Digital Electronics IA-3

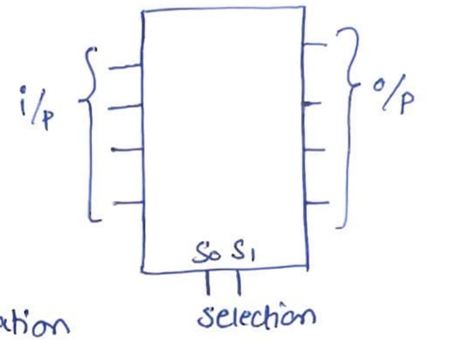
Yashvardhan Singh
Electronics engg. (VLSI)
Roll no. 46
230959136

Q1. # Universal Shift Register (USR)

(i) ↳ performs 4 different function:

1) Shift left
2) Shift Right
3) Parallel In Parallel out
4) Hold operation (Holds Data)

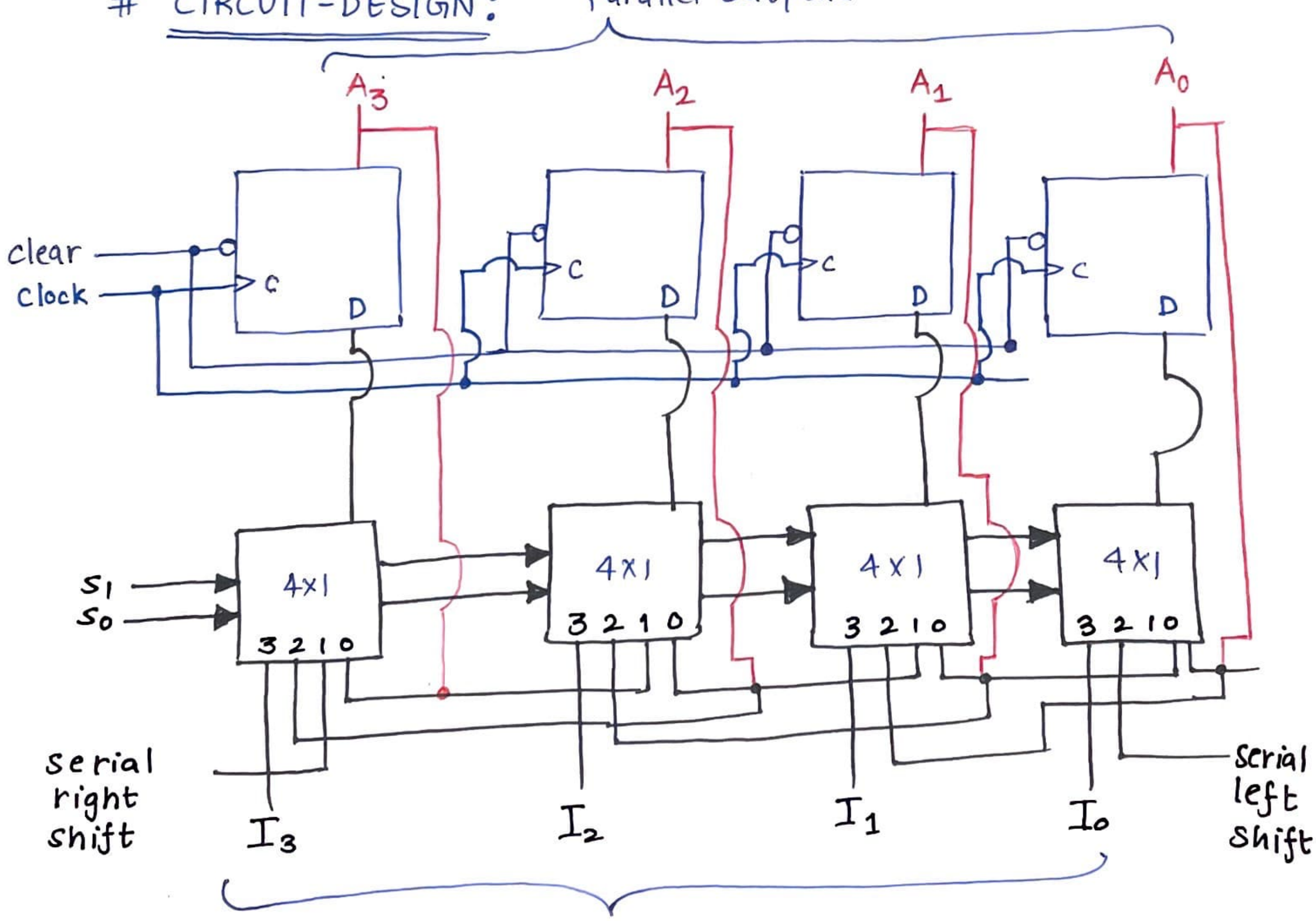
USR Block diagram



So	S1	operation
0	0	hold
0	1	right shift
1	0	left shift
1	1	PIPO

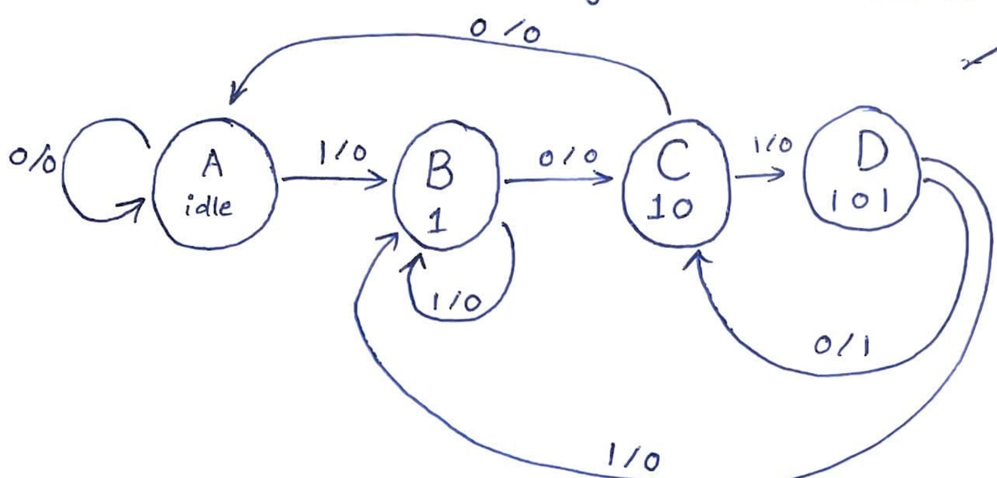
for operation selection, MUX can be used.

CIRCUIT-DESIGN:



Yashwardhan Singh / VLSI-46 / DC IA3 / 230959136 / yus..

Q1. (ii) 1010 overlapping sequence detector.



N = 4 states

\# Mealy ~~Moore~~ State Model

\# flip-flops needed:

$2^{P-1} < N \leq 2^P$

P = 2

state table

Present state	x = 0	x = 1
A	A/0	B/0
B	C/0	B/0
C	A/0	D/0
D	C/1	B/0
—	—	—

assign state vectors

A = 000 ⇒ 00

B = 001 ⇒ 01

C = 010 ⇒ 10

D = 011 ⇒ 11

excitation table :

\# JK excitation :

Q	Q⁺	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

	Q (A B)	X (i/p)	Q^+ (A B)	Z (o/p)	J_A	K_A	J_B	K_B	
A	0 0	0	0 0	0	0	X	0	X	0
A	0 0	1	0 1	0	0	X	1	X	1
B	0 1	0	1 0	0	1	X	X	1	2
B	0 1	1	0 1	0	0	X	X	0	3
C	1 0	0	0 0	0	X	1	0	X	4
C	1 0	1	1 1	0	X	0	1	X	5
D	1 1	0	1 0	1	X	0	X	1	6
D	1 1	1	0 1	0	X	1	X	0	7

Z:

A \ Bx	00	01	11	10
0	0	0	0	0
1	0	0	0	1

$Z = AB\overline{X}$

Solving K-maps :

J_A ⇒

A \ Bx	00	01	11	10
0	0	0	0	1
1	X	X	X	X

$J_A = B\overline{X}$

A \ Bx	00	01	11	10
0	X	X	X	X
1	1	0	1	0

⇐ $K_A = \overline{B}\overline{X} + BX$

$= B \odot X$

J_B ⇒

A \ Bx	00	01	11	10
0	0	1	X	X
1	0	1	X	X

$J_B = X$

A \ Bx	00	01	11	10
0	X	X	0	1
1	X	X	0	1

⇐ K_B

$K_B = \overline{X}$

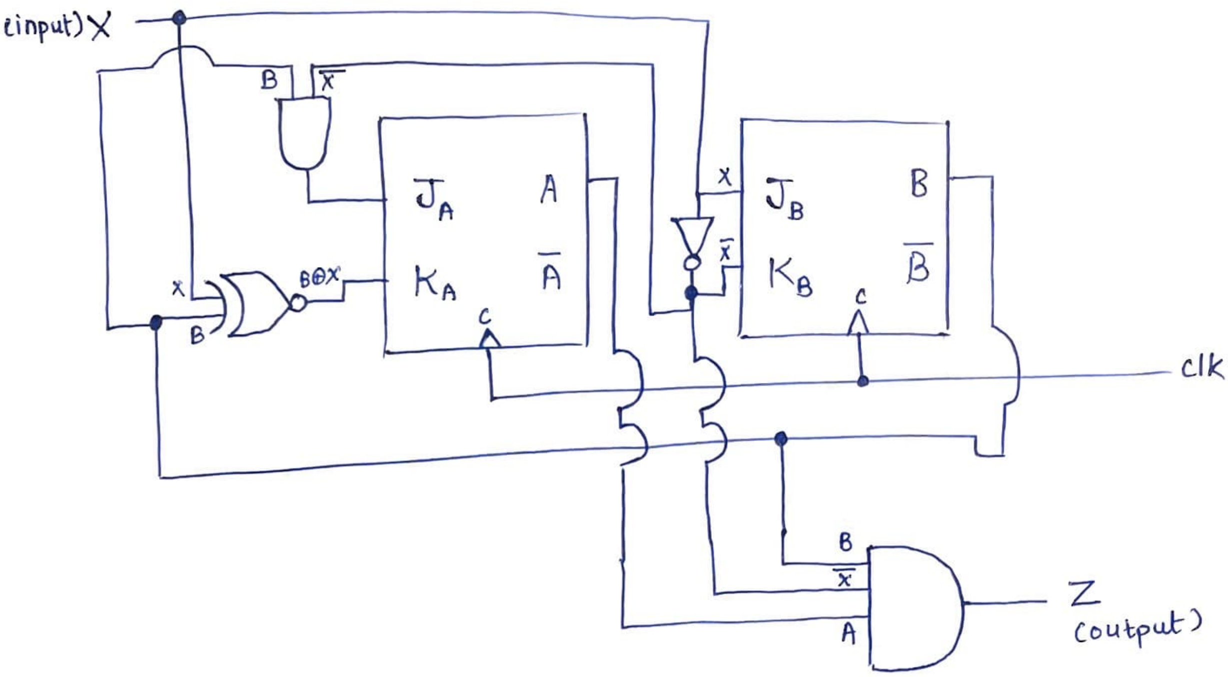
Yashwardhan Singh
230959136
VLSI - 46

Q1) (ii) # obtained equations :

- $J_A = B\overline{X}$
- $K_A = B \oplus X$
- $J_B = X$
- $K_B = \overline{X}$
- $Z = AB\overline{X}$

X : input
Z : output

Circuit design :



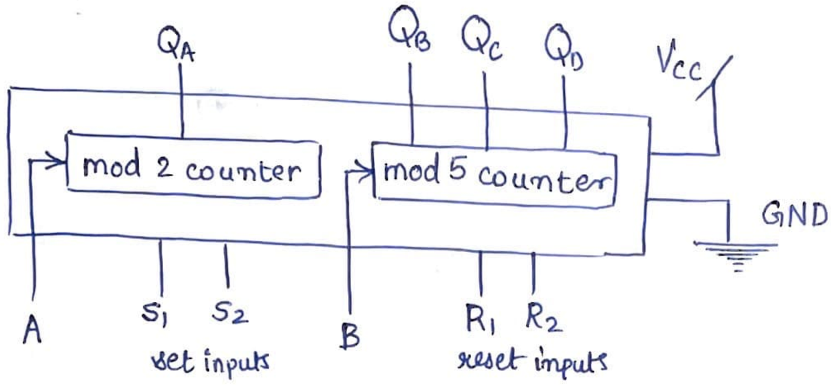
Q1) (iii) • Decade counter is a MOD-10 counter, which means this sequential circuit counts from 0 to 9 and then keeps repeating this process.
(0 ↓ 0000, 9 ↓ 1001)

- IC-7490 can be used for this implementation as it is a BCD Decade Counter.

count	Q_D	Q_C	Q_B	Q_A	
0	0	0	0	0	→ cycle starts
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	0	0	0	0	→ cycle repeats

Yashwardhan Singh
230959136
VLSI-46

Q1) (iii) IC 7490 Decade Counter:



Q1) (iv) 3 input odd parity generator.

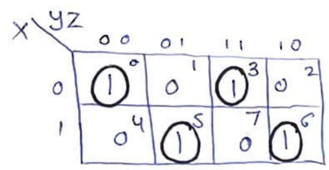
⇒ parity is a concept used for error detection in digital circuits. parity is of 2 kinds : even parity and odd parity.

- odd parity = odd number of 1's.
- even parity = even number of 1's.

- an odd parity generator will generate an extra 1 to make the data have odd number of ones, if it previously had even number of ones.

DATA INPUTS			PARITY (odd)	
x	y	z	P	
0	0	0	1	→ 0
0	0	1	0	→ 1
0	1	0	0	→ 2
0	1	1	1	→ 3
1	0	0	0	→ 4
1	0	1	1	→ 5
1	1	0	1	→ 6
1	1	1	0	→ 7

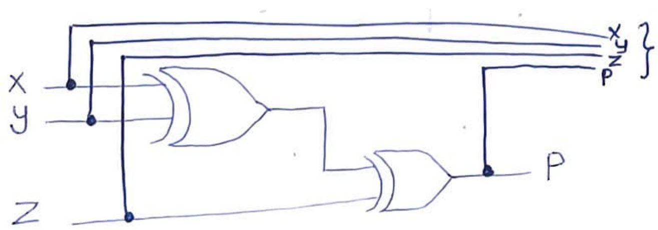
K-Maps :



checkboard config.

$P = \bar{x}\bar{y}\bar{z} + x\bar{y}z + \bar{x}yz + xy\bar{z}$

$= x \oplus (\overline{y \oplus z})$



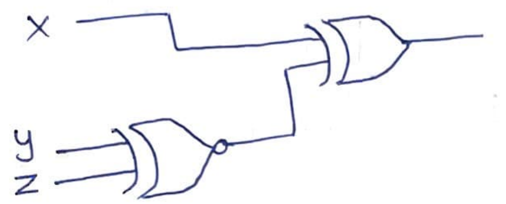
} 4-bit ~~even~~ odd parity generator

Yashwardhan Singh
VLSI - 46
230959136

Similarly, we can design a 3 bit odd parity generator.

Note: truth-table and K-map on prev. page

$$P = x \oplus (\overline{y \oplus z}) = x \oplus (y \odot z)$$



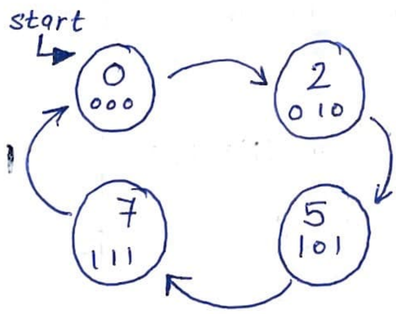
→ 3 bit odd parity generator

Q1] (v) Synchronous counter design

These counters have flip-flops that share all of the same clock signal.

given counter sequence: 0 → 2 → 5 → 7 → 0

- 0 → 0 0 0
- 2 → 0 1 0
- 5 → 1 0 1
- 7 → 1 1 1



state diagram

state & excitation :

	A_n	B_n	C_n	A_{n+1}	B_{n+1}	C_{n+1}	D_A	D_B	D_C
start → 0	0	0	0	0	1	0	0	1	0
1	0	1	0	1	0	1	1	0	1
2	1	0	1	1	1	1	1	1	1
3	1	1	1	0	0	0	0	0	0
cycle repeats → 4	0	0	0	0	1	0	0	1	0
	. . . .			. . . .					
5	0	1	0	1	0	1	1	0	1
6	1	0	1	1	1	1	1	1	1
7	1	1	1	0	0	0	0	0	0

Yashwardhan Singh
VLSI - 46
230959236

Q1] (v) #K-maps :

D_A

A \ BC	00	01	11	10
0	0	1	0	1
1	0	1	0	1

D_B

A \ BC	00	01	11	10
0	1	0	0	1
1	1	0	0	1

D_C

A \ BC	00	01	11	10
0	0	1	0	1
1	0	1	0	1

(1) $D_A = \overline{B}C + B\overline{C}$

$= B \oplus C$

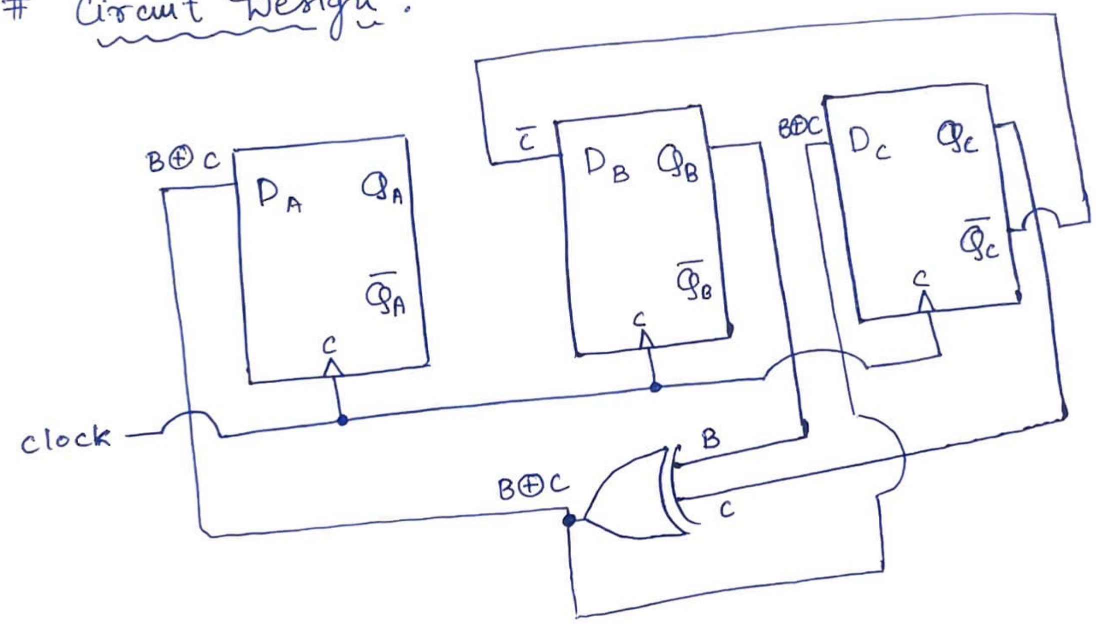
(2) $D_B = \overline{B}\,\overline{C} + B\overline{C}$

$= \overline{C}(B + \overline{B})$

$= \overline{C}$

(3) $D_C = B \oplus C$

Circuit Design :



Q2.] (i) $F = \Sigma m(0,1,2,3,5,7,8,9,10,12,13)$

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

$F = \overline{A}\overline{B} + \overline{A}D + A\overline{C} + A\overline{B}\overline{D}$

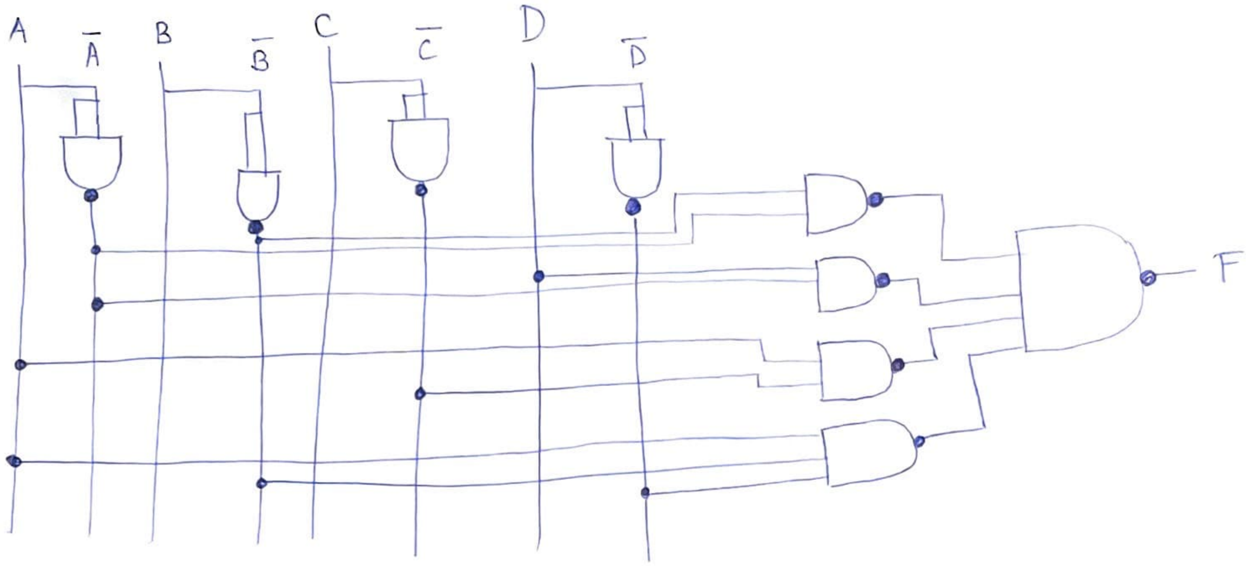
Yashwardhan Singh
230959136
VLSI - 46

Q2] (i) $F = \bar{A}\bar{B} + \bar{A}D + A\bar{C} + A\bar{B}\bar{D}$

now we implement this using universal logic:

$$\bar{\bar{F}} = \overline{\overline{\bar{A}\bar{B} + \bar{A}D + A\bar{C} + A\bar{B}\bar{D}}}$$

$$= \overline{\overline{\bar{A}\bar{B}} \cdot \overline{\bar{A}D} \cdot \overline{A\bar{C}} \cdot \overline{A\bar{B}\bar{D}}}.$$



(ii) $F = \Pi m(2, 8, 9, 10, 11, 12, 14)$

↳ maxterms

AB \ CD	00	01	11	10
00	1 [0]	1 [1]	1 [3]	0 [2]
01	1 [4]	1 [5]	1 [7]	1 [6]
11	0 [12]	1 [13]	1 [15]	0 [14]
10	0 [8]	0 [9]	0 [11]	0 [10]

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

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

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

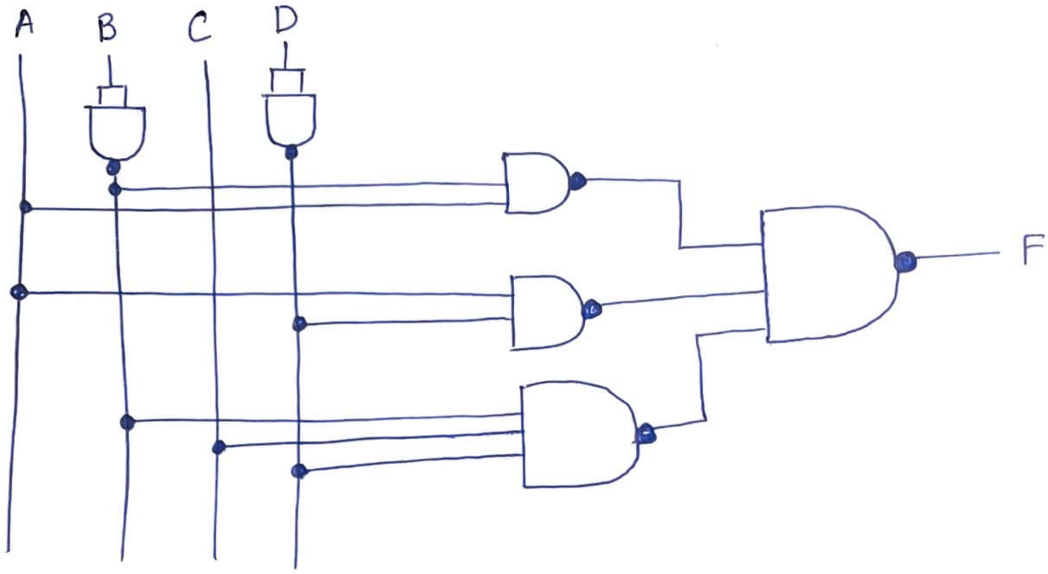
$$= A(\bar{B} + \bar{D}) + \bar{B}C\bar{D}$$

$$= A\bar{B} + A\bar{D} + \bar{B}C\bar{D}$$

Yashwardhan Singh
230959136
VLSI - 46

Q2) (ii) $\overline{\overline{F}} = \overline{\overline{A\bar{B} + A\bar{D} + \bar{B}C\bar{D}}}$

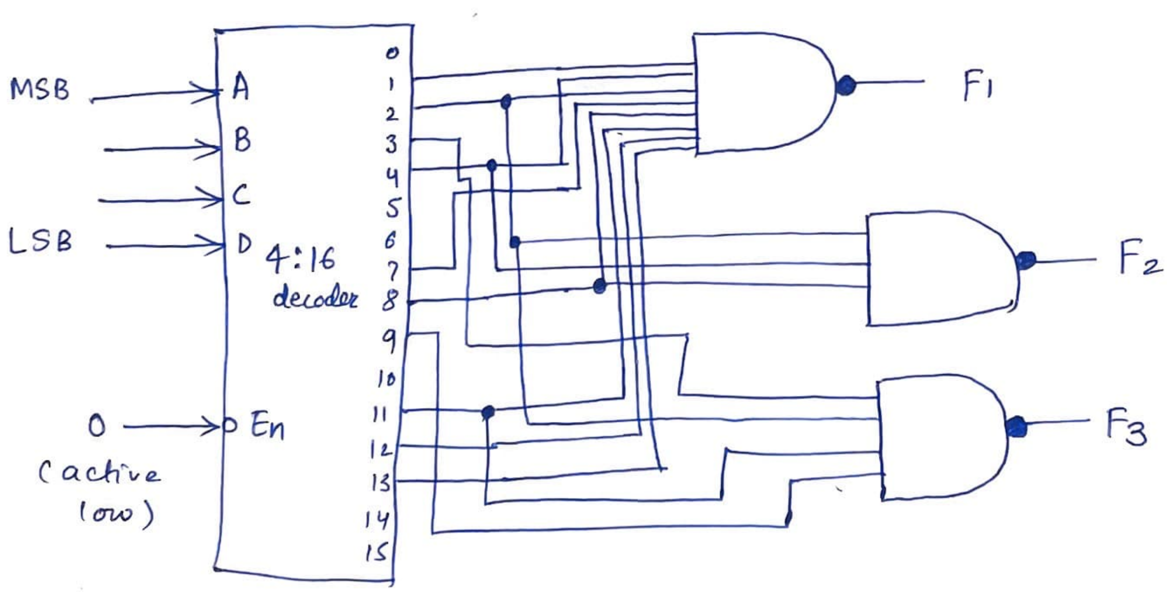
$= \overline{\overline{A\bar{B}} \cdot \overline{A\bar{D}} \cdot \overline{\bar{B}C\bar{D}}}$



Q3] a) $F_1 = \sum m(1,2,4,7,8,11,12,13)$

$F_2 = \sum m(2,4,8)$

$F_3 = \sum m(2,3,9,11)$



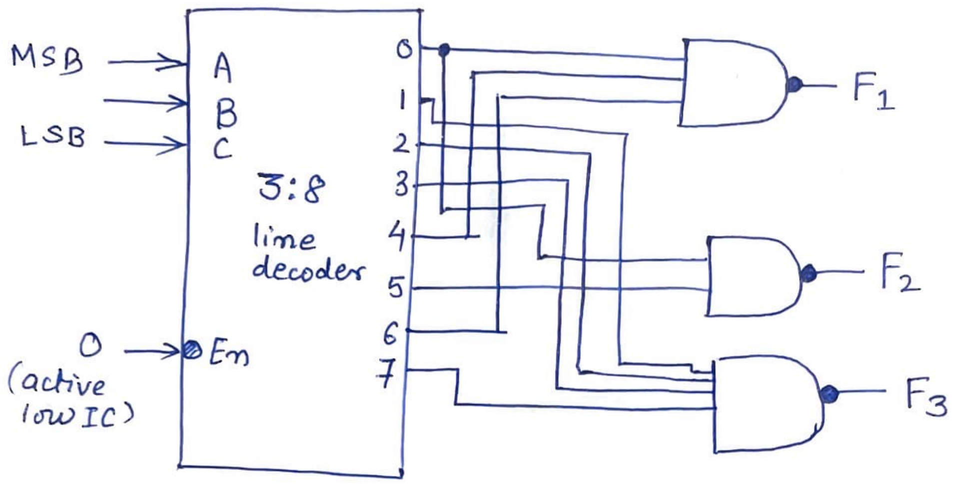
Yashwardhan Singh
230959136
VLSI-46

Q3]

(b) $F_1 = \Sigma m(0,4,6)$

$F_2 = \Sigma m(0,5)$

$F_3 = \Sigma m(1,2,3,7)$

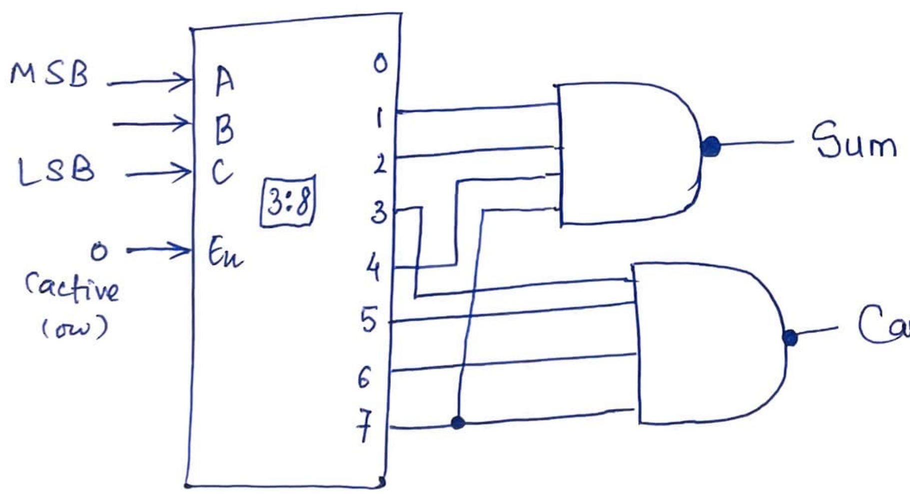


(c) Full adder using decoder and logic gates

we know :

$sum = \Sigma m(1,2,4,7)$

$carry = \Sigma m(3,5,6,7)$

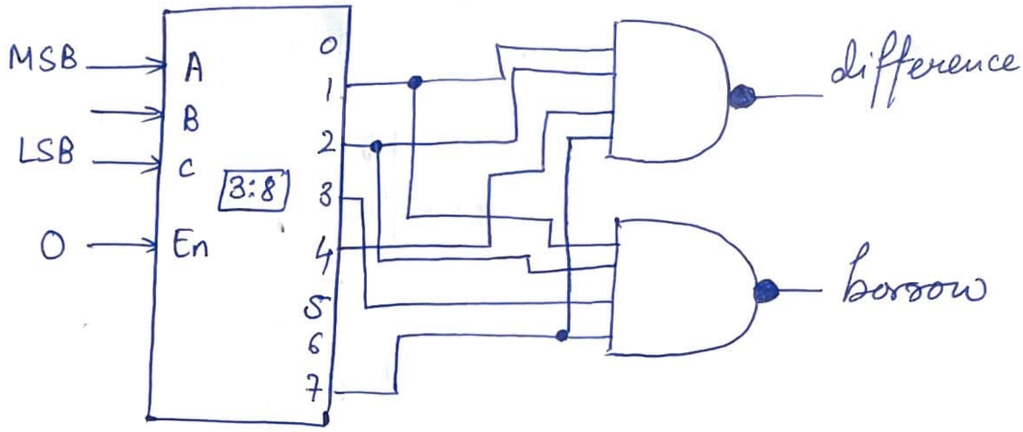


Yashwardhan Singh
230959136
VLSI-46

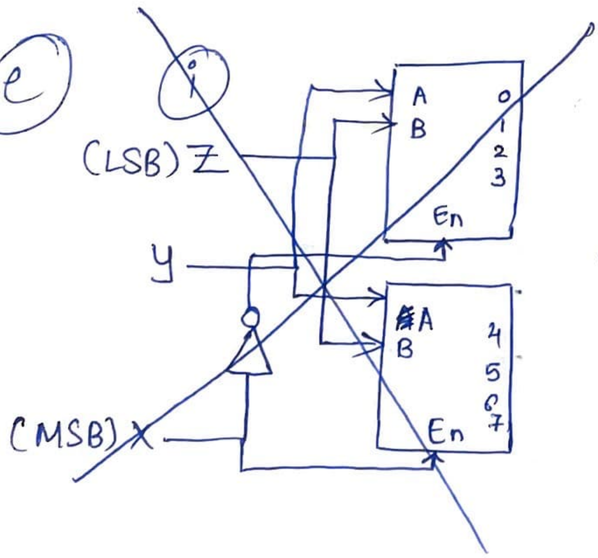
Q3) d) Full Subtractor using decoder

we know :

- difference : $\Sigma m(1, 2, 4, 7)$
- borrow : $\Sigma m(1, 2, 3, 7)$



Q3) e) i)

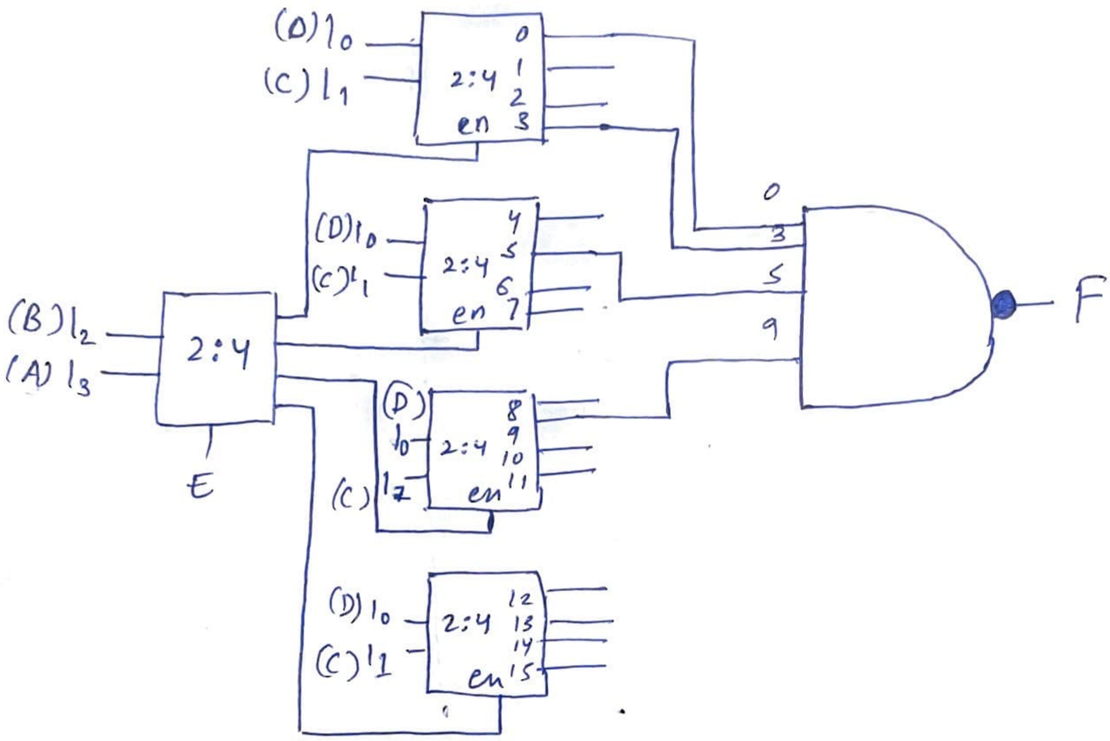


continued on next page

Yashwardhan Singh
VLSI-46
230959136

Q3) (e) $F(A,B,C,D) = \Sigma m(0,3,5,9)$

i)



(ii)