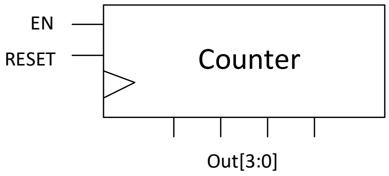
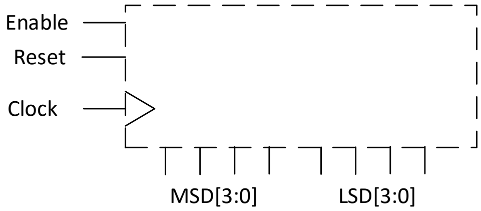
Q1. Using logic gates and two 4-bit counters shown below, design a two-digit saturating Binary Coded Decimal (BCD) counter that counts from 00 to 99.



It should have three inputs: clock, reset and enable. Name the outputs corresponding to the least significant BCD digit as LSD[3:0] and most significant BCD digit as MSD[3:0], i.e. you should design the circuit inside the dashed box using the counters given above



On each rising edge of the clock:

- If reset is a 1, your BCD counter should go to 00.
- Otherwise, if Enable=1, it should increment to the next value (staying at 99 if already at 99)
- Otherwise, it should hold its value.

$ENL = enable \, . \, \overline{MSD3.MSD0.LSD3.LSD0}$

$RESETL = LSD3.LSD0 \, . \, \overline{MSD3.MSD0} + reset$

$ENM = enable.LSD3.LSD0.\overline{MSD3.MSD0}$

$RESETM = reset.$

Let: $X = LSD3.LSD0$

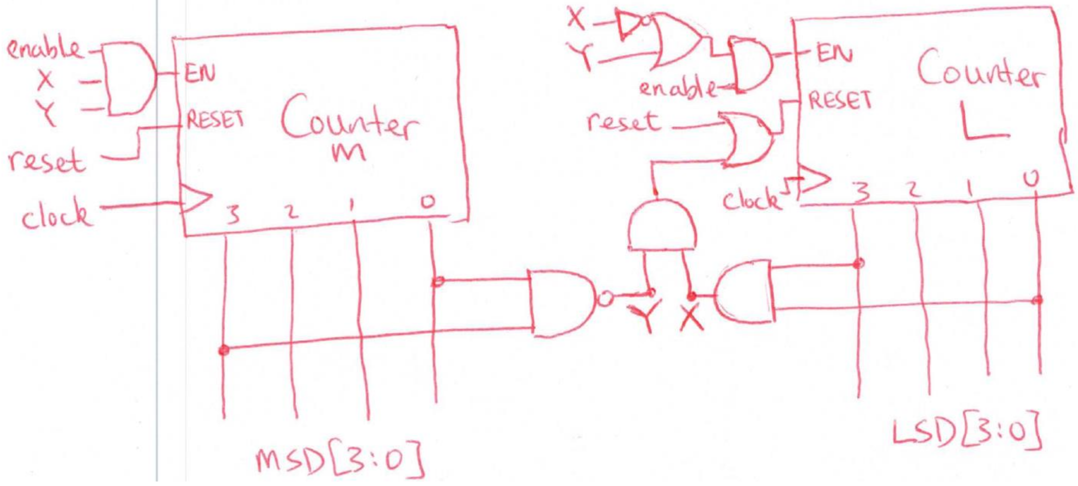
$Y = \overline{MSD3.MSD0}$

$ENL = enable.(\overline{X}+Y)$

$RESETL = reset + X.Y$

$ENM = enable.X.Y$

$RESETM = reset.$



Q2. Implement the following Boolean functions with PAL:

a. Find the corresponding minterms
b. Draw thw PAL programming table
c. Draw the fuse map.

$$w = \sum(3,4,5,6,7,12,13,14,15)$$

$$x = \sum(6,8,9)$$

$$y = \sum(0,2,3,4,5,6,7,11,12,13,14,15)$$

$$z = \sum(5,6,8,9,13)$$

Soln:

a. We have

$$w = B + A'CD$$

$$x = A'BCD' + AB'C'$$

$$y = B + CD + A'D'$$

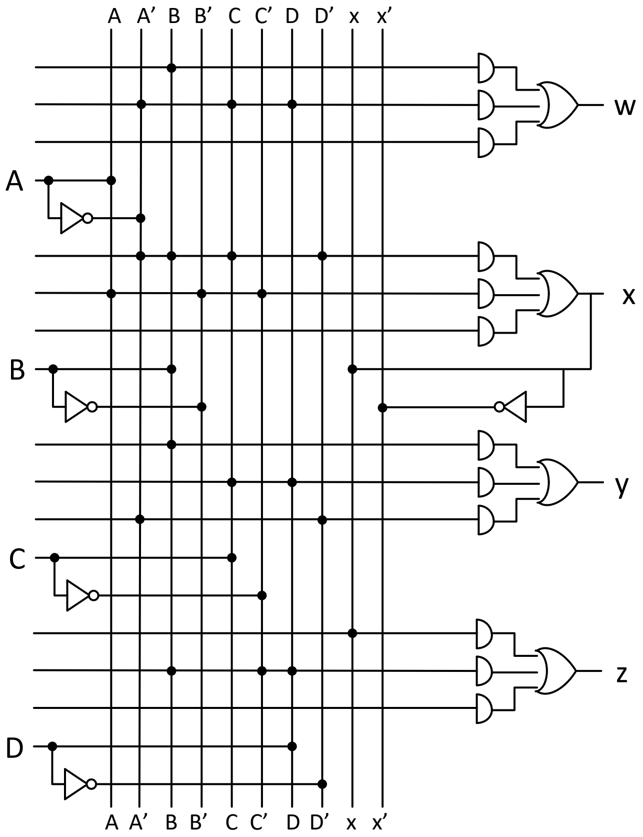
$$z = A'BCD' + AB'C' + BC'D$$

$$= x + BC'D$$

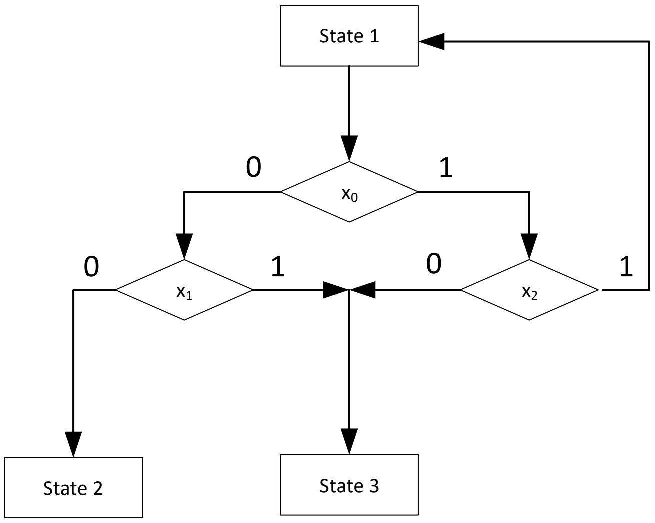
b.

Product	AND Inputs					Outputs
Term	A	B	C	D	x	
1	-	1	-	-	-	w=B+A’CD
2	0	-	1	1	-	
3	-	-	-	-	-	
4	0	1	1	0	-	x=A’BCD’+AB’C’
5	1	0	0	-	-	
6	-	-	-	-	-	
7	-	1	-	-	-	y=B+CD+A’D’
8	-	-	1	1	-	
9	0	-	-	0	-	
10	-	-	-	-	1	z=x+BC’D
11	-	1	0	1	-	
12	-	-	-	-	-	

c.



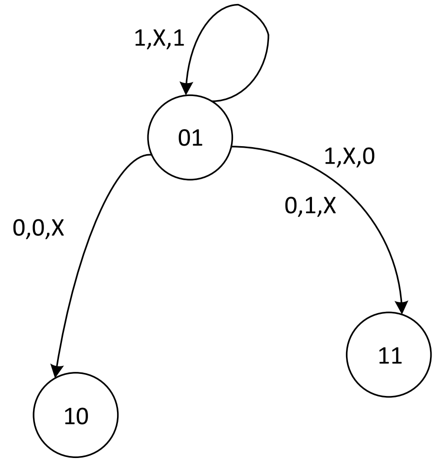
Q3. Draw the state table and the state diagram for the given ASM chart.



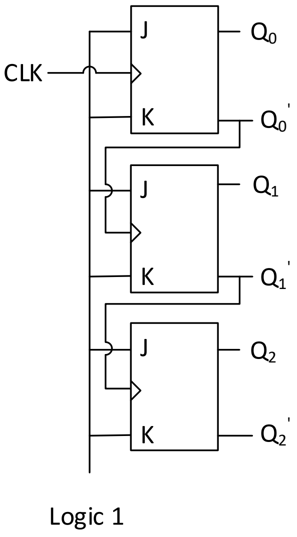
Soln:

Present State	X_0	X_1	X_2	Next state
01	0	0	0	10
01	0	0	1	10
01	0	1	0	11
01	0	1	1	11

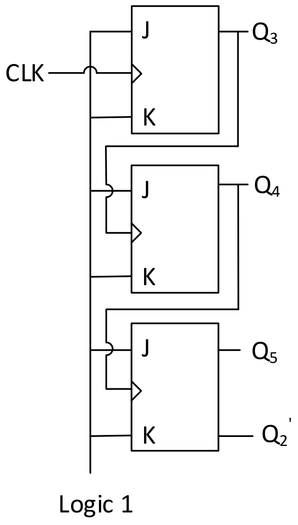
01	1	0	0	11
01	1	0	1	01
01	1	1	0	11
01	1	1	1	01



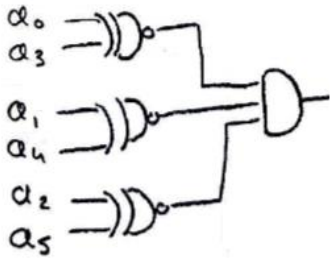
Q4. Design an asynchronous 3 bits down counter and a synchronous 3 bits up counter separately only using J-K flip flops in both (totally 2 separated counters will be designed). Then design a circuit that gives alert at any time when the outputs of these two circuits are exactly same.



Up-counter



Down-counter



Q5. The outputs of four registers *R0, R1, R2, R3*, are connected through 4-to-1-line multiplexers to the inputs of a fifth register, *R5*. Each register is 8-bits. The required transfers are dictated by four timing variables T_0 through T_3 as follows:

$$T_o:\ R5 \leftarrow R0$$

$$T_1:\ R5 \leftarrow R1$$

$$T_2:\ R5 \leftarrow R2$$

$$T_3:\ R5 \leftarrow R3$$

The timing variables are mutually exclusive, that is, only one may be equal to 1 at any given time. Draw a block diagram showing the hardware implementation of the register transfers. Include the connections necessary from the four timing variables to the selection inputs of the multiplexers and to the load input of *R5*.

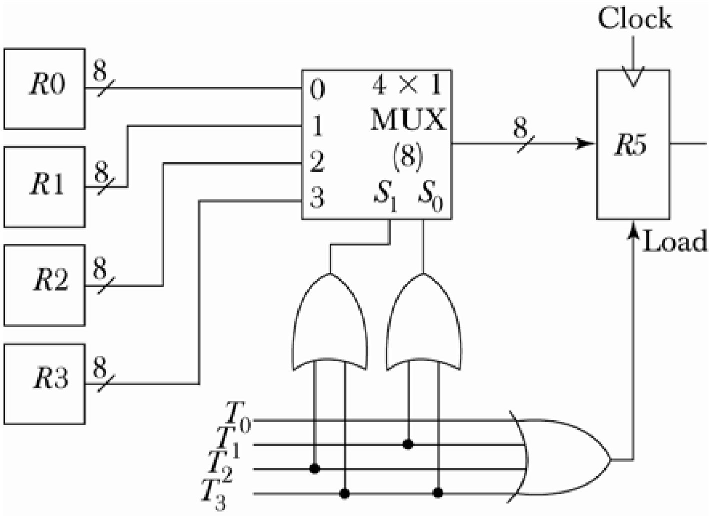
<u>Soln:</u>

$T_0 T_1 T_2 T_3$	$S_1 S_0$	load
0 0 0 0	X X	0
1 0 0 0	0 0	1
0 1 0 0	0 1	1
0 0 1 0	1 0	1
0 0 0 1	1 1	1

$S_1 = T_2 + T_3$

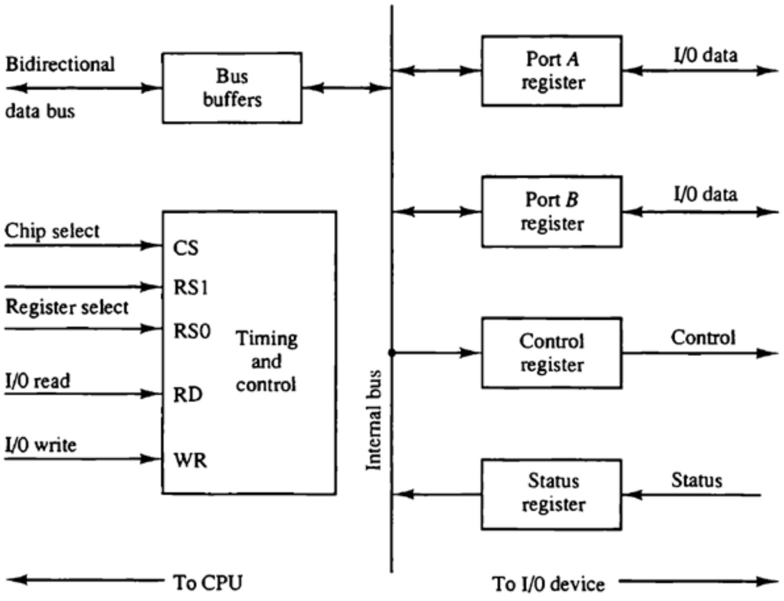
$S_0 = T_1 + T_3$

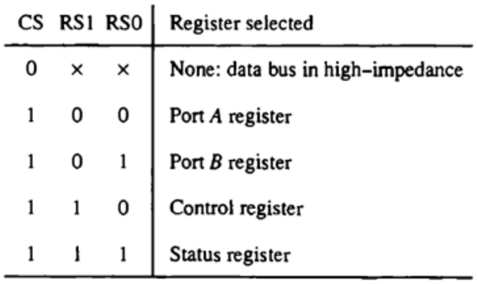
$\text{load} = T_0 + T_1 + T_2 + T_3$



Q6. The addresses assigned to the four registers of the I/O interface of the system shown below are equal to the binary equivalent of 12, 13, 14, and 15.

a. Show the external circuit that must be connected between an 8-bit I/O address from the CPU and the *CS*, *RS1* and *RS0* inputs of the interface.





CS	RS1	RS0	Register selected
0	×	×	None: data bus in high–impedance
1	0	0	Port *A* register
1	0	1	Port *B* register
1	1	0	Control register
1	1	1	Status register

Soln:

$$\begin{array}{rcll} & & \underline{A_1 — A_2} & \underline{A_j\,A_0} \\ 12 & = & 000011 & 0\;0 \\ 13 & = & 000011 & 0\;1 \\ 14 & = & 000011 & 1\;0 \\ 15 & = & \underbrace{000011}_{\text{To CS}} & \underset{\substack{\nearrow \\ \text{RSI}}}{1}\;\underset{\substack{\nwarrow \\ \text{RSO}}}{1} \end{array}$$

$$\begin{aligned} CS &= A_2 A_3 A'_4 A'_5 A'_6 A'_7 \\ RS1 &= A_1 \\ RS0 &= A_0 \end{aligned}$$

b. Six interface of the type shown in the figure are connected to a CPU that uses an I/O address of 8-bits. Each of the six *CS* inputs is connected to a different address line. The two low order address lines are connected to the *RS1* and *RS0* of all six interface units. Determine the 8-bit address of each register in each interface.

Soln

Interface	Port A	Port B	Control Reg	Status Reg
≠ 1	10000000	10000001	10000010	10000011
2	01000000	01000001	01000010	01000011
3	00100000	00100001	00100010	00100011
4	00010000	00010001	00010010	00010011
5	00001000	00001001	00001010	00001011
6	00000100	00000101	00000110	00000111

Q7. a. How many 128 x 8 RAM chips are needed to provide a memory capacity of 2048 bytes?

b. How many lines of the address bus must be used to access 2048 bytes of memory?

c. How many of these lines will be common to all chips?

d. How many lines must be decode for chip select?

e. Specify the size of the decoders.

Soln:

$$\frac{2048}{128} = 16\,\text{chips}$$

$2048 = 2^{11}$ 11 lines to address 2078 bytes.

$128 = 2^{7}$ 7 lines to address each chip

4 lines to decoder for selecting 16 chips

4 × 16 decoder

Q8. A computer employs RAM chips of 256 x 8 and ROM chips of 1024 x 8. The computer system needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped I/O configuration is used. The two highest-order bits of the address bus are assigned 00 for RAM and 10 for interface registers.

a. How many RAM and ROM chips are needed?

b. Draw a memory-address map for the system.

c. Give the address range in hexadecimal for RAM, ROM and interface.

Soln:

RAM	2048 /256 = 8 chips;	$2048 = 2^{11}$;	$256 = 2^{8}$
ROM	4096 /1024 = 4 chips;	$4096 = 2^{12}$;	$1024 = 2^{10}$
Interface	4 × 4 = 16 registers; $16 = 2^{4}$		

Component	Address	16 15 14 13	12 11 10 9	8 7 6 5	4 3 2 1
RAM	0000–07FF	0 0 0 0	0 ←→ 3×8 decoder	x x x x	x x x x
ROM	4000–4FFF	0 1 0 0	←→ 2×4 decoder x x	x x x x	x x x x
Interface	8000–800F	1 0 0 0	0 0 0 0	0 0 0 0	x x x x