Book

9.10 for $s=0 \rightarrow F = A+B$.

" $S=1 \rightarrow F=A+1$

So,

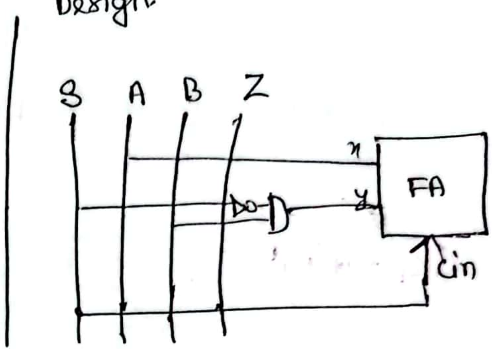
S	X	Y	Z_{in}	F
0	A	B	0	A + B
1	A	0	1	A+1

So, $X = AS + AS' = A(S+S') = A$.

$Y = S'B$.

$Z = S$.

Design



9.11 Said that $F = A-B$ produces correct difference if $A \geq B$

~~and~~ because $A - B \geq 2^n$. then Cout = 1

$\Rightarrow A + \bar{B} + 1 \geq 2^n$

$\Rightarrow A - B + \cancel{2^n} - \cancel{1} + \cancel{1} \geq \cancel{2^n}$

$\Rightarrow A \geq B$.

So when $A < B$ which means ~~A~~ the difference will ~~give~~ borrow 1 the result will be $A - B - 1 \geq 2^n$

$\Rightarrow A + \bar{B} \geq 2^n$

$\Rightarrow A - B + \cancel{2^n} - 1 \geq \cancel{2^n}$

$\Rightarrow A \geq B+1$ Which means there will be a borrow 1.

The relationship between F and a borrow in the most significant position is $A \geq B+1$.

9.12

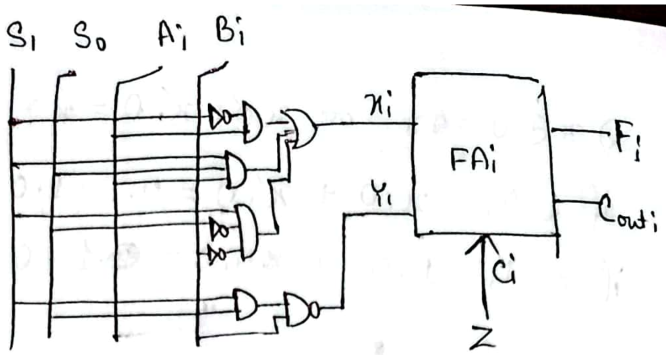
S_1	S_0	Cin	A X	B Y	Z	F
0	0	0	A	B	0	A+B
0	1	0	A	0	0	A
1	0	0	$\bar{B}$	~~B~~ 0	0	$\bar{B}$
1	1	0	A	$\bar{B}$	0	$A+\bar{B}$
0	0	1	A	B	1	A+B+1
0	1	1	A	0	1	A+1
1	0	1	$\bar{B}$	~~B~~ 0	1	$\bar{B}+1$
1	1	1	A	$\bar{B}$	1	$A+\bar{B}+1$

$Z = C_{in}$;

$x_i = S_1'A + S_1 S_0 A + S_1 S_0' B'$

$Y_i = S_1' S_0' B + S_1 S_0 \overline{B}$

$= S_1 S_0 \oplus B$



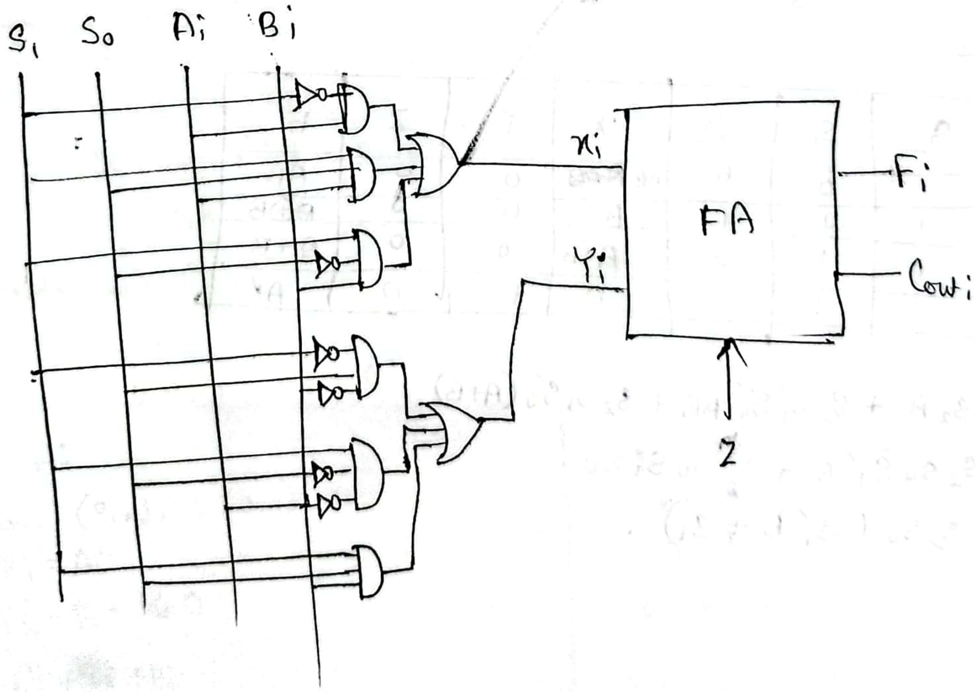
9.13

S_1	S_0	C_{in}	x	y	Z	F
0	0	0	A	0	0	A
0	1	0	A	$\overline{B}$	0	A − B − 1
1	0	0	B	$\overline{A}$	0	B − A − 1
1	1	0	A	B	0	A + B
0	0	1	A	0	1	A + 1
0	1	1	A	$\overline{B}$	1	A − B
1	0	1	B	$\overline{A}$	1	B − A
1	1	1	A	B	1	A + B + 1

$Z = C_i$

$x_i = S_1'A + S_1 S_0 A + S_1 S_0' B$

$Y_i = S_1' S_0 \overline{B} + S_1 S_0' \overline{A} + S_1 S_0 B$



9.14

a) $x \oplus 0 = x$, or, $x.0' + x'.0 = x$:

if $x=0$; $x.0' + x'.0 = 0.1 + 1.0 = 0$.

if $x=1$; $1.0' + 0.0 = 1.1 + 0 = 1$.

b) $x \oplus 1 = x'$ which means, $x.1' + x'.1 = x'$

if $x=0$; $0.1' + 0'.1 = 0.0 + 1.1 = 1$.

if $x=1$; $1.1' + 1'.1 = 0 + 0 = 0$.

c) $x \oplus y' = x(y')' + x'y'$.

$= xy + x'y'$.

$= x \odot y$. (Proved)

9.16

S_2	S_1	S_0	x	Y	Z	F
1	0	0	AB	0	0	AB
1	0	1	$A\oplus B$	0	0	$A\oplus B$
1	1	0	A+B	0	0	A+B
1	1	1	A'	0	0	A'

~~x =~~

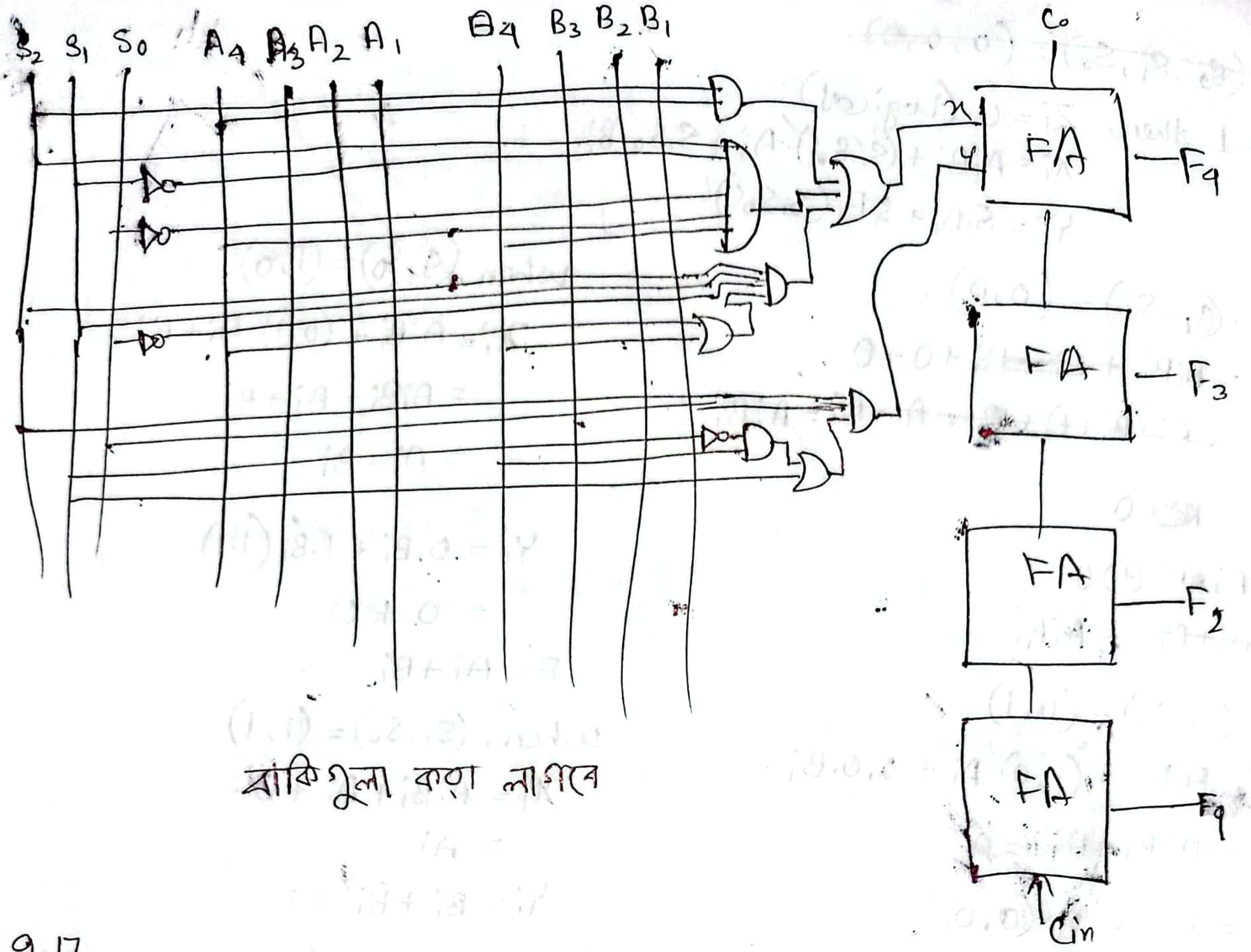
9.16

S_2	S_1	S_0	x	Y	Z	F
1	0	0	AB	0	0	AB
1	0	1	A	B	0	$A\oplus B$
1	1	0	A+B	0	0	A+B
1	1	1	A	1	0	A'

$x = S_2 A + S_2 S_1' S_0' AB + S_2 S_1 S_0' (A+B)$

$Y = S_2 S_0 S_1' B + S_2 S_0 S_1$

$= S_2 S_0 (S_1' B + S_1)$.



বাকিগুলা করা লাগবে

9.17

Given, $Z_i = S_2' C_i$

$x_i = A_i B_i + (S_2 S_1' S_0')' A_i + S_2 S_1 S_0' B_i$

$Y_i = S_0 B_i + S_1 B_i' (S_2 S_1 S_0')'$

if $S_2 = 0$, then $Z_i = C_i$ (arithmatic)

~~again is $(S_1, S_0) = (0,0)$~~

$x_i = A_i B_i + (0 . S_1' S_0')' A_i + 0 . S_1 S_0' B_i$

$= A_i B_i + A_i + 0 = A_i$

$Y_i = S_0 B_i + S_1 B_i' (0 . S_1 S_0')'$

$= S_0 B_i + S_1 B_i'$

when, $(S_1, S_0) = (0,0)$

$x_i = A_i$

$Y_i = 0 . B_i + 0 . B_i' = 0$

$F = A_i$

when, $(S_1, S_0) = (0,1)$

$x_i = A_i$

$Y_i = B_i + 0$

$F = A_i + B_i$

when $(S_1, S_0) = (1,0)$

$x_i = A_i$

$Y_i = 0 + B_i'$

$F = A_i + \overline{B_i}$

when $(S_1, S_0) = (1,1)$

$x_i = A_i$

$Y_i = B_i + B_i' = 1$

$F = A_i + 1$

~~if $(S_2, S_1, S_0) = (0,0,0)$~~

if $S_2 = 1$ then, $Z_i = 0$. (Logical)

$x_i = A_iB_i + (S_1'S_0')' A_i + S_1S_0' B_i$

$Y_i = S_0B_i + S_1B_i(S_1S_0')'$

when, $(S_1, S_0) = (0,0)$.

$x_i = A_iB_i \cancel{+A_i+B_i} + 0 + 0$.

$\cancel{= A_i(B_i+1)+0 = A_i+B_i} = A_iB_i$

$Y_i = 0$.

$F = A_i B_i \oplus 0$.

$\cancel{= A_i+B_i} = A_iB_i$

when, $(S_1, S_0) = (0,1)$

$x_i = A_iB_i + (1.0)' A_i + 0.0.B_i$.

$= A_iB_i + A_i = A_i$

$Y_i = B_i + 0.B_i'(0.0)'$

$= B_i$

$F = A_i \oplus B_i$

when, $(S_1, S_0) = (1,0)$

$x_i = A_iB_i + (0.1)' A_i + B_i$

$= A_iB_i + A_i + B_i$

$= A_i + B_i$

$Y_i = 0.B_i + 1.B_i(1.1)'$

$= 0 + 0$

$F = A_i + B_i$

when, $(S_1, S_0) = (1,1)$

$x_i = A_iB_i + A_i + 0$.

$= A_i$

$Y_i = B_i + B_i' = 1$

$F = A_i \oplus 1 = A_i'$

S_2	S_1	S_0	x	Y	Cin	F
0	0	0	A_i	0	0	A_i
0	0	1	A_i	B_i	0	A_i+B_i
0	1	0	A_i	$\overline{B_i}$	0	$A_i+\overline{B_i}$
0	1	1	A_i	0	1	A_i+1
1	0	0	A_iB_i	0	0	A_iB_i
1	0	1	A_i	B_i	0	$A_i \oplus B_i$
1	1	0	A_i+B_i	0	0	A_i+B_i
1	1	1	A_i	1	0	A_i'

Table করা লাগবে না
বললে করব.

Q.18

a) $F = A + \overline{B}$

if $A = B$. that means $\overline{B} = \overline{A}$

so, $A + \overline{A} = 1$.

So, $F = 1$.

b). $C_{out} = 1$ if $A + \bar{B} \geq 2^n$.

$\Rightarrow A - B - 1 \geq 2^n \quad [\because \bar{B} = -B-1]$

$\Rightarrow A - B + \cancel{2^n} - 1 \geq \cancel{2^n} \quad [\because -1 = \text{all } 1 = 2^n - 1]$

$\Rightarrow A \geq B + 1$.

Sub: All the $C_{out}=1$ if $F \geq 2^n$ equations:

① $F = A$.

$= A + 0 + 0$.

So, $C_{out} = 0$ always.

② $F = A + 1$.

$C_{out} = 1$ if $F \geq 2^n$

or, $A + 1 \geq 2^n$

or, $A \geq 2^n - 1$.

③ $F = A + B$.

$C_{out} = 1$ if $A + B \geq 2^n$.

④ $F = A + B + 1$

$C_{out} = 1$ if $A + B + 1 \geq 2^n$

$\Rightarrow A + B \geq 2^n - 1$.

⑤ $F = A - B - 1$

$C_{out} = 1$ if $A - B - 1 \geq 2^n$

Here, $-B - 1 = \overline{B}$

$\Rightarrow -B + 2^n - 1 = \overline{B}$

So, $A + \overline{B} \geq 2^n$.

or, $A - B + \cancel{2^n} - 1 \geq \cancel{2^n}$.

or, $A \geq B + 1$.

⑥ $F = A - B$

$C_{out} = 1$ if $A - B \geq 2^n$

$-B = \overline{B} + 1$

~~$\Rightarrow -B = -B - 1 + 1$~~

So, $A + \overline{B} + 1 \geq 2^n$

$\Rightarrow A - B - 1 + 1 \geq 2^n$

$\Rightarrow A - B + \cancel{2^n} - \cancel{1} + \cancel{1} \geq \cancel{2^n}$

$\Rightarrow A \geq B$.

⑦ $F = A - 1$

$C_{out} = 1$ if $A - 1 \geq 2^n$

$\Rightarrow A + \cancel{2^n} - 1 \geq \cancel{2^n}$

$\Rightarrow A \geq 1$.

⑧ $F = A$ with $C_{out} = 1$.

Here, $F = A + 1 - 1$

$= A + \cancel{1} + 2^n - \cancel{1}$

$= A + 2^n$.

So, always $C_{out} = 1$.

Next

(S

Quiz-1 (Set-A)

① Given F = A − B − 1 We know, $F \geq 2^n$ then $C_{out} = 1$

a) ~~if Cout = 1~~ then, $A+\overline{B} \geq 2^n$. $\left[\because \overline{B} = -B-1\right]$

We know,

$$\overline{B} = -B-1$$

$$\Rightarrow \overline{B} = -B + (2^n - 1) \quad [\because \text{all} 1 = -1 = (2^n-1)]$$

So, $A + \overline{B} \geq 2^n$

$$\Rightarrow A + \cancel{2^n} - 1 - B \geq \cancel{2^n}$$

$$\Rightarrow A - B - 1 \geq 0 .$$

$$\Rightarrow A - B \geq 1 .$$

$$\Rightarrow A \geq 1 + B .$$

b) F = A − B.

~~if Cout = 1~~ then, $A + \overline{B} + 1 \geq 2^n$ $\left[\because -B = \overline{B}+1\right]$

$$\Rightarrow A - B + (\cancel{2^n} - \cancel{1}) + \cancel{1} \geq \cancel{2^n}$$

$$\Rightarrow A - B \geq 0 .$$

$$\Rightarrow A \geq B .$$

c) F = A − 1

So, $A - 1 \geq 2^n$.

$$\Rightarrow A + \cancel{2^n} - 1 \geq \cancel{2^n} \quad [\because -1 = \text{all} 1 \text{ and, all} 1 = 2^n - 1]$$

$$\Rightarrow A \geq 1$$

d) $F = A$; $C_{out} = 1$

So, $F = A - 1 + 1 = A + \text{all } 1 + 1$

So, $A + \text{all } 1 + 1 \geq 2^n$

$\Rightarrow A + \cancel{2^n} \cancel{-1} \cancel{+1} \geq \cancel{2^n}$

$\Rightarrow$.

d) $F = A$; $C_{out} = 1$

So, $F = A - 1 + 1$

$= A + \text{all}1 + 1$

$= A + 2^n \cancel{-1} \cancel{+1}$

$= A + 2^n$.

always, $C_{out} = 1$.

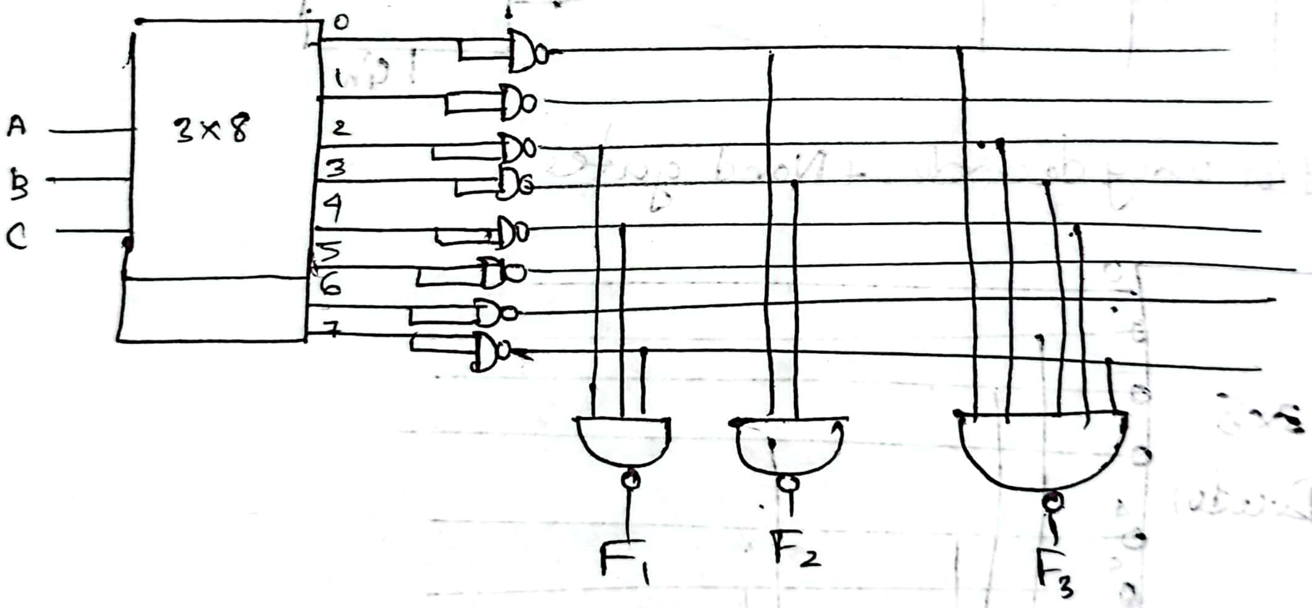
(2) $F_1(A,B,C) = \sum(2,4,7)$

$$= (A'BC' + AB'C' + ABC)''$$

$$= \{(A'BC')'(AB'C')'(ABC)'\}'$$

$F_2(A,B,C) = \sum(0,3) = \{(A'B'C')'(A'BC)'\}'$

$F_3(A,B,C) = \sum(0,2,3,4,7) = \{(A'B'C')'(A'BC')'(A'BC)'(AB'C')'(ABC)'\}'$



Set - B

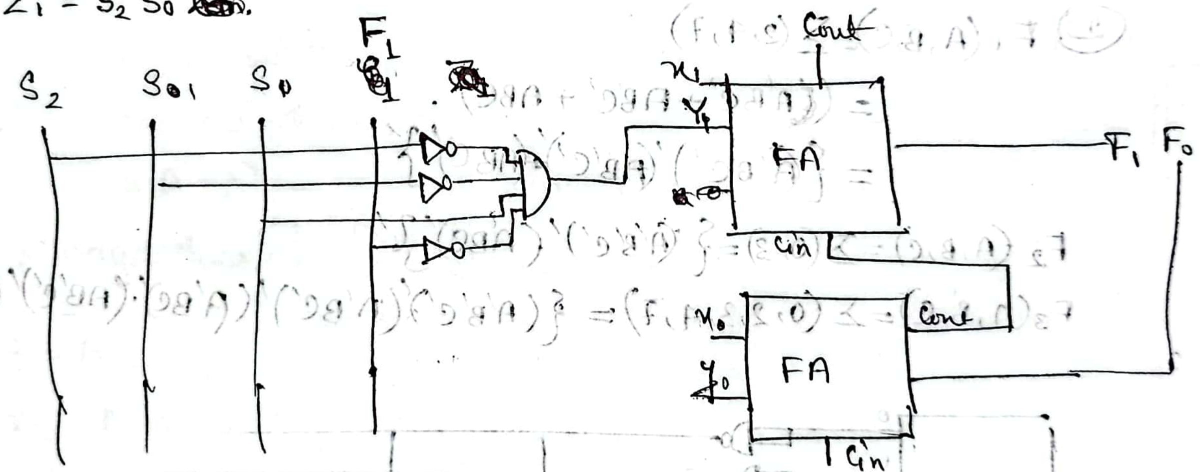
(1)

S_2	S_1	S_0	F^n	x_i	Y_i	Z_i
0	0	0	$F^n = E, C_{out} = 1$	E	0	1
0	0	1	$E - F - 1 = E + \bar{F}$	E	$\bar{F}$	0
0	1	0	$E + 1$	E	0	1
1	0	0	$\overline{E+F}$	$\overline{E+F}$	0	0
1	1	0	$E \oplus F$	$E \oplus F$	0	0

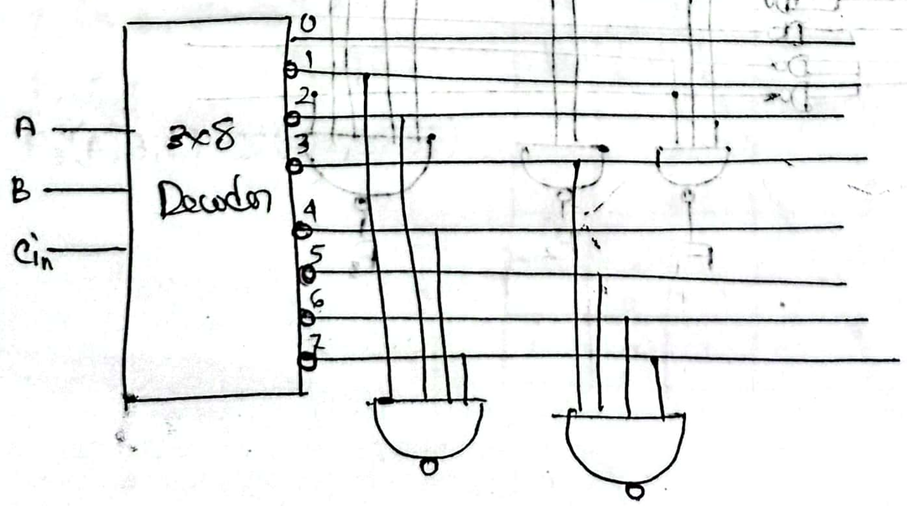
$$x_i = S_2'E + S_2 S_1' S_0' (\overline{E+F}) + S_2 S_1 S_0' (E \oplus F)$$

$$Y_i = S_2' S_1' S_0 \cdot \bar{F}$$

$$Z_i = S_2' S_0'$$



(2) Full adder using decoder + Nand gates



~~Quiz-1 (A)~~ Set C.

① $x_i = A_i + (S_2' S_1' S_0')' A_i B_i + S_2 S_1' S_0' B_i$

$y_i = S_2 S_1 S_0 B_i + S_0 B_i' (S_2 . S_1 S_0')'$

$z_i = S_2' C_{in}$.

i) $(S_2, S_1, S_0, C_{in}) = (0,0,0,0)$ [Given]

$z_i = C_{in} = 0$.

$x_i = A_i + (1.1.1)' A_i B_i + 0.1.0.B_i$

$= A_i$.

$y_i = 0 + 0.B_i'(0.0.0')'$

$= 0$.

$F = A_i$

ii) $(S_2, S_1, S_0, C_{in}) = (1,0,0,0)$

$z_i = 0$ ~~[logical]~~

$x_i = A_i + (0.1.1)' A_i B_i + 1.1.1.B_i$

$= A_i + A_i B_i + B_i$

$= A_i (B_i + 1) + B_i$

$= A_i + B_i$ [OR - logical]

$y_i = 1.0.0.B_i + 0.B_i'(1.0.1)'$

$= 0$

$F = A_i + B_i$

$= A_i \vee B_i$

iii) $(S_2, S_1, S_0, C_{in}) = (0,0,1,1)$

$z_i = C_{in}$. [Arithmatic].

$x_i = A_i + (1,1,0)' A_i B_i + (0.1.0) B_i$

$= A_i + A_i B_i + 0$.

$= A_i (1 + B_i) = A_i$

$y_i = 0.0.1.B_i + 1.B_i'(0.0.0)'$

$= 0 + B_i' = B_i'$

$F = A + \overline{B_i}$

iv) $(S_2, S_1, S_0, C_{in}) = (1,1,1,1)$

$z_i = 0$ [Arithmatic, since $C_{in} = 1$]

$x_i = A_i + (0)' A_i B_i + (1.0.0) B_i$

$= A_i + A_i B_i + 0$

$= A_i$

$y_i = 1.1.1.B_i + 1.B_i'(1.1.0)'$

$= B_i + B_i' = 1$

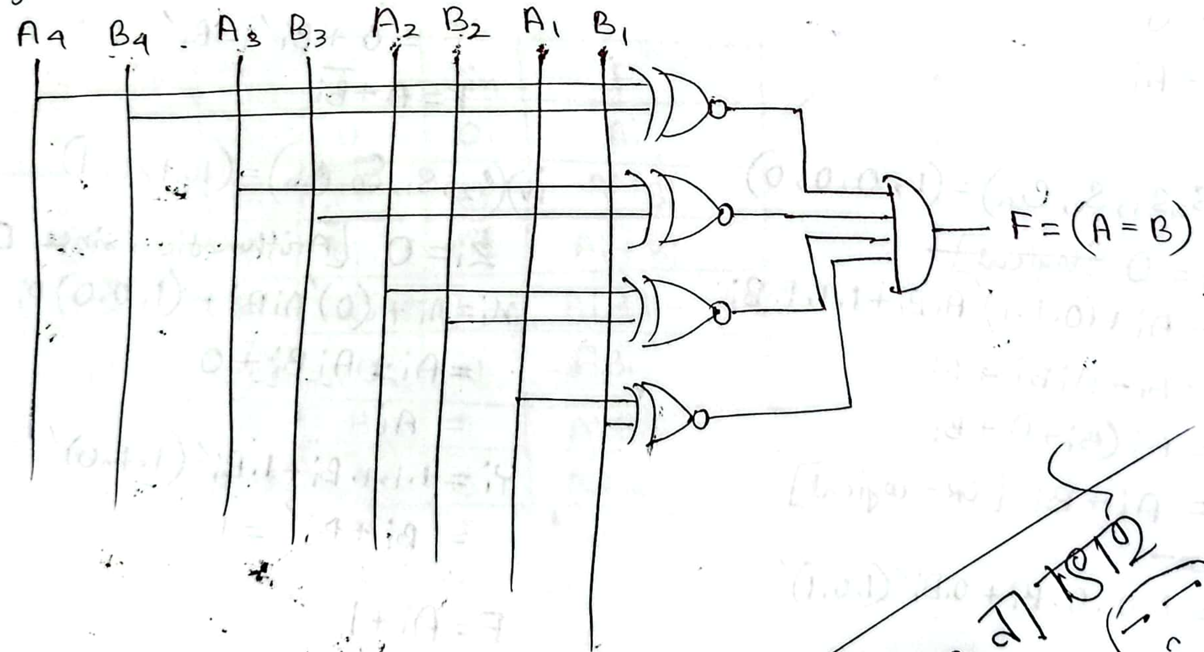
$F = A_i + 1$

② 4 bit comparator.

$A_4 B_4$	$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$F \rightarrow (A=B)$
$A_4 > B_4$				0
$A_4 < B_4$	A			0
$A_4 = B_4$	$A_3 > B_3$			0
$A_4 = B_4$	$A_3 < B_3$	B		0
$A_4 = B_4$	=	$A_2 > B_2$		0
$A_4 = B_4$	=	$A < B_2$		0
$A_4 = B_4$	=	=	$A_1 > B_1$	0
$A_4 = B_4$	=	=	$A_1 < B_1$	0
$A_4 = B_4$	=	=	=	1
~~$A_4 \neq B_4$~~	~~=~~	~~≠~~	~~≠~~	
~~$A_4 = B_4$~~	~~=~~		~~=~~	~~1~~

So, $F = (A_4 \odot B_4)(A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)$.

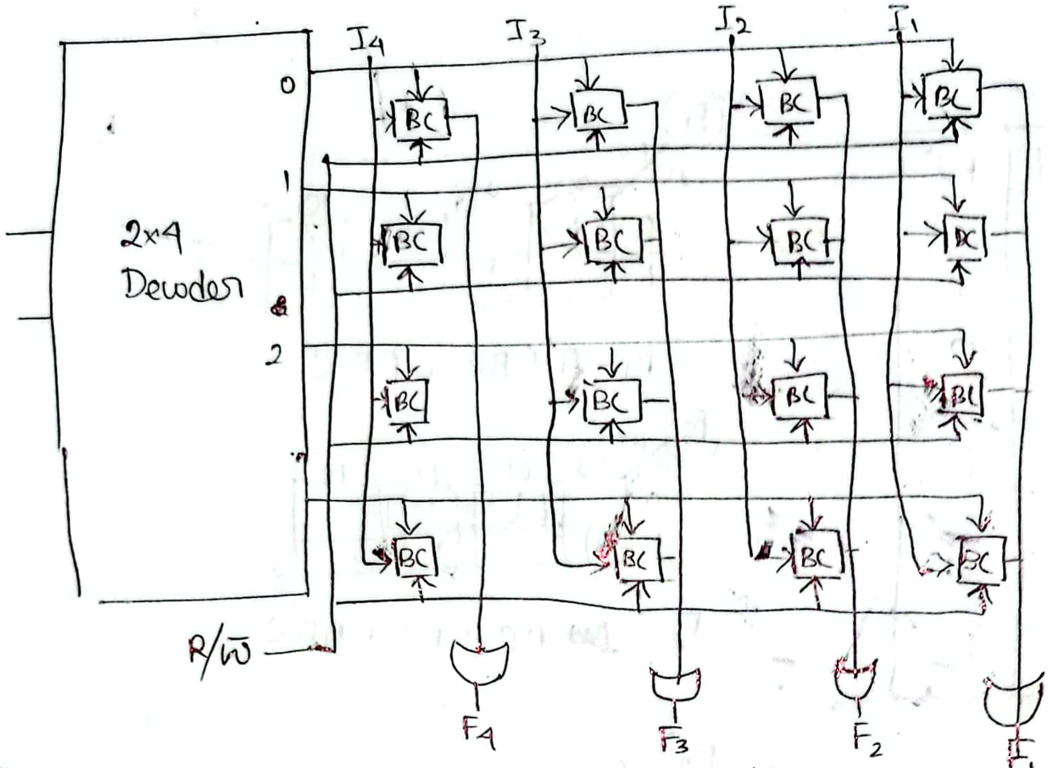
Design.



eta Sure না ভাই take Risk

Assignment 1

(0) 4x4 RAM:



No, '011' cannot be read/write from this RAM as it has 3 words and the RAM drawn is 4x4 RAM. It will read and write data with 4 bits.

(1) The printer that is ready and available will ~~start~~ printing first ~~for~~ first two parallel transfer methods.
In case of Simple I/O, each printer have a dedicated line where the computer places data and any printer that is ready receives it first

In case of Simple Strobe I/O, computer sends a strobe with data and the first printer to receive it starts first.

Again for Single-Handshake I/O, computer sends ~~data~~ request to ~~a printer~~ each printer in sequence. The printer that responds with an acknowledgement first gets to receive the data. The prioritize device is the one who handshakes first.

Lastly for double Handshake I/O, the printer who responds first gets the data and start printing first. It also ensures that the printer received the command precisely and is on it's way to output

Thus first two methods do not have any prioritization whereas the last two do have and which printer gets priority depends on it's responsiveness.

(2) $F1(A,B,C) = \Pi(1,3,4,7) = \Sigma(0,2,5,6)$

$F_2(A,B,C) = \Pi(0,2,5,6) = \Sigma(1,3,4,7)$.

TT:

A	B	C	F1	F2
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

F_1

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

$F_1 = A' + BC' + AB'C$

F_2

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

$F_2 = A'C + BC + AB'C'$

F_1'

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

$F_1' = A'C + BC + AB'C$

F_2'

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

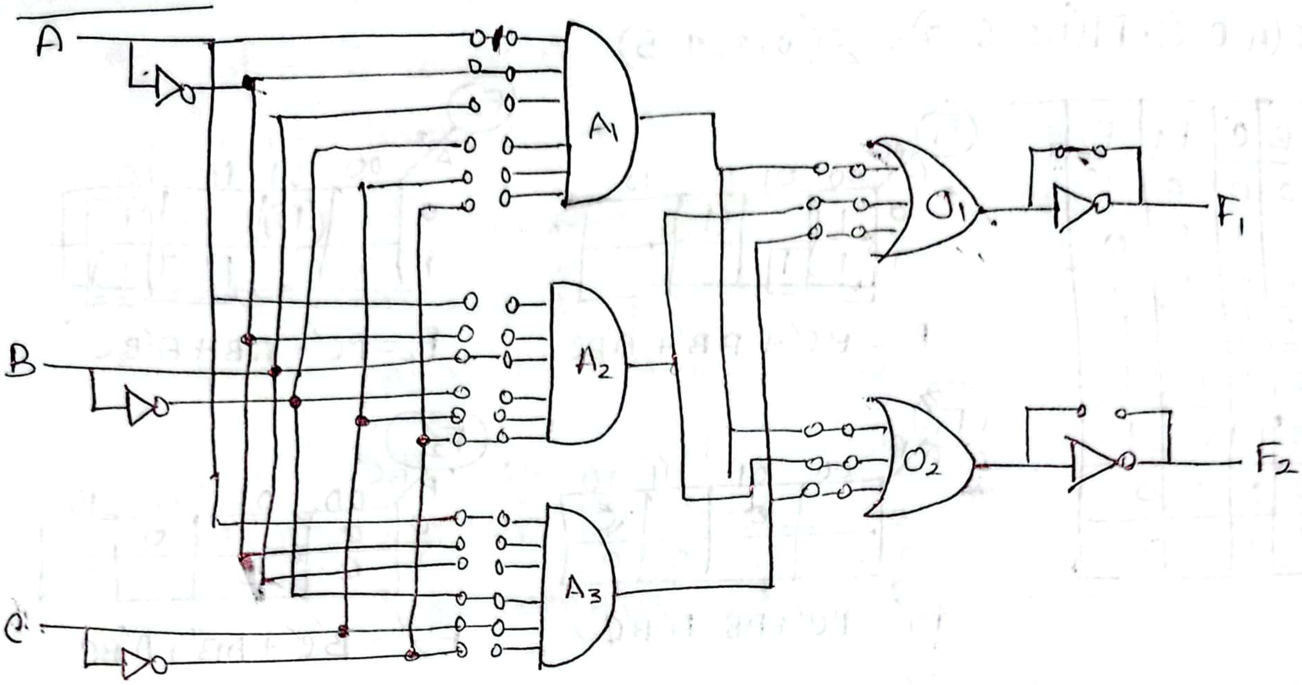
$F_2' = A' + BC' + AB'C$

Let's take F_1 and F_2'

Program table:

		A	B	C	F_1	F_2
A'	1	0	–	–	1	1
BC'	2	–	1	0	1	1
$AB'C$	3	1	0	1	1	1
					T	C

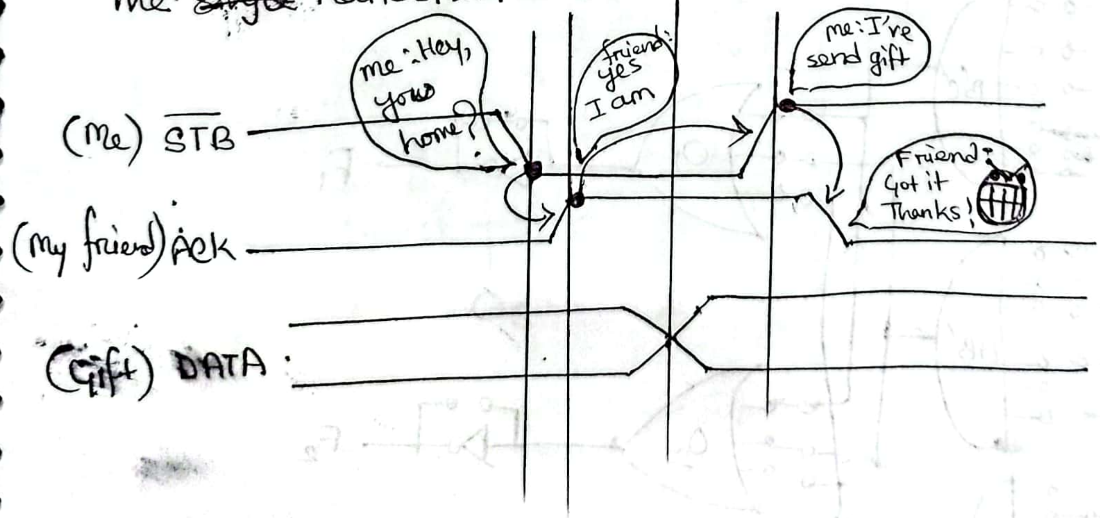
Circuit



~~(3) According to single handshake I/O, I would send the gift as I ha~~

(3) According to ~~single or~~ double handshake I/O of parallel transfer methods calling my friend ~~was~~ is similar to sending a strobe to ensure that my friend is ready to receive. As he did not respond I would not send the gifts.. I would have send if he was available.

The double ~~single~~ handshak I/O method shows:



(4) $F1(A,B,C) = \prod(0,3,4,5) = \sum(1,2,6,7)$
$F2(A,B,C) = \prod(1,2,6,7) = \sum(0,3,4,5)$

TT:

A	B	C	F_1	F_2
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	0

F_1

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

$F_1 = B'C' + AB' + A'BC$

F_2

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

$F_2 = BC' + AB + A'B'C$

F_1'

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

$F_1' = BC' + AB + A'B'C$

F_2'

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

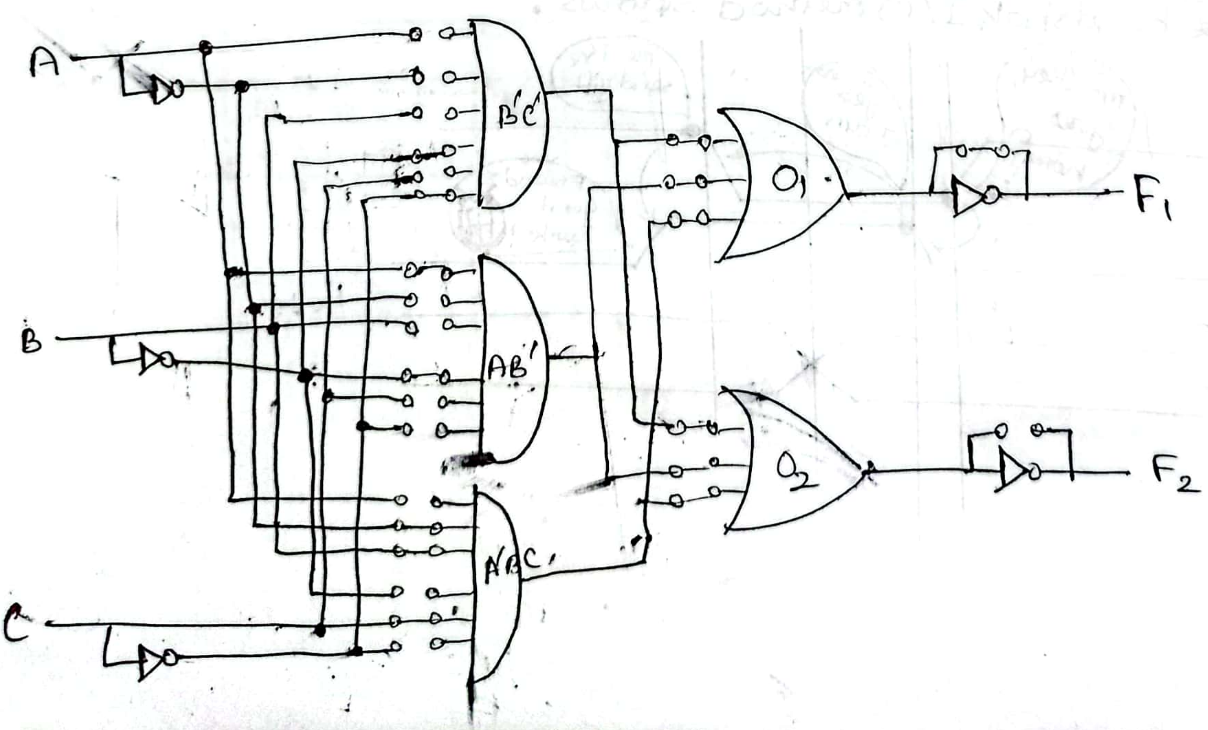
$F_2' = B'C' + AB' + A'BC$.

Let's take, F_1 and F_2'

Program table:

	Program Terms	Inputs A	B	C	Outputs F_1	F_2
$B'C'$	1	–	0	0	1	1
AB'	2	1	0	–	1	1
$A'BC$	3	0	1	1	1	1
					T	C

Circuit:



(5) Double handshake I/O is the best :

First of all it sends strobe and seeks acknowledge of the receiver if it's ready or available to receive or not.

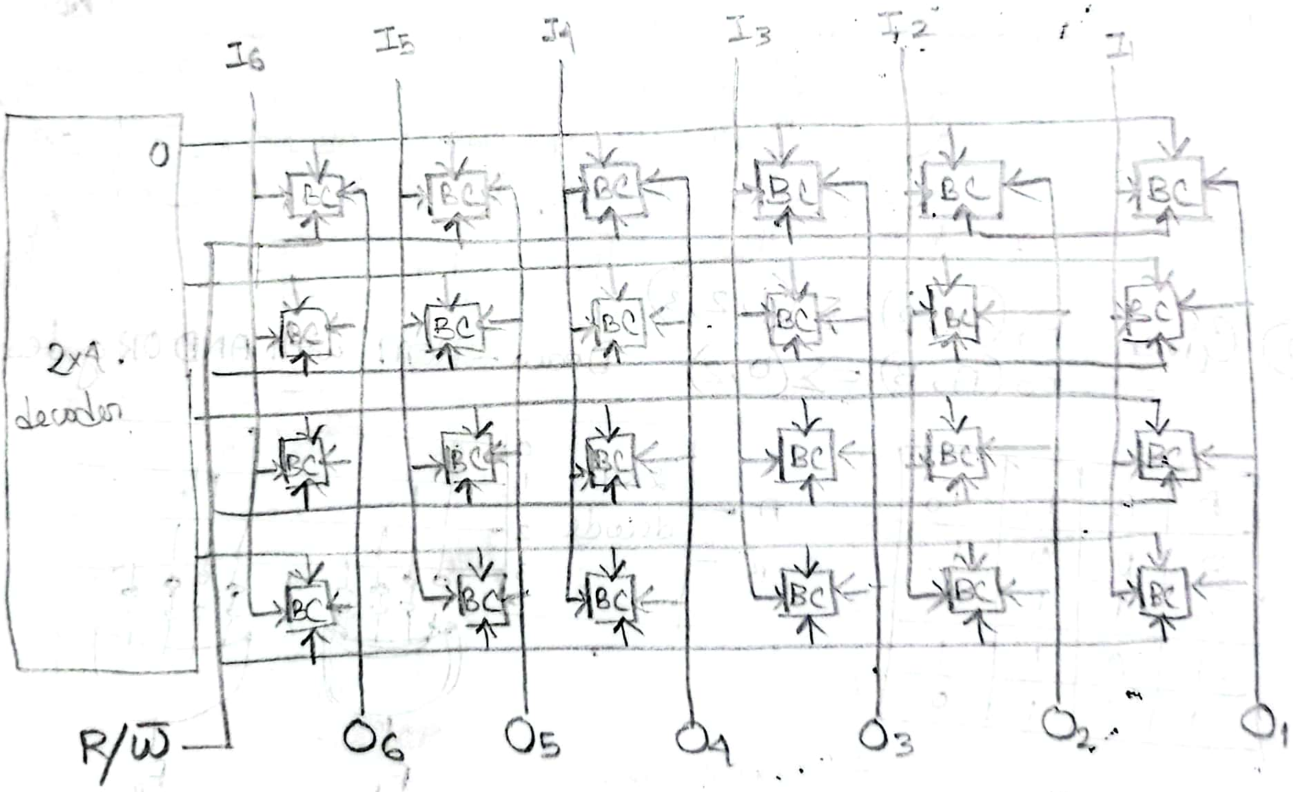
Then again after sending data, sender sends another strobe ~~claim~~ informing they sent data.

The receiver sends the 2nd ack letting sender know that the data is successfully received.

The communication btw sender receiver is good here.

Thus it is best among all. [Fig: 3 মার্কের mark বেশি থাকলে]

(6) 4x6 RAM



lecture–Practice : ~~2048 × 8 ROM~~ 2048 bit ROM – word size = 8 words.

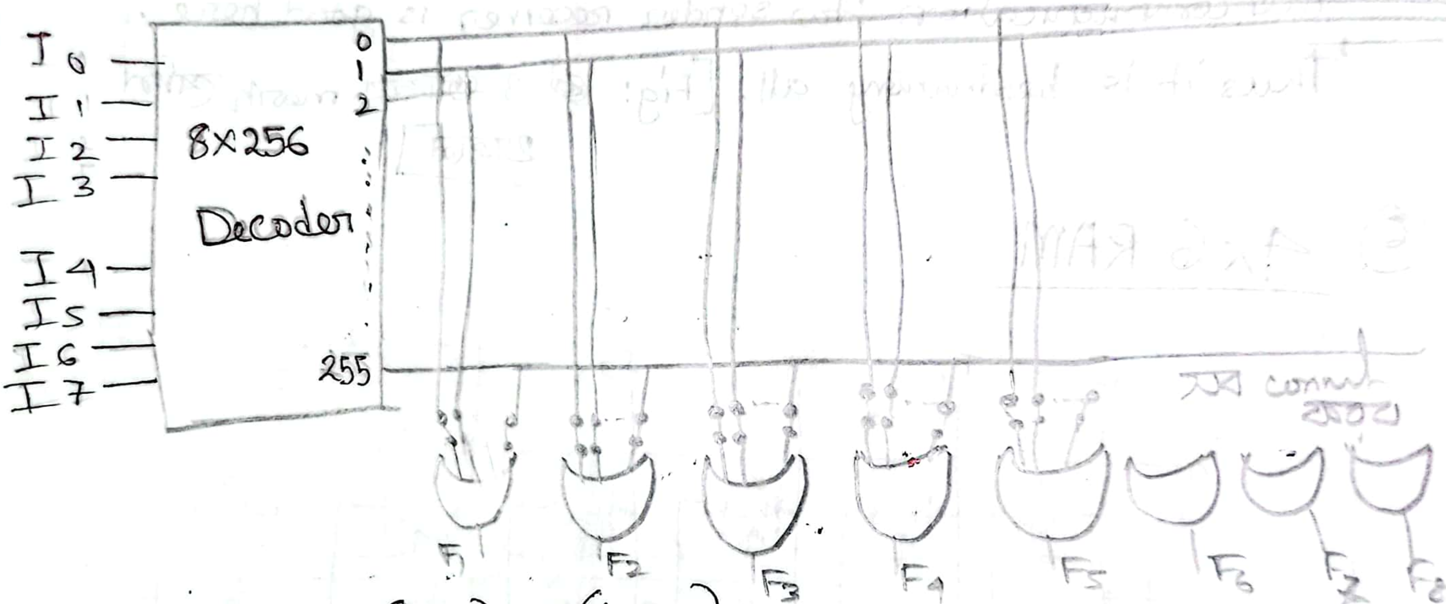
(*) $2048/8 = 256$:

So, 256×8 ROM. or, $2^8 \times 8$ ROM.

Here, $n = 8$, $m = 8$. $[\because 2^n \times m \text{ ROM}]$

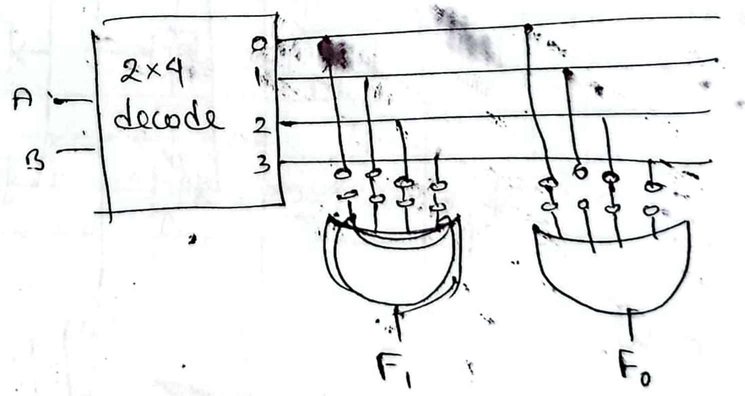
range $= (0000\ 0000 - 1111\ 1111)_2$ or $(0-255)_{10}$

Design: Logic diagram

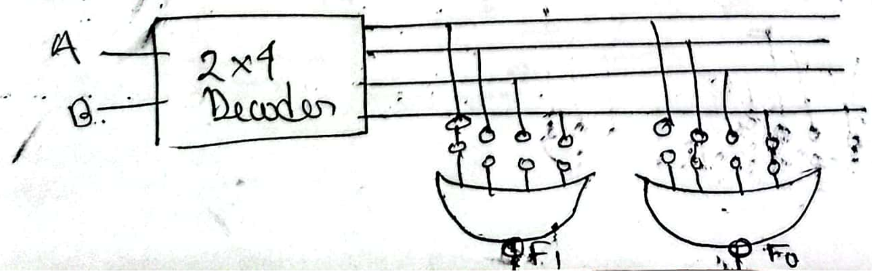


(*) Given $F_1(A, B) = \Sigma(1, 2, 3)$
$F_0(A, B) = \Sigma(0, 2)$ Draw – ROM with AND OR gates

A	B	F_1	F_0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	0

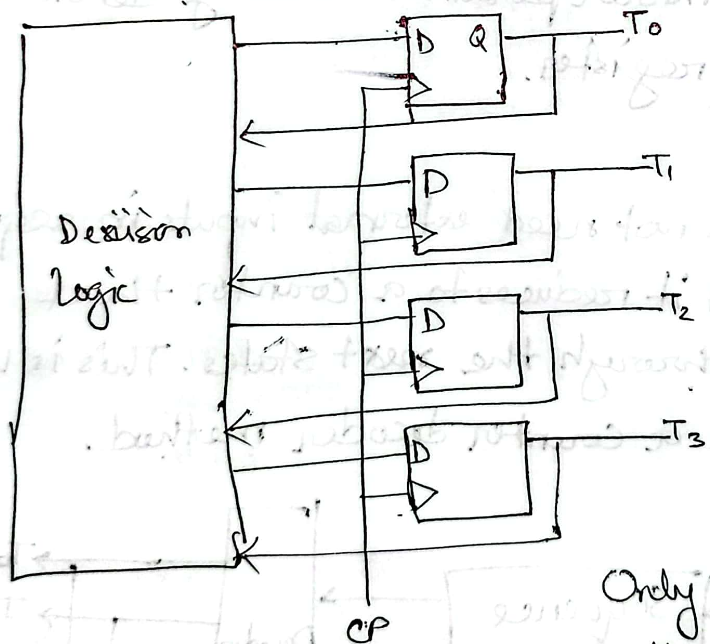


যদি AND OR invert gate থাকে then,



Assignment-2

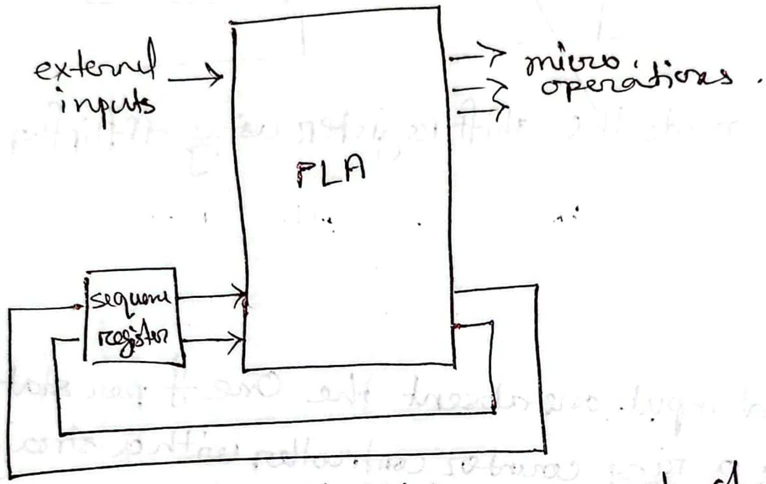
(1) When external inputs are absent the One ff per state method reduces to a ring counter controller with a straight single bit shift register.



Here, as external inputs are not present the control sequence repeats over and over again which makes it a ring counter.

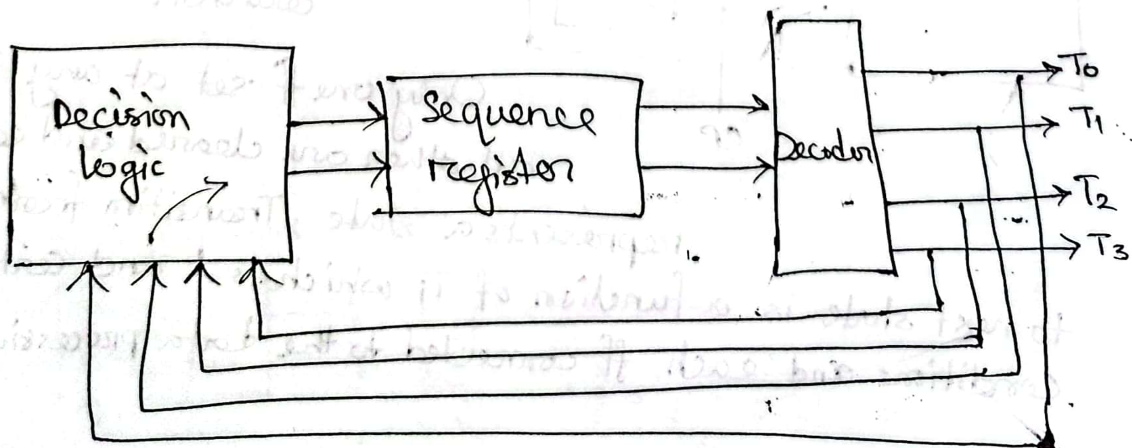
Only one ff set at any time and all other are cleared and each ff represents a state. Transition from present to next state is a function of Ti which is 1 and certain input conditions and each ff connected to the data processing section

② We can improve "Sequence register and decoder" by implementing a PLA instead of combinational circuits and reducing numbers of ICs and wires. A sequence register and decoder consists of decoder and ~~decis~~ decision logic that is replaced by PLA.



external sequence register establishes present states and the outputs of PLA initiates micro operation including determining next states of sequence register.

If the control circuit does not need external inputs in sequence register and decoder that it reduces to a counter that countinously sequences through the next states. This is why it is sometimes called a counter decoder method.



SAP2 codes — subroutine times maths :

দেওয়া থাকে

Tstates

~~MOV~~

MVI → 7 .

DCR → 4

JNZ → 10/7

RET → 10

NOP → 4

From PDF :

① time = 1 μs or (frequency বলে দিবে → 1MHz)

$f = \frac{1}{t}$

Code :

```
        MVI C, 46H
AGAIN : DCR C
        JNZ AGAIN
        NOP
        RET .
```

Solution :— 46 H = 70(Dec) . ⟶ [formula → loops × tstates × time]

MVI C : 1 × 7 × 1 μs

DCR C : 70 × 4 × 1

JNZ : 69 × 10 × 1

no JNZ : 1 × 7 × 1

NOP : 1 × 4 × 1

RET : 1 × 10 × 1 .

(+)

998 μs ≈ 1 ms

Code :

```
        MVI B, 0AH
LOOP1 : MVI C, 47H
LOOP2 : DCR C
        JNZ LOOP2
        DCR B
        JNZ LOOP
        RET
```

Solution :— ~~Here,~~

inner loop থেকে start করব

~~MVI C : 71 × 7 × 1 μs~~

DCR C : 71 × 4 × 1 μs .

JNZ : 70 × 10 × 1

no JNZ : (+) 1 × 7 × 1

991 μs

Here,

0AH = 10D

47H = 71D .

বাইরের loop-এ আসব

mvI C : 10× 7 × 1μs

loop2 : 10× 991

DCR B : 10 × 4 × 1

JNZ : 9×10×1

no JNZ : 1 × 7 ×1

mvI B : 1 × 7 × 1μs

(+)

10124 μs

RET : 1×10 × 1μs + (10124 μs)

delay time = 10134 μs (Ans)

Fall-22-এর code :

t = 0.5μs

```
       MVI C, 05H
START: MVI B, 65H
ODD:   MVI A, 32H
EVEN:  DCR A
       JNZ EVEN
       DCR B
       JNZ ODD
       DCR C
       JNZ START
       RET
```

Solution :-

(inner loop আগে) →

DCR A : 50× 4×0.5 μs

JNZ EVEN 49 × 10 ×0.5

no JNZ : 1 × 7×0.5

(+)

348.5μs

Here,

05H = 5D

65H = 101D

32H = 50D

MVI A : 101× 7× 0.5

EVEN : 101 × 348.5

DCR B : 101× 4×0.5

JNZ ODD : 100 ×10 ×0.5

no JNZ : 1 × 7 ×0.5

36257.5

বাইরের loop গুলো → MVI B : 5 × 7 × 0.5 μs . + Loop ODD : 5 × 36257.5 .
DCR C : 5 × 4 × 0.5 μs
JNZ START : 4 × 10 × 0.5 .
no. JNZ : 1 × 7 × 0.5

MVI C : 1 × 7 × 0.5
RET : 1 × 10 × 0.5

181834.7 μs .

Fall-22

2/b: Handshaking sequence:

1. READY bit (bit 0, port 2) goes high.
2. Input the data in port 1 to CPU.
3. ACKNOWLEGE bit (bit 7, port 4) goes high to reset READY bit.
4. Reset the ACKNOWLEGE bit

During execution, all T states do nothing when we give NOP or No Operation. It is used to waste time or delay data processing.
It take 4 T states to fetch NOP.
So if we put NOP in a loop that executes 100 times, a delay of 400T states is created.

Quiz-4

Set -A

② Instruction cycle for ADD 9H

T_1 : $E_P \ \overline{L_m}$

T_2 : C_P

T_3 : $\overline{CE}$ $\overline{L_I}$

T_4 : $\overline{L_m}$. $\overline{E_I}$

T_5 : $\overline{CE}$ $\overline{L_B}$

T_6 : $\overline{L_A}$ E_U

Set B

② Instruction cycle for LDA 7H

T_1 : $E_P \ \overline{L_m}$

T_2 : C_P

T_3 : $\overline{CE}$, $\overline{L_I}$

T_4 : $\overline{L_m}$ $\overline{E_I}$

T_5 : $\overline{CE}$ $\overline{L_A}$

T_6 : ×

Set -C

② microinstructions for ADD.

T_1 : MAR ← PC

T_2 : PC ← PC-1

T_3 : IR ← RAM[MAR]

T_4 : MAR ← IR (3....0)

T_5 : B ← RAM[MAR]

T_6 : ACC ← ACC+B

Control Words for Machine Cycle : ADD

	C_P	E_P	$\overline{L_m}$	$\overline{CE}$	$\overline{L_I}$	$\overline{E_I}$	$\overline{L_A}$	E_A	S_U	E_U	$\overline{L_B}$	$\overline{L_O}$	
T_1, CON =	0	1	0	1	1	1	1	0	0	0	1	1	= 5E3
T_2, CON =	1	0	1	1	1	1	1	0	0	0	1	1	= BE3
T_3, CON =	0	0	1	0	0	1	1	0	0	0	1	1	= 273
T_4, CON =	0	0	0	1	1	0	1	0	0	0	1	1	= 1A3
T_5, CON =	0	0	1	0	1	1	1	0	0	0	0	1	= 2E1
T_6, CON =	0	0	1	1	1	1	0	0	0	1	1	1	= 3C7

Set-D

② Micro operations of LDA.

T_1 : MAR ← PC
T_2 : PC ← PC − 1
T_3 : IR ← RAM
T_4 : ~~IR~~ MAR ← IR
T_5 : ACC ← RAM.
T_6 : ×

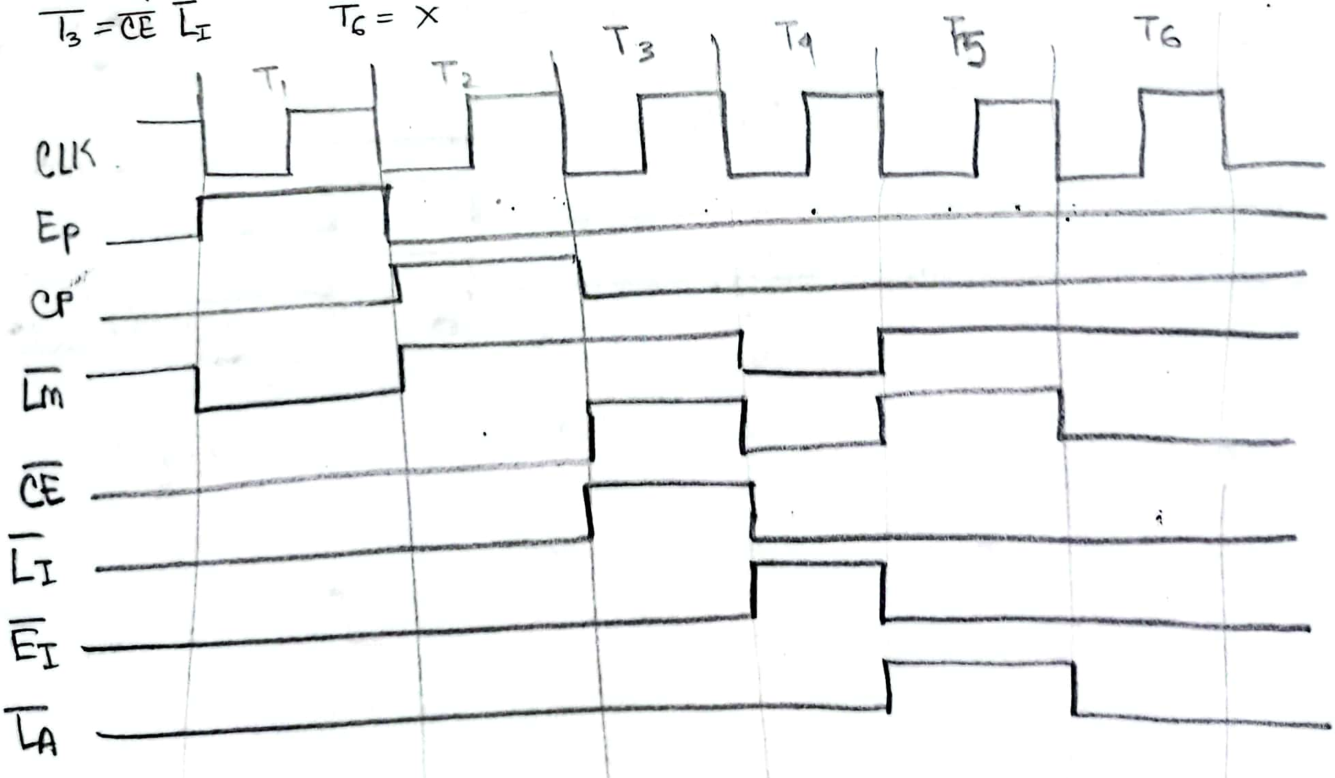
Control words for LDA :

CON =	C_P	E_P	$\overline{L_m}$	$\overline{CE}$	$\overline{L_I}$	$\overline{E_I}$	$\overline{L_A}$	E_A	S_U	E_U	$\overline{L_B}$	$\overline{L_O}$	
T_1 =	0	1	0	1	1	1	1	0	0	0	1	1	= 5E3
T_2 =	1	0	1	1	1	1	1	0	0	0	1	1	= BE3
T_3 =	0	0	1	0	0	1	1	0	0	0	1	1	= 2B3
T_4 =	0	0	0	1	1	0	1	0	0	0	1	1	= 1A3
T_5 =	0	0	1	0	1	1	0	0	0	0	1	1	= 2C3
T_6 =	0	0	1	1	1	1	1	0	0	0	1	1	= 3E3

Set-E

② Timing diagram for LDA 6H :

$T_1 = E_P\ \overline{L_m}$
$T_2 = C_P$
$T_3 = \overline{CE}\ \overline{L_I}$
$T_4 = \overline{L_m}\ \overline{E_I}$
$T_5 = \overline{CE}\ \overline{L_A}$
$T_6 = ×$



Sd - F

(2) Timing diagram for SUB 8H.

T_1: E_p $\overline{L_m}$

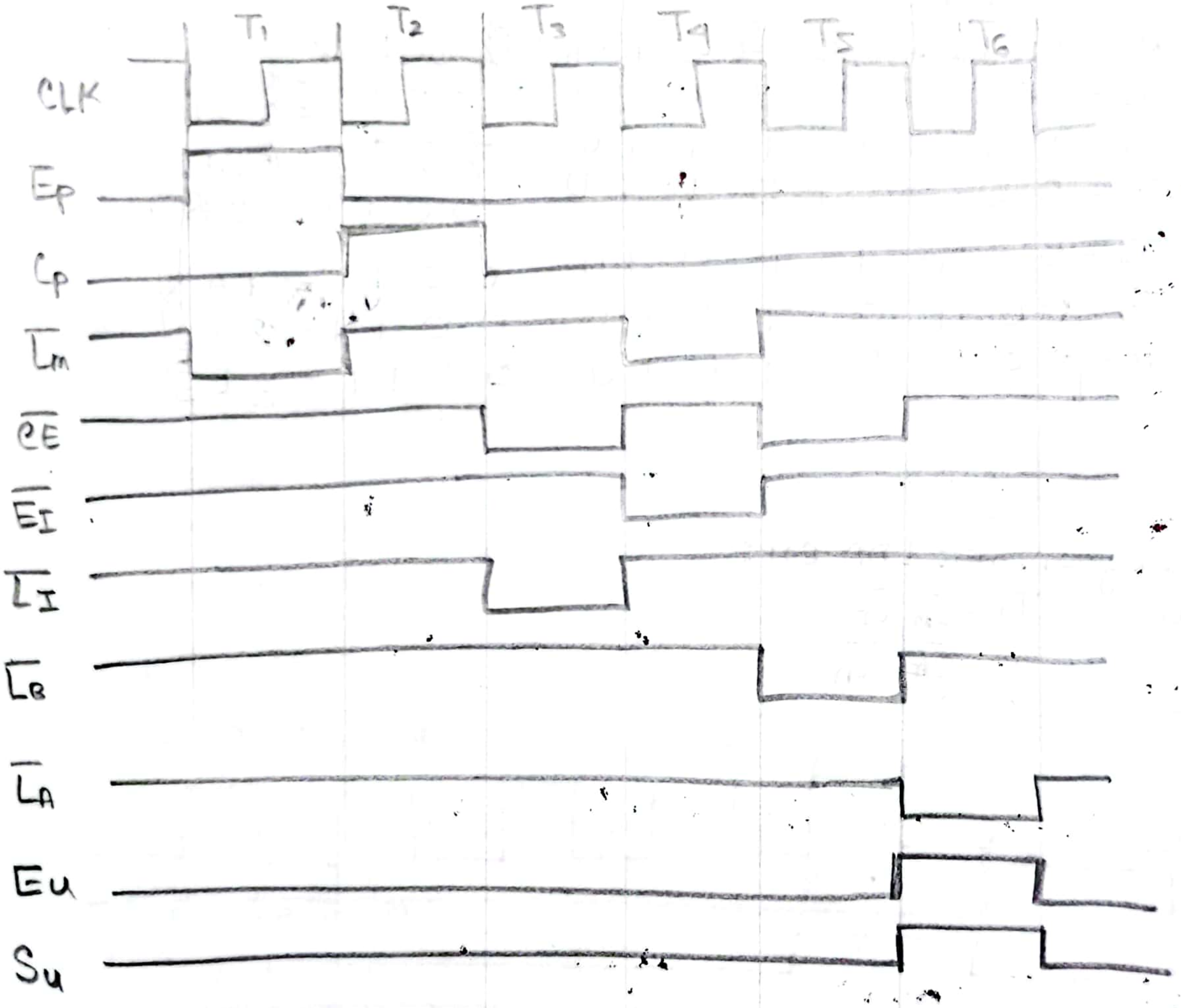
T_2: C_p

T_3: $\overline{CE}$ $\overline{L_I}$

T_4: $\overline{L_m}$ $\overline{E_I}$

T_5: $\overline{CE}$ $\overline{L_B}$

T_6: $\overline{L_A}$ E_U S_U



Quiz-2 | Set G :- 2048/16 = 128 ; range = $(0000\,000 - 1111\,111)_2$ or $(0-127)_{10}$.

01 → +4
10 → −4

Set-A

① Design বলা (UV করব) Given, −17 × 23.

$x = -17 = 01111$

$y = 23 = 10111$

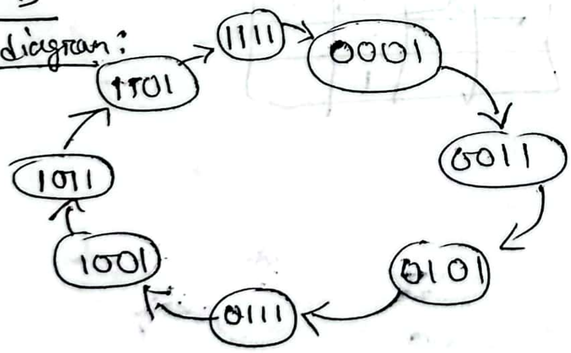
$-y = -23 = 01001$

U	V	x	x-1	steps
00000 (−y)+ 01001 01001 rs→ 00100	00000 10000	0111 $\boxed{1}$ 0011 $\boxed{1}$	0 1	1
(+) 0 00100				2
rs→ 000100 + 0 00010	01000	0001 $\boxed{1}$	1	3
rs→ 00001 + 0 00001	00100	0000 $\boxed{1}$	1	4
rs→ 00000 (+y) + 01111 01111	10010	0000 $\boxed{0}$	1	5

So, rs→ 0011111001 = Final result. (মিলে না ☹)

Set-B

State diagram:



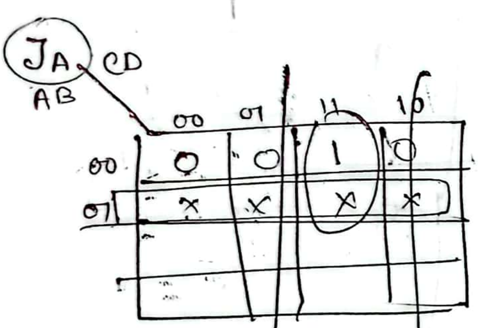
~~State table:~~

00 0X
01 1X
10 X1
11 X0

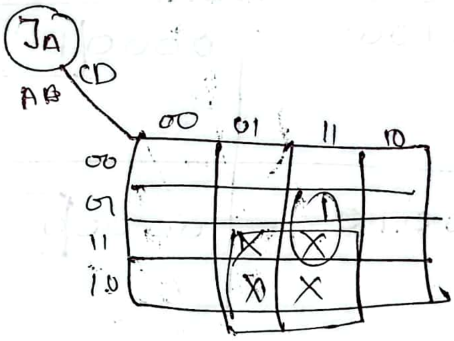
State Table:

PS				NS				FF							
A	B	C	D	A	B	C	D	J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D
0	0	0	1	0	0	1	1	0	X	0	X	1	X	X	0
0	0	1	1	0	1	0	1	0	X	1	X	X	1	X	0
0	1	0	1	0	1	1	1	0	X	X	0	1	X	X	0
0	1	1	1	1	0	0	1	1	X	X	1	X	1	X	0
1	0	0	1	1	0	1	1	X	0	0	X	1	X	X	0
1	0	1	1	1	1	0	1	X	0	1	X	X	1	X	0
1	1	0	1	1	1	1	1	X	0	X	0	1	X	X	0
1	1	1	1	0	0	0	1	X	1	X	1	X	1	X	0

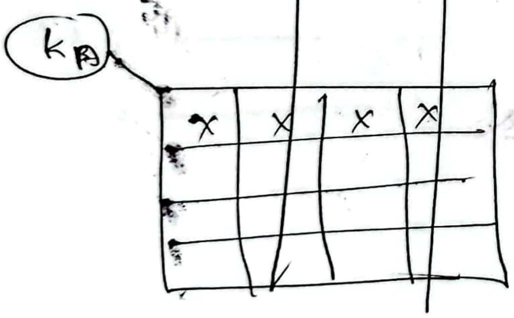
3
5
7
9
11
13
15

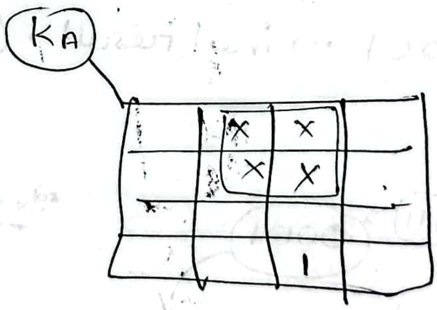


$J_A = A'B + CD.$



$J_A = AD + BCD.$





$K_A =$

set D

$F1 = abc + \bar{a}b + a.$

$= abc + \bar{a}bc + \bar{a}b\bar{c} + ab + a\bar{b}$

$= abc + \bar{a}bc + \bar{a}b\bar{c} + ab\bar{c} + a\bar{b}\bar{c} + a\bar{b}c$

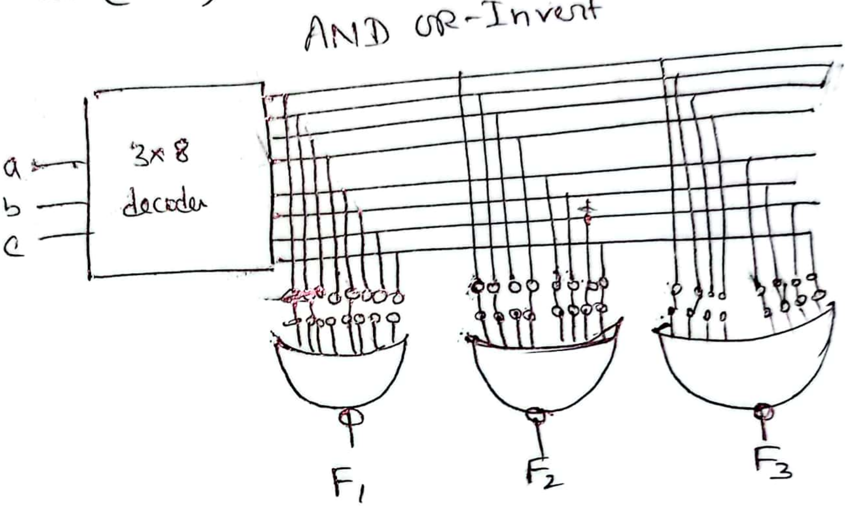
$= \Sigma(7, 3, 2, 6, 5)$

$F2 = \Sigma(2)$

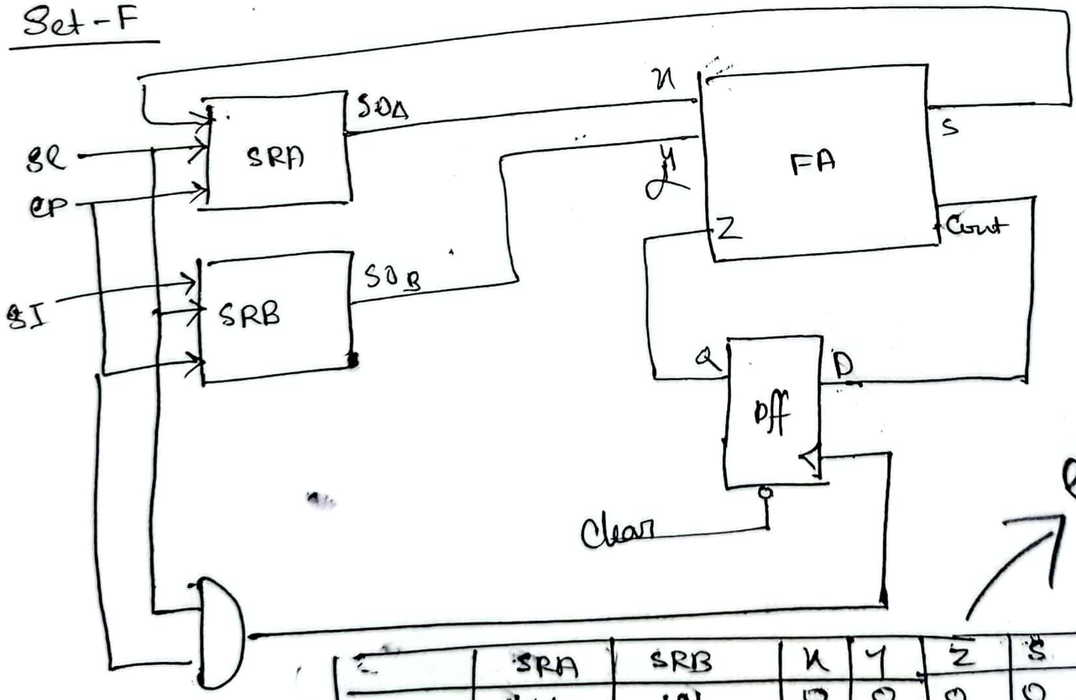
$F3 = \Sigma(\bar{a}cb + \bar{a}\bar{c}b) = \Sigma(3, 2)$

a	b	c	F1	F2	F3
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

AND OR-Invert



Set-F



explain করা লাগবে

	SRA	SRB	x	y	z	S	C
	111	101	0	0	0	0	0
	011	10	1	1	0	0	1
	001	1	1	0	1	0	1
	100	x	1	1	1	1	1

So, SRA = 100 and CarryOut = 1.