0	0	0	1		1	1	0	1	0		1	1	0	1	1		0	1	1	0	0		1	0	1	0	1		0	0	1	1	0		0	0	1	1	1		1	1
a	b		D	B																																																																																					
0	0		0	0																																																																																					
0	1		1	1																																																																																					
1	0		1	0																																																																																					
1	1		0	0																																																																																					
a	b	B ₋		D	B																																																																																				
0	0	0		0	0																																																																																				
0	0	1		1	1																																																																																				
0	1	0		1	1																																																																																				
0	1	1		0	1																																																																																				
1	0	0		1	0																																																																																				
1	0	1		0	0																																																																																				
1	1	0		0	0																																																																																				
1	1	1		1	1																																																																																				
$D = (a \oplus b)$ $B = \overline{ab}$																																																																																									

The K-map tables give:

$$D =$$

$b_ \backslash ab$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$D = (a \oplus b \oplus b_)$$

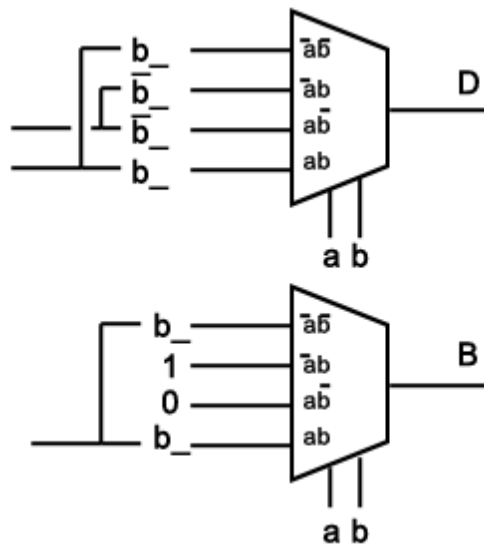
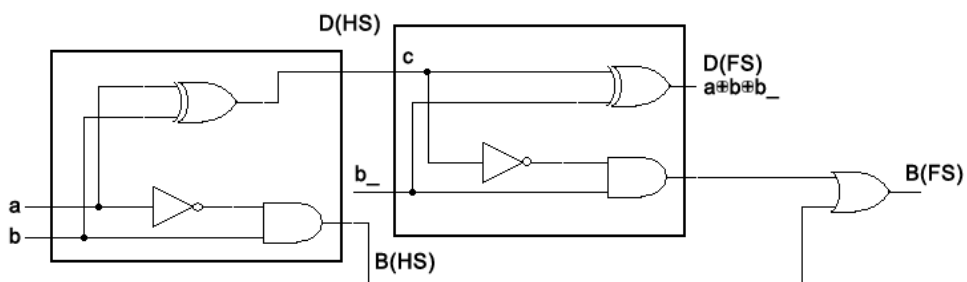
$$= \bar{a}\bar{b}\bar{b_} + \bar{a}b\bar{b_} + a\bar{b}\bar{b_} + abb_$$

$$B =$$

$b_ \backslash ab$	00	01	11	10
0	0	1	0	0
1	1	1	1	0

$$D = \bar{a}b + b_ (a \oplus b)$$

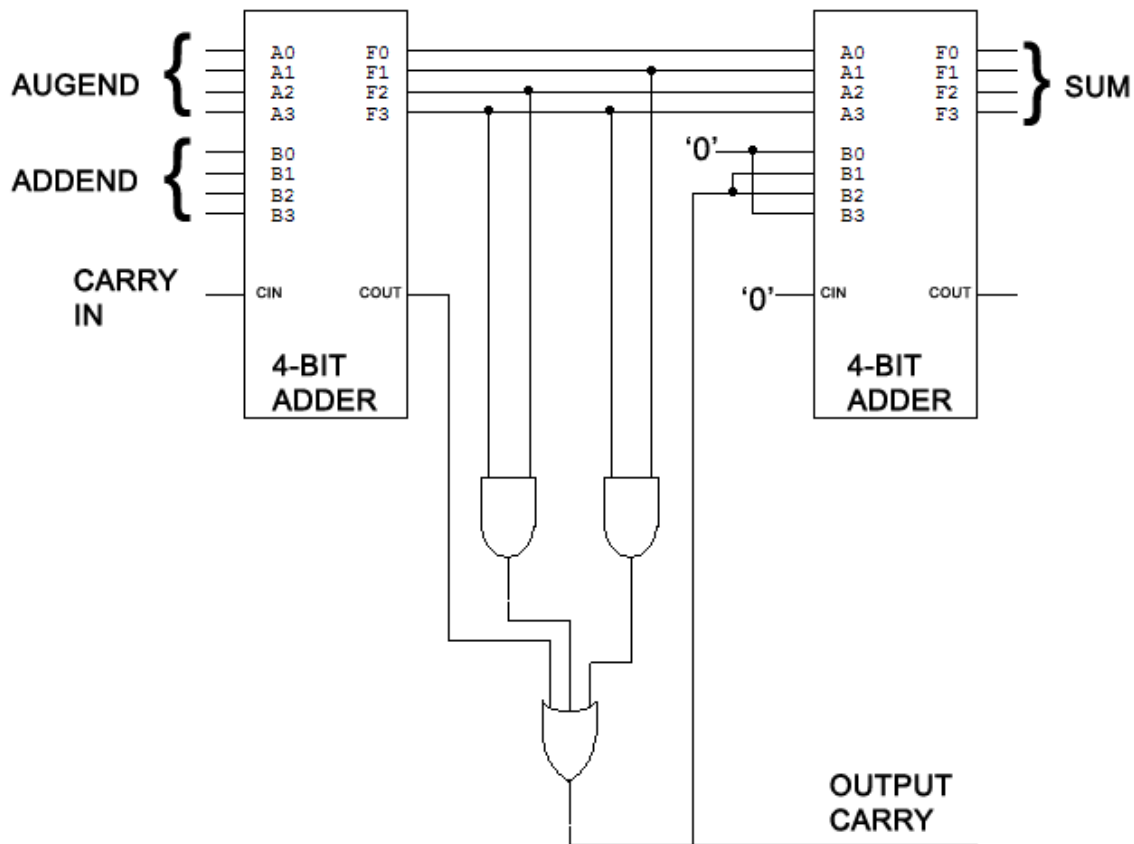
$$= \bar{a}b + b_ ab + b_ \bar{a}b$$



Q4.

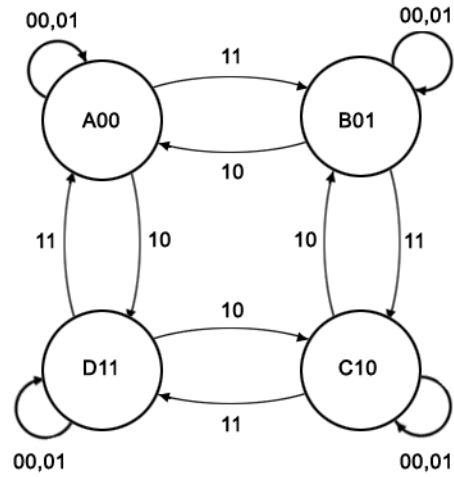
The algorithm to do binary addition of two BCD numbers, with the result in BCD, involves a binary adder to do the actual arithmetic operation along with another adder to offset the result whenever the binary sum exceeds the BCD range, which is from 0 to 9. When this happens, the addition is scaled up by 6. The operation has to take care of the carry-out which will be asserted whenever the BCD range is exceeded or the actual binary addition generates a carry. The Boolean expression for the carry bit is:

$$C = K + Z_8Z_4 + Z_8Z_2$$



Q5.

State Diagram:



State Transition Table:

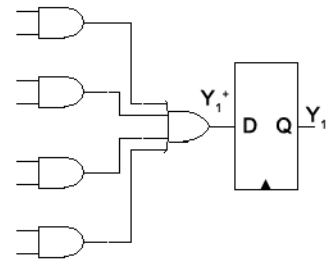
Present	Next			
Y1 Y0	EF 00	01	10	11
00	00	00	11	01
01	01	01	00	10
10	10	10	01	11
11	11	11	10	00

K-Maps for next state equations:

$$Y_1^+ = y_1 y_0 \backslash EF$$

	00	01	11	10
00	0	0	0	1
01	0	0	1	0
11	1	1	0	1
10	1	1	1	0

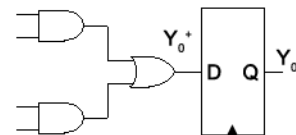
$$Y_1^+ = \bar{E}y_1 + Fy_1\bar{y}_0 + \bar{F}y_1y_0 + EF\bar{y}_1y_0 + E\bar{F}\bar{y}_1\bar{y}_0$$



$$Y_0^+ = y_1 y_0 \backslash EF$$

	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$Y_0^+ = \bar{E}y_0 + E\bar{y}_0$$



Q6.

Analysis:

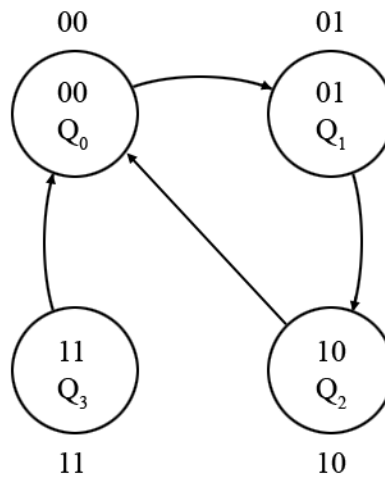
$$T_A = Q_A + Q_B$$

$$T_B = \overline{Q_A} + Q_B$$

State transition table:

		CLK		NEXT STATE	
Q_A	Q_B	T_A	T_B	Q_A^+	Q_B^+
0	0	0	1	0	1
0	1	1	1	1	0
1	0	1	0	0	0
1	1	1	1	0	0

State diagram:



Outputs are of the state itself.

This circuit is a counter “00”->“01”->“10” and back to “00”..., if ever started in “11” state, then the next state on the pulse will set state to “00”, “01” and so on.