Gayatri Vidya Parishad College of Engineering for Women (Autonomous)

Madhurawada, Visakhapatnam

(Affiliated to Andhra University, Visakhapatnam)

I B.Tech. II Semester – Regular Examinations, June/ July – 2025

Subject Name:	Digital Logic Design [24EC11RC05]	Max. Marks: 70 M
Faculty Name :	P V K Chaitanya	Branches: ECE, CSE, CSE(AIML), IT

ANSWER KEY

1 a)

16 | 2365

16 | 147 — 13 (D)

16 | 9 — 3

0 — 9

$$(2365)_{10} = (93D)_{16}$$

8 | 2365

8 | 295 — 5

8 | 36 — 7

8 | 4 — 4

0 — 4

$$(2365)_{10} = (4475)_8$$

$\underbrace{0010}_{2}\ \underbrace{1010}_{A}\ \underbrace{1010}_{A}\ \underbrace{1111}_{F}$

$$(1010\,1010\,1111)_2 = (2AAF)_{16}$$

$\underbrace{010}_{2}\ \underbrace{101}_{5}\ \underbrace{010}_{2}\ \underbrace{101}_{5}\ \underbrace{111}_{7}$

$$(10101010101111)_2 = (25257)_8$$

1 b)

given data word is 11110000

$\therefore m = 8$

if no: of parity bits is 'p', then

$$2^p \geq x+p+1$$

if $p=4$, $2^4 \geq 8+4+1$

$16 \geq 13$ TRUE

$\therefore$ no: of parity bits to be added is 4

1 1 1 1 0 0 0 0

bit location table

bit designation	bit location	binary representation	bit values
P_1	1	0001	—
P_2	2	0010	—
D_3	3	0011	1
P_4	4	0100	—
D_5	5	0101	1
D_6	6	0110	1
D_7	7	0111	1
P_8	8	1000	—
D_9	9	1001	0
D_{10}	10	1010	0
D_{11}	11	1011	0
D_{12}	12	1100	0

using even parity

$P_1 \rightarrow 1, 3, 5, 7, 9, 11$
$\quad$ _ 1 1 1 0 0 $\longrightarrow$ Odd no: of 1's $\rightarrow P_1 = 1$

$P_2 \rightarrow 2, 3, 6, 7, 10, 11$
$\quad$ _ 1 1 1 0 0 $\longrightarrow$ Odd no: of 1's $\rightarrow P_2 = 1$

$P_4 \rightarrow 4, 5, 6, 7, 12$
$\quad$ _ 1 1 1 0 $\longrightarrow$ Odd no: of 1's $\rightarrow P_4 = 1$

$P_8 \rightarrow 8, 9, 10, 11, 12$
$\quad$ _ 0 0 0 0 $\longrightarrow$ Even no: of 1's $\rightarrow P_8 = 0$

The 12-bit hamming code is 111111100000

2 a)

28 − 78

28 → 0001 1100

78 → 0100 1110

↓ 1's

1011 0001

+1

−78 $\xrightarrow{\text{2's Comp.}}$ 1011 0010

28 →	0001 1100
+(−78)	+ 1011 0010
	1 1001 1110

MSB bit is '1'. So, the answer is negative.

To get the magnitude, find 2's complement of result

1100 1110

↓ 1's

0011 0001

+1

0011 0010

$(00110010)_2 = (50)_{10}$

∴ Answer is −50

128 − 456

128 → 0010000000

456 → 0111001000

↓ 1's

1000110111

+1

−456 $\xrightarrow{\text{2's Comp.}}$ 1000111000

128 →	00 1000 0000
+(−456) →	+ 10 0011 1000
	1 0 1011 1000

MSB bit is '1'. So the answer is negative.

To get the magnitude, find 2's complement of result

10 1011 1000

↓ 1's

01 0100 0111

+1

01 0100 1000

$(0101001000)_2 = (328)_{10}$

∴ Answer is −328.

2 b)

Decimal	BCD	Excess-3	5421
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0010
3	0011	0110	0011
4	0100	0111	0100
5	0101	1000	1000
6	0110	1001	1001
7	0111	1010	1010
8	1000	1011	1011
9	1001	1100	1100

From the adjacent Table

98457

$= (1001\,1000\,0100\,0101\,0111)_{BCD}$

$= (1100\,1011\,0111\,1000\,1010)_{Excess\text{-}3}$

$= (1100\,1011\,0100\,1000\,1010)_{5421}$

a) $F(A,B,C,D) = \sum(2,3,12,13,14,15)$

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

$\to \bar{A}\bar{B}C$

$\to AB$

$\therefore F = AB + \bar{A}\bar{B}C$

3 b) $BDE + \bar{B}\bar{C}D + CDE + \bar{A}\bar{B}CE + \bar{A}\bar{B}C + \bar{B}\bar{C}\bar{D}E$

$= BDE + \bar{B}\bar{C}(D+\bar{D}E) + CDE + \bar{A}\bar{B}C(1+E)$

$A + \bar{A}B = \cancel{AB}\ A+B$

$1 + A = 1$

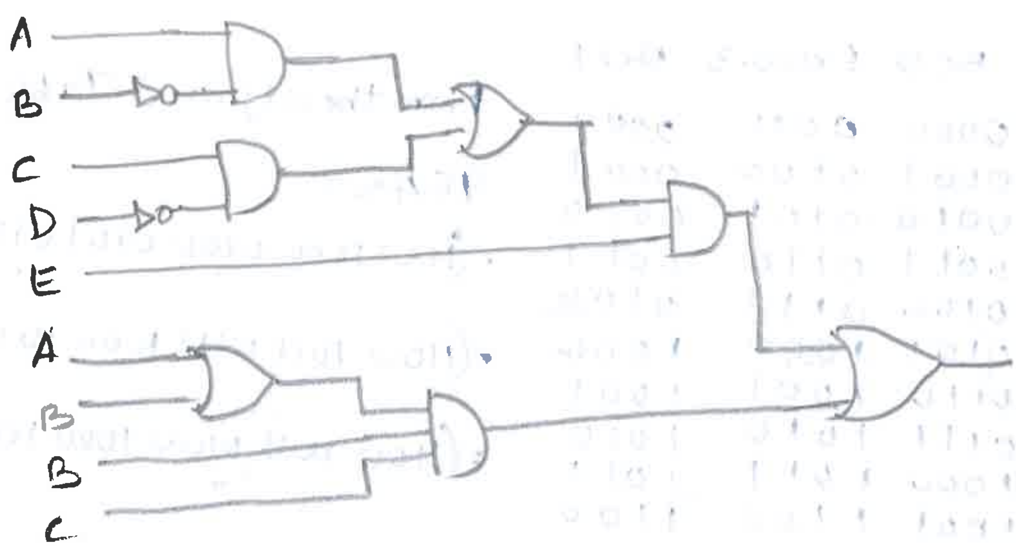
$= BDE + \bar{B}\bar{C}(D+E) + CDE + \bar{A}\bar{B}C$

$= BDE + CDE + \bar{B}\bar{C}D + \bar{B}\bar{C}E + \bar{A}\bar{B}C$

4 a) $(A\bar{B} + C\bar{D})E + BC(A+B)$

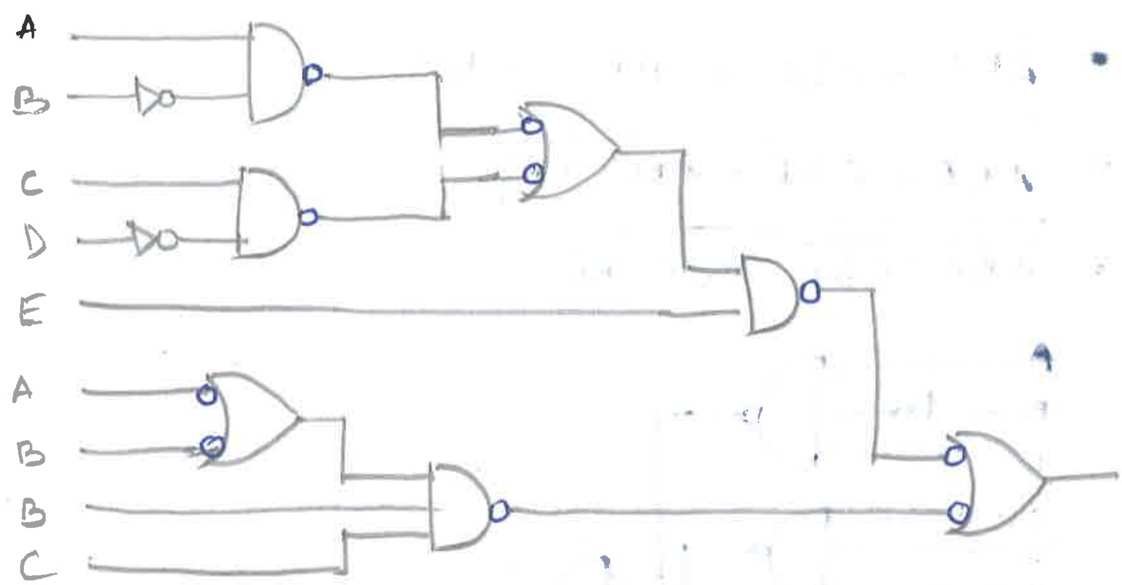
Step 1

Draw AOI Logic



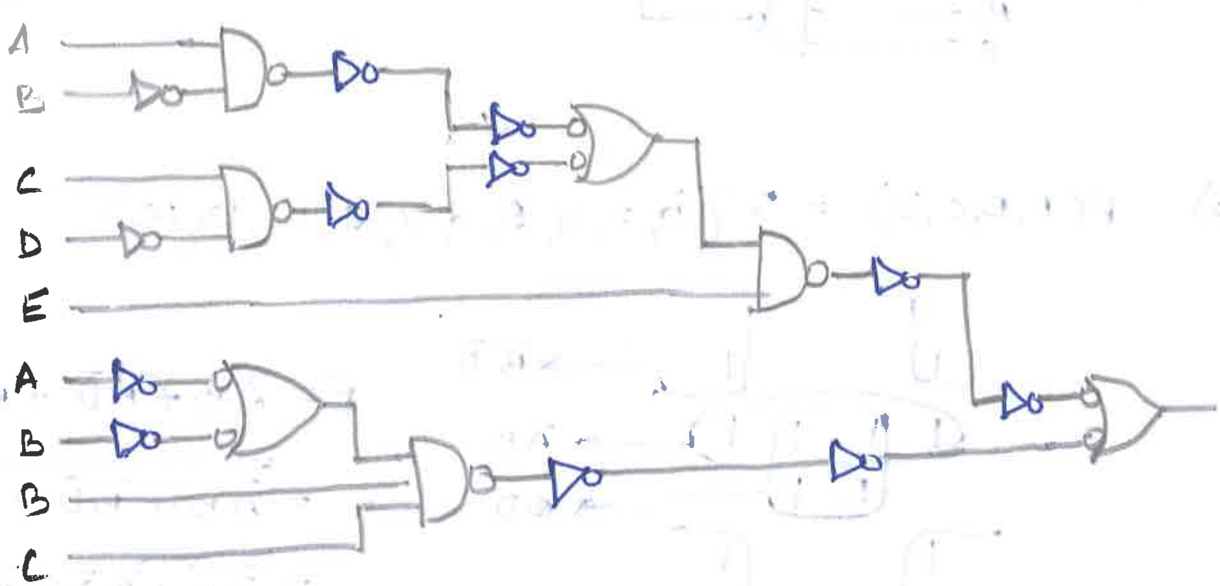
Step 2

Add bubbles at i/p of OR gate and o/p of AND gate



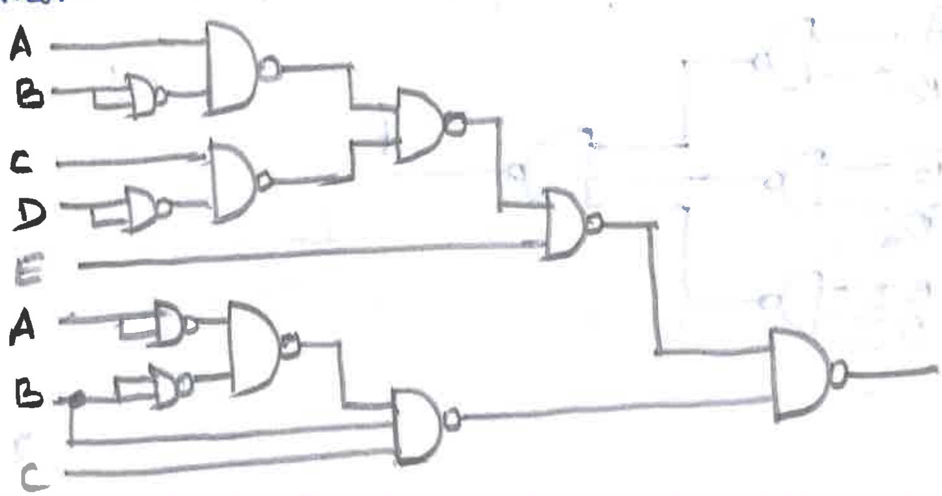
Step 3

Add an inverter on each line that received a bubble



Step 4

Remove double inversions & replace bubbled OR with NAND



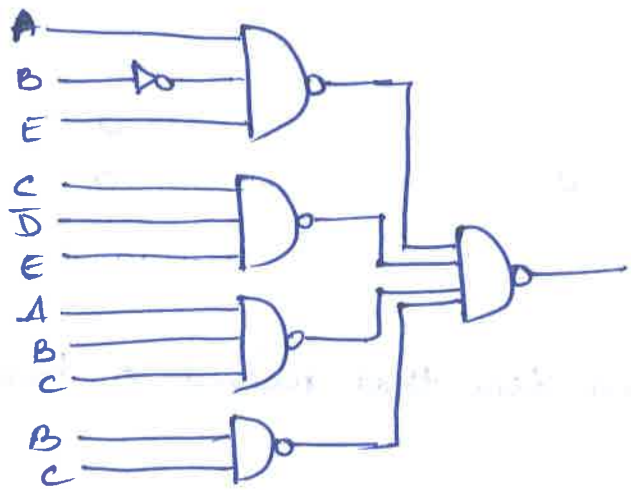
Alternate Approach

$$(A\bar{B} + C\bar{D})E + BC(A + B)$$

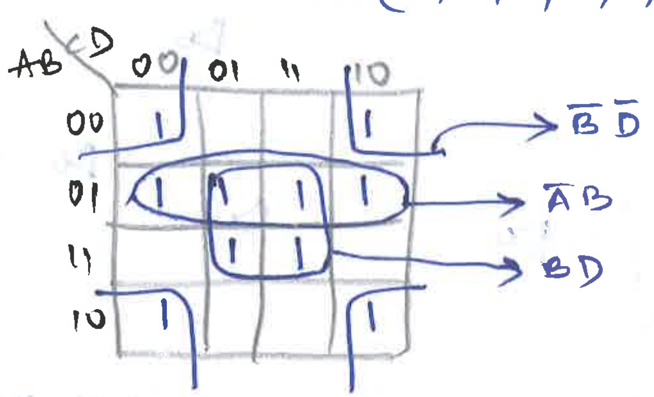
$$\bullet\ A\bar{B}E + C\bar{D}E + ABC + BC$$

$$= \overline{\overline{A\bar{B}E + C\bar{D}E + ABC + BC}}$$

$$= \overline{\overline{A\bar{B}E} \cdot \overline{C\bar{D}E} \cdot \overline{ABC} \cdot \overline{BC}}$$



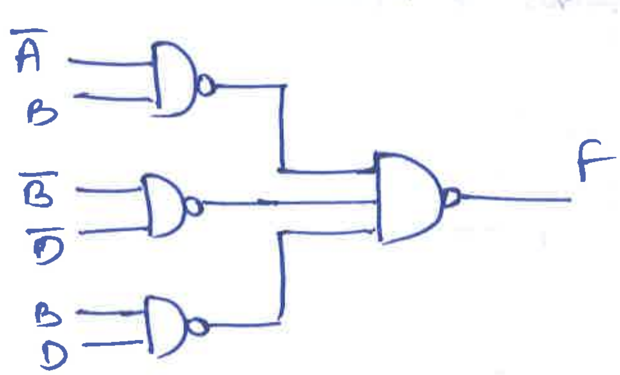
4 b) $F(A,B,C,D) = \sum(0,2,4,5,6,7,8,10,13,15)$



$$F = \bar{A}B + \bar{B}\bar{D} + BD$$

$$= \overline{\overline{\bar{A}B + \bar{B}\bar{D} + BD}}$$

$$= \overline{\overline{\bar{A}B} \cdot \overline{\bar{B}\bar{D}} \cdot \overline{BD}}$$



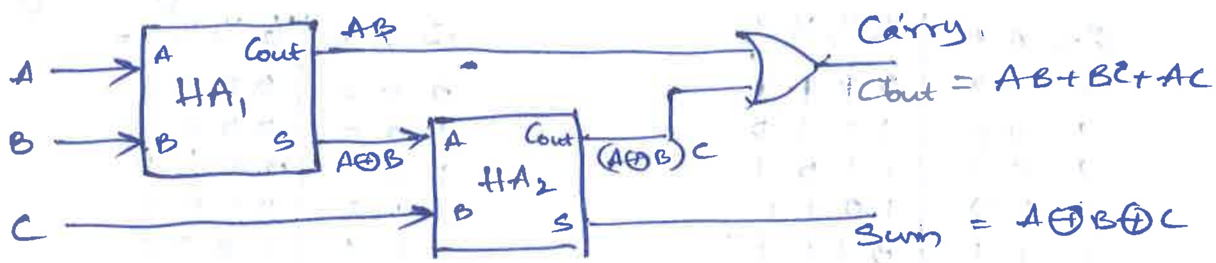
a) A combinational logic circuit is formed by the combination of different logic gates.
It is a type of digital circuit where the output is solely determined by the current combination of the inputs.

Full Adder

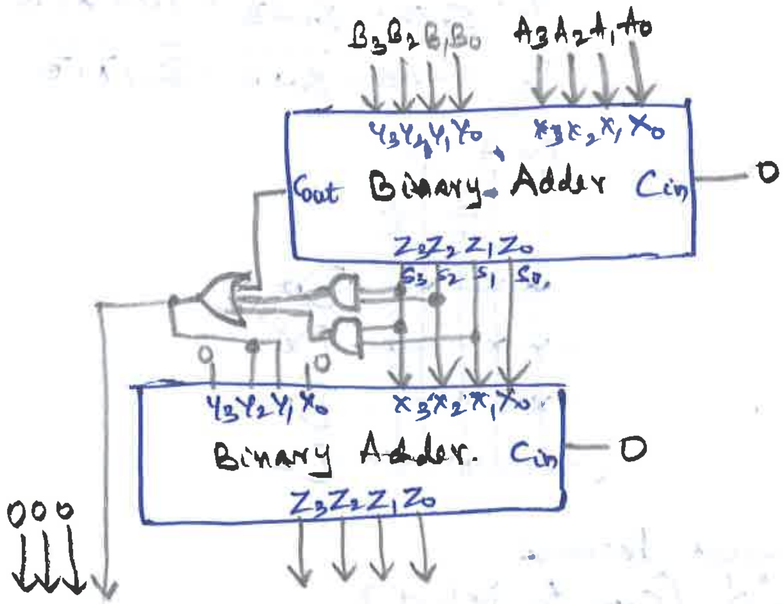
A	B	C	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$Sum = A \oplus B \oplus C$$

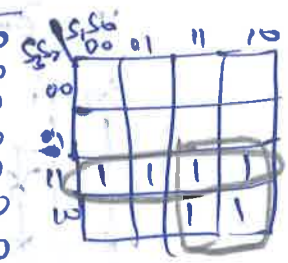
$$C_{out} = AB + BC + AC$$



5 b) BCD Adder



$S_3S_2S_1S_0$	Y
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	0
0111	0
1000	0
1001	0
1010	1
1011	1
1100	1
1101	1
1110	1
1111	1



$S_3S_2 + S_3S_1$

if c=1 or if result is invalid, then add 0110
if C=0 and result is valid BCD, then add 0000

a)

4-bit binary parallel Adder / Subtractor

Control signal

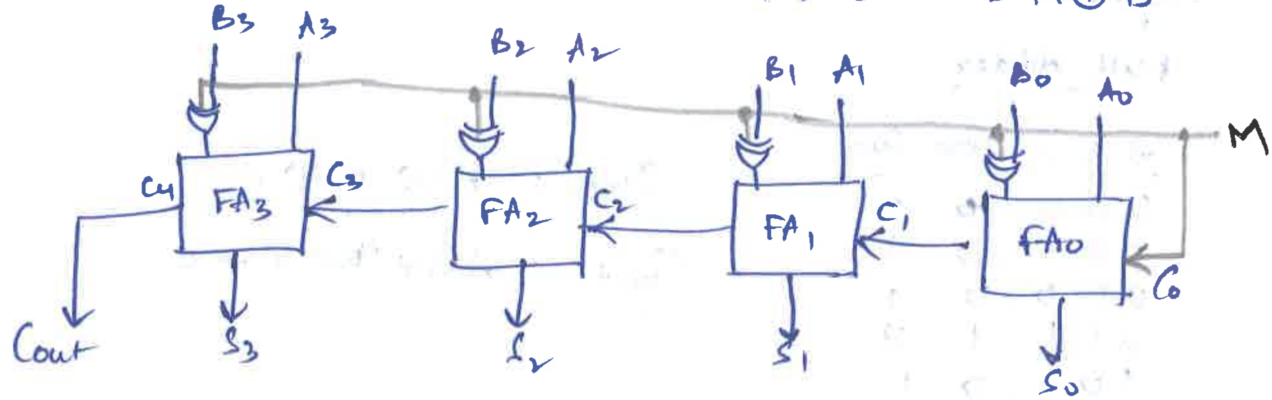
M=0 → Addition

M=1 → Subtraction

M	Y
0	B
1	$\bar{B}$

$Y = \bar{M}B + M\bar{B}$

$= M \oplus B$



6b)

2:4 Decoder using NAND gates

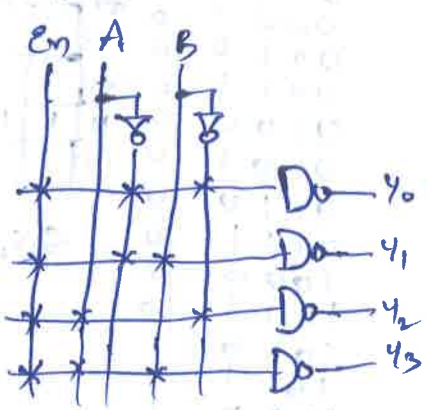
En	A	B	Y3	Y2	Y1	Y0
0	x	x	Φ	Φ	Φ	Φ
1	0	0	1	1	1	0
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1

$Y_0 = \overline{En\,\bar{A}\,\bar{B}}$

$Y_1 = \overline{En\,\bar{A}\,B}$

$Y_2 = \overline{En\,A\,\bar{B}}$

$Y_3 = \overline{En\,A\,B}$



2:4 Decoder using NOR gates

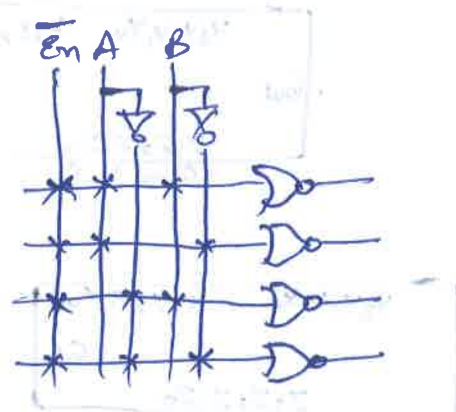
En	A	B	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$Y_0 = En\,\bar{A}\,\bar{B} = \overline{\overline{En\,\bar{A}\,\bar{B}}} = \overline{\overline{En}+A+B}$

$Y_1 = En\,\bar{A}\,B = \overline{\overline{En\,\bar{A}\,B}} = \overline{\overline{En}+A+\bar{B}}$

$Y_2 = En\,A\,\bar{B} = \overline{\overline{En\,A\,\bar{B}}} = \overline{\overline{En}+\bar{A}+B}$

$Y_3 = En\,A\,B = \overline{\overline{En\,A\,B}} = \overline{\overline{En}+\bar{A}+\bar{B}}$



NAND realization is convenient because

i) Selected o/p becomes '0' (Easy to Detect)

ii) NAND logic gives this behaviour naturally

iii) NAND gates are faster & cheaper

7 a)

SR flip flop to JK flip flop

Inputs J K	Present State Q_n	Next State Q_{n+1}	Flip flop Excitations S R
0 0	0	0	0 x
0 0	1	1	x 0
0 1	0	0	0 x
0 1	1	0	0 1
1 0	0	1	1 0
1 0	1	1	x 0
1 1	0	1	1 0
1 1	1	0	0 1

for S

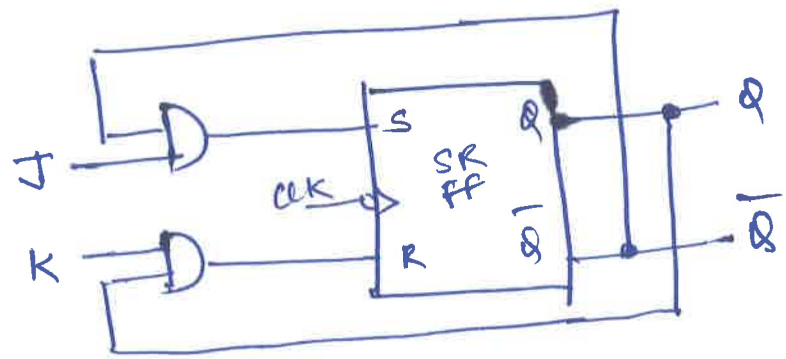
J \ KQ_n	00	01	11	10
0		X		
1	1	X		1

$S = J\overline{Q}_n$

for R

J \ KQ_n	00	01	11	10
0	X		1	X
1			1	

$R = KQ_n$



7 b)

$m = 1 \longrightarrow$ Right Shift

$m = 0 \longrightarrow$ Left Shift

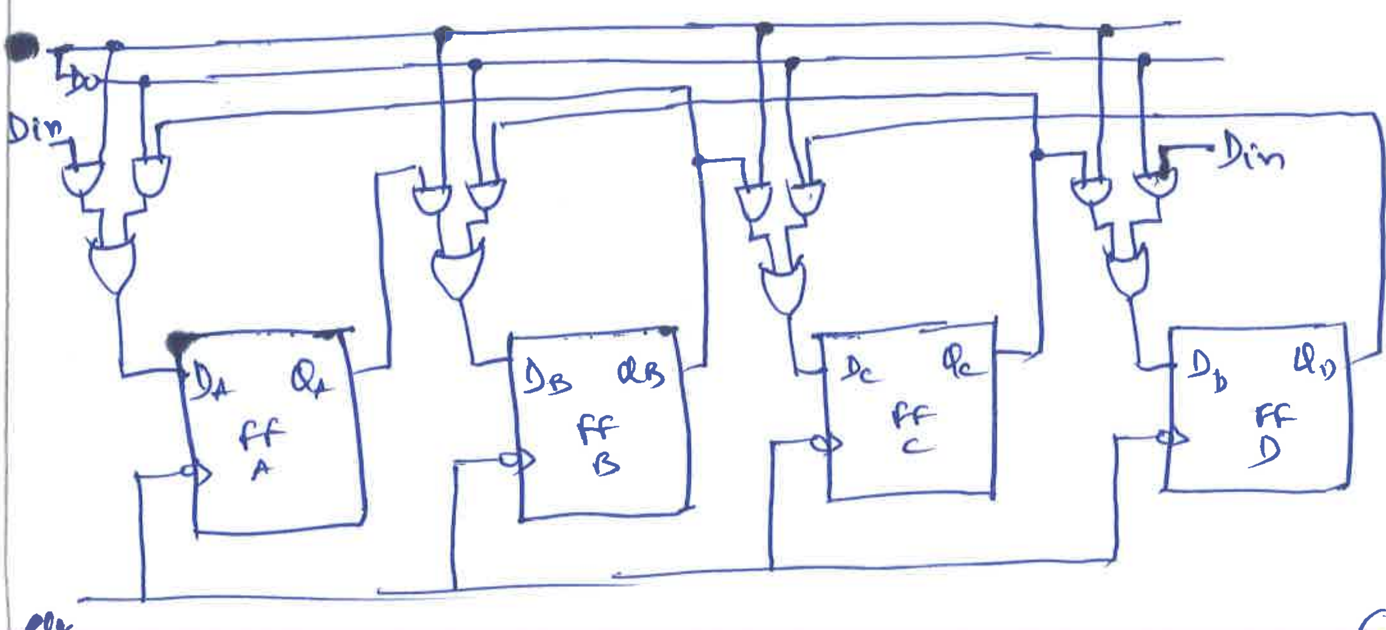
m	D_A	D_B	D_C	D_D	D_{out}
1	D_{in}	Q_A	Q_B	Q_C	Q_D
0	Q_B	Q_C	Q_D	D_{in}	Q_A

$D_A = mD_{in} + \overline{m}Q_B$

$D_B = mQ_A + \overline{m}Q_C$

$D_C = mQ_B + \overline{m}Q_D$

$D_D = mQ_C + \overline{m}D_{in}$



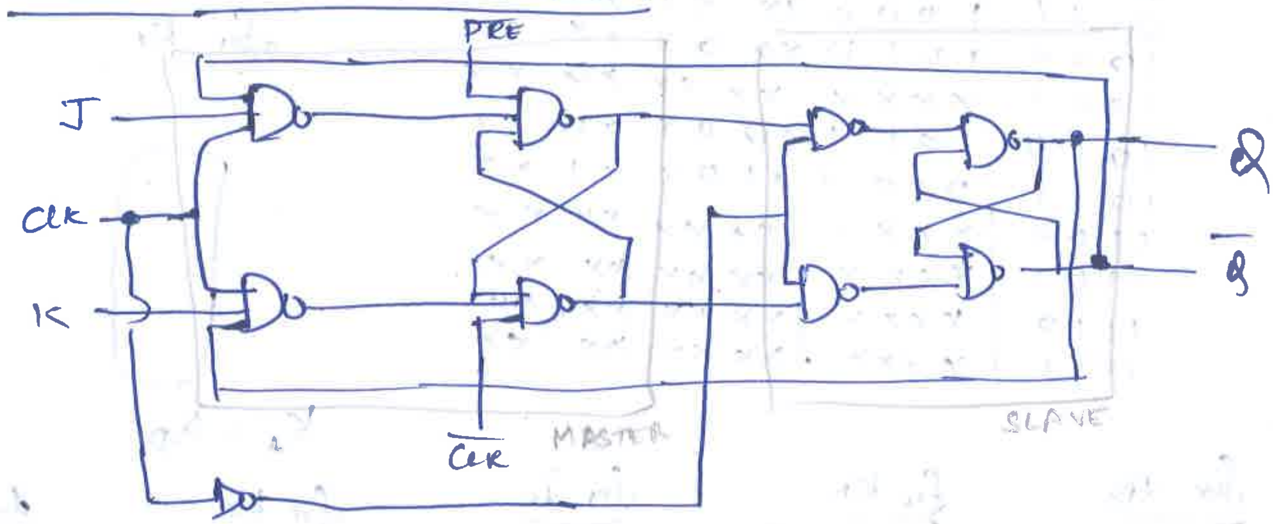
8.

Drawback of JK flipflop

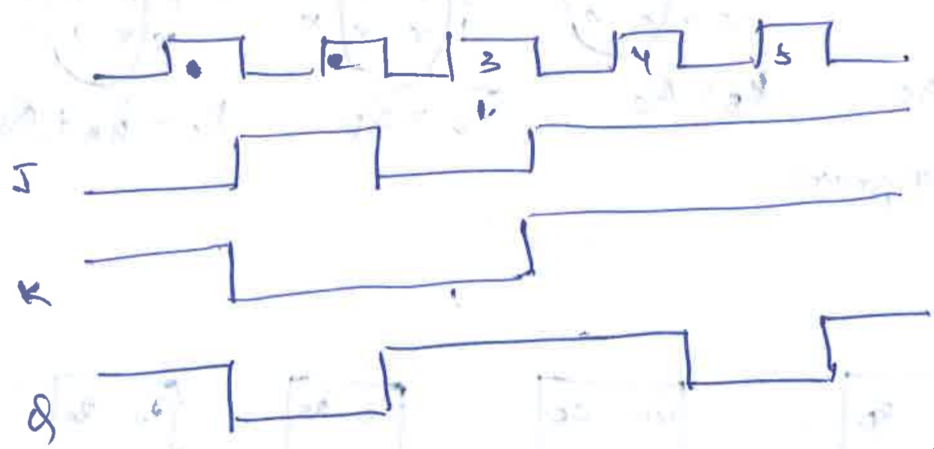
For JK flipflop, if J = K = 1, then the flipflop keep on changing from 0 to 1, 1 to 0, 0 to 1, ... and so, at the end of clock pulse, its state will be uncertain. This condition is called race-around condition.

- To overcome this problem, we need to use Master-Slave Flip flop.

Master - Slave JK flip flop



The operation would be same as negative edge-triggered JK flip flop.



Output of Master is given to the input of the slave. Clock of Master is inverted and given to the clock of the slave.

1. Counter to count states 1, 2, 4, 5, 7, 8, 10, 11, 1, ...

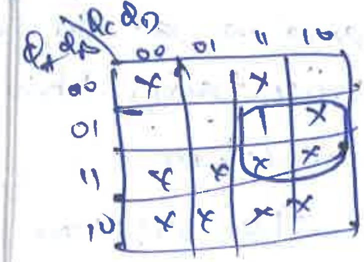
no. of states is 9.

∴ We require 4 flipflops.

Excitation Table of Counter

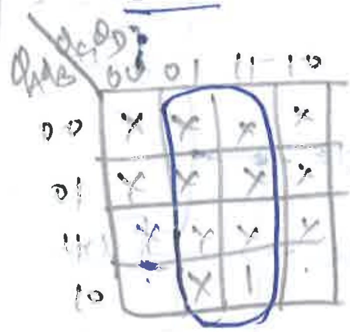
Present State $Q_A Q_B Q_C Q_D$	Next State $Q_{A+1} Q_{B+1} Q_{C+1} Q_{D+1}$	J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D
0 0 0 0	x x x x	x	x	x	x	x	x	x	x
0 0 0 1	0 0 1 0	0	x	0	x	1	x	x	1
0 0 1 0	0 1 0 0	0	x	1	x	x	1	0	x
0 0 1 1	x x x x	x	x	x	x	x	x	x	x
0 1 0 0	0 1 0 1	0	x	x	0	0	x	1	x
0 1 0 1	0 1 1 1	0	x	x	0	1	x	x	0
0 1 1 0	x x x x	x	x	x	x	x	x	x	x
0 1 1 1	1 0 0 0	1	x	x	1	x	1	x	1
1 0 0 0	1 0 1 0	x	0	0	x	1	x	0	x
1 0 0 1	x x x x	x	x	x	x	x	x	x	x
1 0 1 0	1 0 1 1	x	0	0	x	x	0	1	x
1 0 1 1	0 0 0 1	x	1	0	x	x	1	x	0
1 1 0 0	x x x x	x	x	x	x	x	x	x	x
1 1 0 1	x x x x	x	x	x	x	x	x	x	x
1 1 1 0	x x x x	x	x	x	x	x	x	x	x
1 1 1 1	x x x x	x	x	x	x	x	x	x	x

for J_A



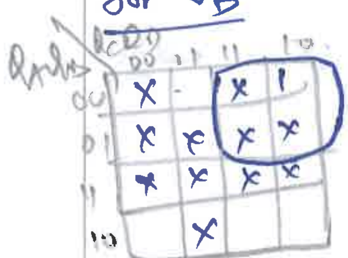
$J_A = Q_B Q_C$

for K_A



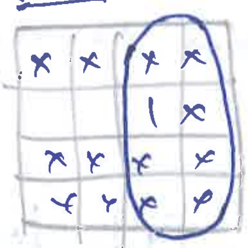
$K_A = Q_D$

for J_B



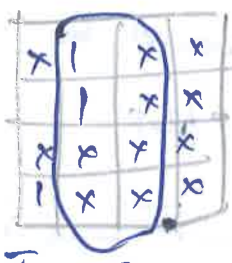
$J_B = \bar{Q}_A Q_C$

for K_B



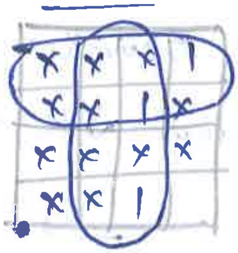
$K_B = Q_C$

for J_C



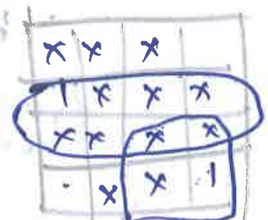
$J_C = Q_D$

for K_C



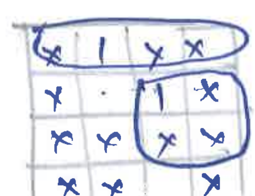
$K_C = \bar{Q}_A + Q_D$

for J_D



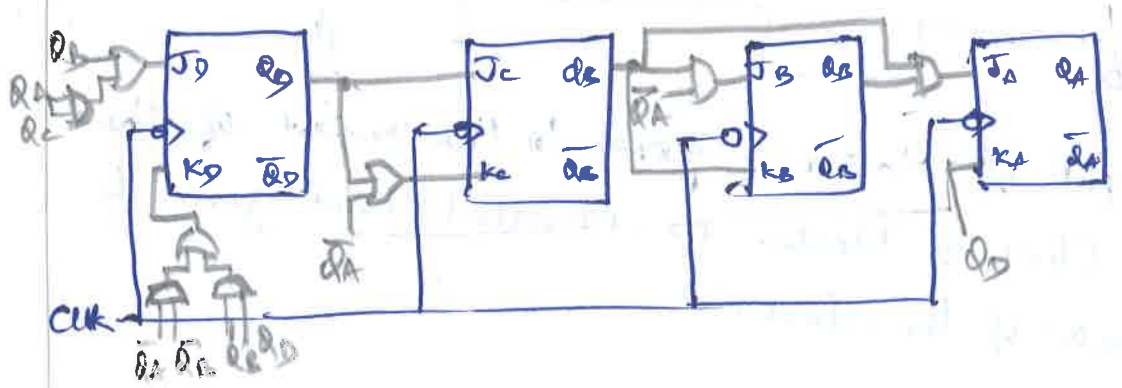
$J_D = Q_B + Q_A Q_C$

for K_D



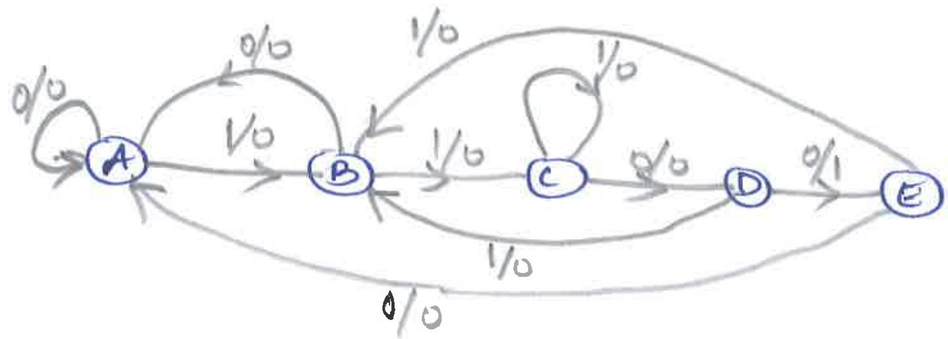
$K_D = \bar{Q}_A \bar{Q}_B + Q_B Q_D$

Logic Diagram



a) **Sequence Detector to detect 1100**

A → Initial State
B → 1
C → 11
D → 110
E → 1100



State Table

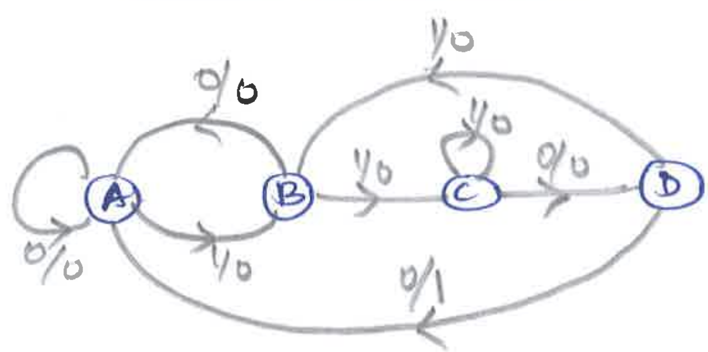
PS	NS, Z $x=0$	NS, Z $x=1$
A	A, 0	B, 0
B	A, 0	C, 0
C	D, 0	C, 0
D	E, 1	B, 0
E	A, 0	B, 0

States A and E are equivalent states.
Let us remove state E and replace it with state A.

∴ **Final State Table**

PS	NS, Z $x=0$	NS, Z $x=1$
A	A, 0	B, 0
B	A, 0	C, 0
C	D, 0	C, 0
D	A, 1	B, 0

Final State Diagram



10 b) State Reduction identifies and removes the redundant (or) equivalent states.

An Example:

Advantages:-

Reduces Hardware Complexity
Reduces Cost
Increases Speed
Improves Efficiency
Simplifies Design

Verified

Prepared by,
P V K Chaitanya
Assistant Professor
ECE, GVPCEW