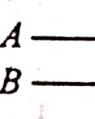
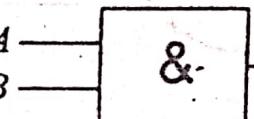
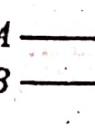
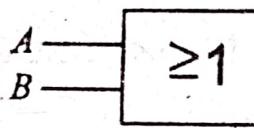
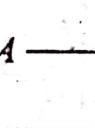
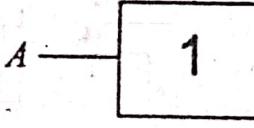
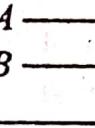
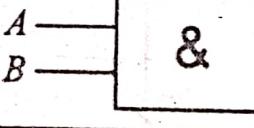
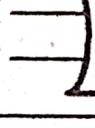
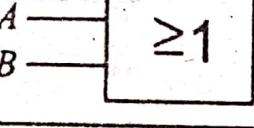
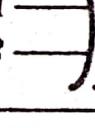
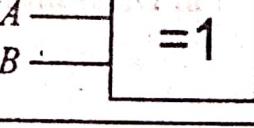
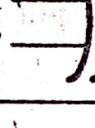
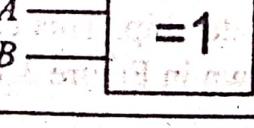


3.3 IEEE Symbols for Logic Gates

The institute of Electrical and Electronics Engineers (IEEE) recommends rectangular shape symbols for logic gates: The original logic symbols have been utilized for years and will be retained in the rest of this book. IEEE symbols for gates are listed below:

Gate	Common Symbol	IEEE Symbol
AND	 $f = AB$	 $f = AB$
OR	 $f = A + B$	 $f = A + B$
NOT	 $f = \bar{A}$	 $f = \bar{A}$
NAND	 $f = \overline{AB}$	 $f = \overline{AB}$
NOR	 $f = \overline{A + B}$	 $f = \overline{A + B}$
Exclusive-OR	 $f = A \oplus B$	 $f = A \oplus B$
Exclusive-NOR	 $f = \overline{A \oplus B}$	 $f = \overline{A \oplus B}$

Boolean Algebra: A system of algebra in which there are only two possible values for a variable (often expressed as true and false or as 1 and 0) and in which the basic operations (are) the logical operations AND and OR.

Boolean Identities:

$$\textcircled{1} \quad A + 0 = A$$

$$A \cdot 0 = 0$$

$$\textcircled{2} \quad A + 1 = 1$$

$$A \cdot 1 = A$$

$$\textcircled{3} \quad A + A = A$$

$$A \cdot A = A$$

$$\textcircled{4} \quad A + \bar{A} = 1$$

$$A \cdot \bar{A} = 0$$

$$\textcircled{5} \quad \bar{\bar{A}} = A$$

6. Commutative Law:

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

7. Associative Law:

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

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

8. Distributive Law:

$$A + (B + C) = A \cdot B + A \cdot C$$

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$

9. De Morgan's Theorem:

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

$$\overline{AB} = \overline{A} + \overline{B}$$

$$I = L + A$$

$$\begin{aligned}
 \text{Q1} \quad f &= ABCD + \overline{ABCD} + \overline{BC} \\
 &= BCD(A + \bar{A}) + \overline{BC} \quad [\because A + \bar{A} = 1] \\
 &= BCD + \overline{BC} \\
 &= XD + \bar{X} \quad [\text{let, } X = BC] \\
 &= (X + \bar{X}) \cdot (\bar{X} + D) \\
 &= \bar{X} + D \\
 &= \overline{BC} + D
 \end{aligned}$$

(Proved)

Congruence Theorem:

$$\begin{aligned}
 AB + \overline{AC} + BC &= AB + \overline{AC} + BC(\bar{A} + A) \\
 &= AB + \overline{AC} + \bar{A}BC + ABC \\
 &= AB + ABC + \overline{AC} + \bar{A}BC \\
 &= AB(1 + C) + \overline{AC}(1 + B) \\
 &= AB + \overline{AC}
 \end{aligned}$$

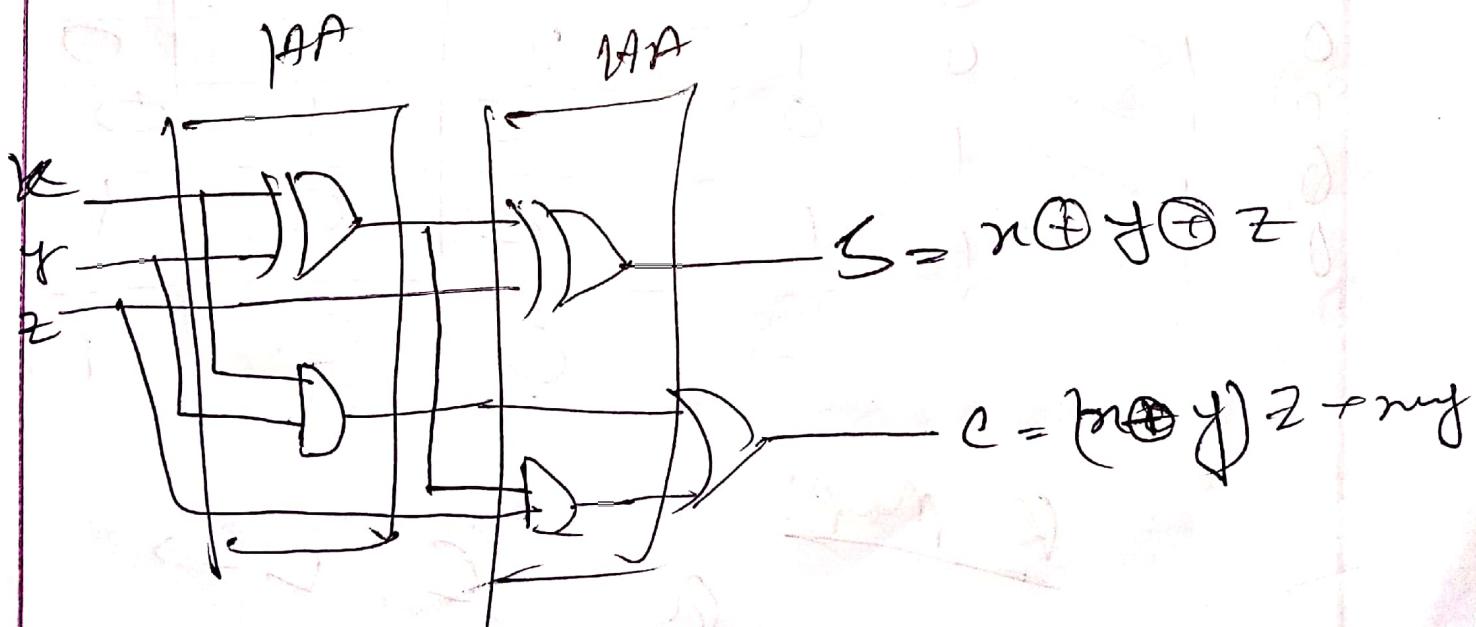
(Proved)

$$\overline{ABC\bar{D}} + \overline{AB\bar{C}D}$$

$$\begin{aligned}
 F &= AB + \overline{A}\overline{B}\overline{C}\overline{D} + CD + \overline{A}\overline{B}\overline{C}D \\
 &= AB(C + \overline{C}) + \overline{A}\overline{B}\overline{C}\overline{D} + CD(B + \overline{B}) + \overline{A}\overline{B}\overline{C}D \\
 &= ABC + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C}\overline{D} + BC\overline{D} + \overline{B}\overline{C}D + \overline{A}\overline{B}\overline{C}D \\
 &= ABC(D + \overline{D}) + A\overline{B}\overline{C}(D + \overline{D}) + \overline{A}\overline{B}\overline{C}\overline{D} + (A + \overline{A})BC\overline{D} \\
 &\quad + (A + \overline{A})\overline{B}CD + \overline{A}\overline{B}\overline{C}D
 \end{aligned}$$

$$\begin{aligned}
 &= \overline{1110} \cdot 0 + \overline{1110} \cdot 1 + \overline{1011} \cdot 0 + \overline{1011} \cdot 1 + \overline{0110} \cdot 0 + \overline{0110} \cdot 1 + \overline{1010} \cdot 0 + \overline{1010} \cdot 1 + \overline{0010} \cdot 0 + \overline{0010} \cdot 1 + \overline{0001} \cdot 0 + \overline{0001} \cdot 1 \\
 &= \overline{ABC}D + A\overline{B}CD + A\overline{B}C\overline{D} + A\overline{B}\overline{C}D + \overline{ABC}\overline{D} + A\overline{B}C\overline{D} + \overline{ABC}\overline{D} + A\overline{B}\overline{C}\overline{D} + \overline{ABC}\overline{D} + A\overline{B}CD + \overline{ABC}D + A\overline{B}C\overline{D}
 \end{aligned}$$

$$= \text{m}_0 = \text{m}_1 = \text{m}_2 = \text{m}_3 = \text{m}_4 = \text{m}_5 = \text{m}_6 = \text{m}_7 = \text{m}_8 = \text{m}_9 = \text{m}_{10} = \text{m}_{11} = 0$$



<u>A</u>	<u>B</u>	<u>SOP</u> minterm	<u>Symbol</u>	<u>(POS)</u> Maxterm	<u>Symbol</u>
0	0	$\bar{A}\bar{B}$	m_0	$A+B$	M_0
0	1	$\bar{A}B$	m_1	$A+\bar{B}$	M_1
1	0	$A\bar{B}$	m_2	$\bar{A}+\bar{B}$	M_2
1	1	AB	m_3	$\bar{A}+\bar{B}$	M_3

$$(0+\bar{x}) \cdot (\bar{x}+x) =$$

$$m_1 = \bar{A}B$$

$$0+\bar{x} =$$

$$\Rightarrow \overline{m}_1 = \overline{\bar{A}B}$$

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

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

$$= M_1$$

$$(A+A)x = Ax$$

$$A + 2m_1 = \overline{M_1} / \overline{m}_1 = M_1$$

provided

$$(\bar{A}+\bar{B})x + (\bar{A}+\bar{B})\bar{x} =$$

$$(\bar{A}+\bar{B})x + (0+1)x =$$

$$\bar{A}x + \bar{B}x =$$

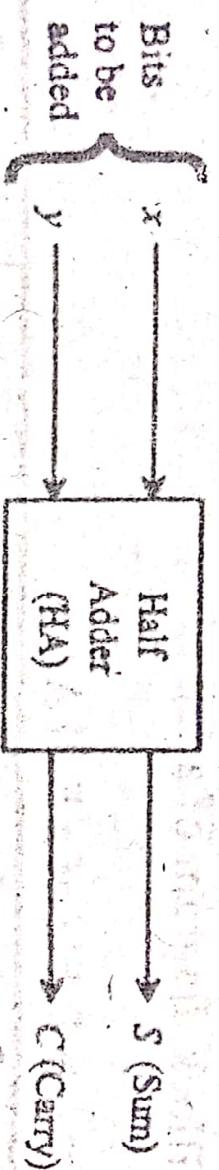


FIGURE 4.6 Block Diagram of a Half-Adder

TABLE 4.5 Truth Table of the Half-Adder

<u>Inputs</u>		<u>Outputs</u>		<u>Decimal Value</u>
x	y	C	S	
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	2

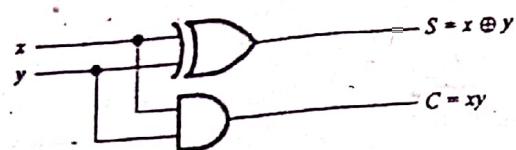


FIGURE 4.7 Logic diagram of the half-adder

This combination will provide 16 (2^4) combinations of 1's and 0's. Because only ten of these combinations (0000 through 1001) are allowed in BCD, the invalid combinations 1010 through 1111 can never occur in BCD. Therefore, these six binary inputs are considered as don't cares. This means that it does not matter what binary values are assumed by f_3, f_2, f_1, f_0 for $WXYZ = 1010$ through 1111. Figure 4.5 shows the K-maps and the logic circuit.

4.5 Typical Combinational Circuits

This section describes typical combinational circuits. Topics include binary adders, subtractors, comparators, decoders, encoders, multiplexers, and demultiplexers. These digital components are implemented in MSI chips.

4.5.1 Binary / BCD Adders and Binary Subtractors

When two bits x and y are added, a sum and a carry are generated. A combinational circuit that adds two bits is called a "half-adder." Figure 4.6 shows a block diagram of the half-adder. Table 4.5 shows the truth table of the half-adder. From Table 4.5, $S = \bar{x}y + x\bar{y} = x \oplus y$, $C = xy$

Figure 4.7 shows the logic diagram of the half-adder.

Next, consider addition of two 4-bit numbers as follows (next page):

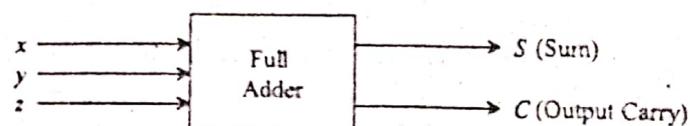
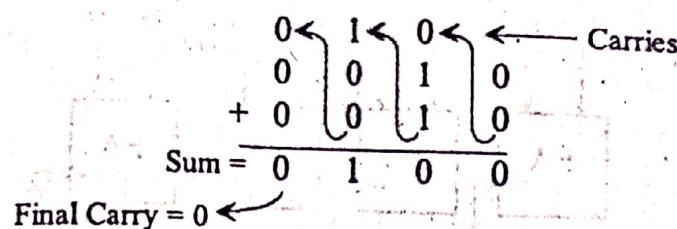


FIGURE 4.8 Block diagram of a full adder

TABLE 4.6 Truth Table of a Full Adder

Inputs			Outputs		Decimal Value
x	y	z	C	S	
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	2
1	0	0	0	1	1
1	0	1	1	0	2
1	1	0	1	0	2
1	1	1	1	1	3



This addition of two bits will generate a sum and a carry. The carry may be 0 or 1. Also, there will be no previous carry while adding the least significant bits (bit 0) of the two numbers. This means that two bits need to be added for bit 0 of the two numbers. On the other hand, addition of three bits (two bits of the two numbers and a previous carry, which may be 0 or 1) is required for all the subsequent bits. Note that two half-adders are required to add three bits. A combinational circuit that adds three bits, generating a sum and a carry (which may be 0 or 1), is called a "full adder." Figure 4.8 shows the block diagram of a full adder. The full adder adds three bits, x , y , and z , and generates a sum and a carry. Table 4.6 shows the truth table of a full adder.

From the truth table, $S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz = (\bar{x}y + x\bar{y})\bar{z} + (xy + \bar{x}\bar{y})z$

Let $w = \bar{x}y + x\bar{y}$ then $\bar{w} = xy + \bar{x}\bar{y}$. Hence, $S = w\bar{z} + \bar{w}z = w \oplus z = x \oplus y \oplus z$

Also, from the truth table, $C = \bar{x}yz + \bar{x}y\bar{z} + xy\bar{z} + xyz = (\bar{x}y + x\bar{y})z + xy(z + \bar{z})$
 $= wz + xy$

where $w = (\bar{x}y + x\bar{y}) = x \oplus y$. Hence, $C = (x \oplus y)z + xy$.

Another form of Carry can be written as follows:

$$\begin{aligned} C &= \bar{x}yz + \bar{x}y\bar{z} + xy\bar{z} + xyz = \bar{x}yz + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz + xyz \\ &= yz(\bar{x} + x) + xz(y + \bar{y}) + xy(z + \bar{z}) = yz + xz + xy \end{aligned}$$

Figure 4.9 shows the logic diagram of a full adder.

Note that the names half-adder and full adder are based on the fact that two half-adders are required to obtain a full adder. This can be obtained as follows. One of the two half-adders with inputs, x and y will generate the sum, $S_0 = x \oplus y$ and the carry, $C_0 = xy$. The sum (S_0) output can be connected to one of the inputs of the second half-adder with z as

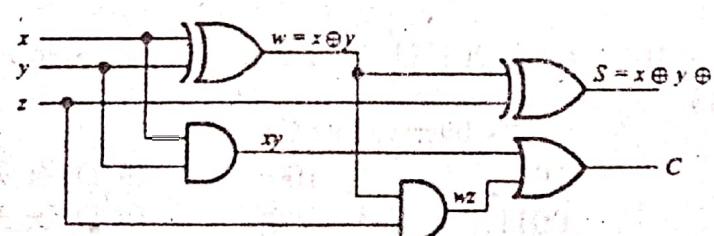


FIGURE 4.9 Logic diagram of a full adder

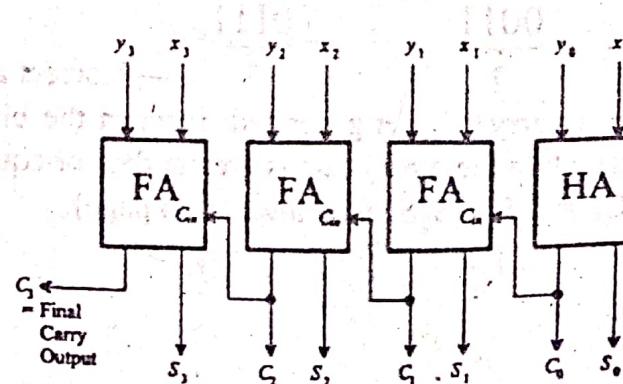


FIGURE 4.10 4-bit binary adder using one half-adder and three full adders

From the above truth table, we have
 $R = x \oplus y \oplus z$ and $B = \bar{x}y + \bar{x}z + yz$.
 It is advantageous to implement addition and subtraction with full-adders since both operations can be obtained using a single logic circuit.

4.5.2 Comparators

The digital comparator is a widely used combinational system. Figure 4.13 shows a 2-bit

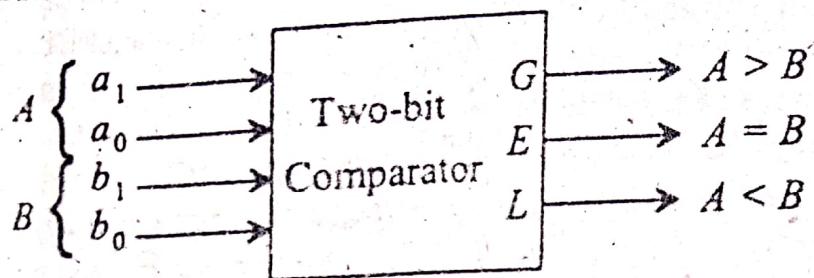


FIGURE 4.13 Block diagram of a two-bit comparator

TABLE 4.1 Truth Table for the 2-Bit Comparator

Inputs				Outputs		
a_1	a_0	b_1	b_0	G	E	L
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

K-map for G :

$a_1 \backslash a_0$	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

$$G = a_1 \bar{b}_1 + a_0 \bar{b}_1 \bar{b}_0 + a_1 a_0 \bar{b}_0$$

K-map for E :

$a_1 \backslash a_0$	00	01	11	10
00	1			
01		1		
11			1	
10				1

$$\begin{aligned} E &= \bar{a}_1 \bar{a}_0 \bar{b}_1 \bar{b}_0 + \bar{a}_1 a_0 \bar{b}_1 b_0 + a_1 a_0 b_1 b_0 + a_1 \bar{a}_0 b_1 \bar{b}_0 \\ &= \bar{a}_1 \bar{b}_1 (\bar{a}_0 \bar{b}_0 + a_0 b_0) + a_1 b_1 (a_0 b_0 + \bar{a}_0 \bar{b}_0) \\ &= (a_0 b_0 + \bar{a}_0 \bar{b}_0)(a_1 b_1 + \bar{a}_1 \bar{b}_1) \\ &= (a_0 \odot b_0)(a_1 \odot b_1) \end{aligned}$$

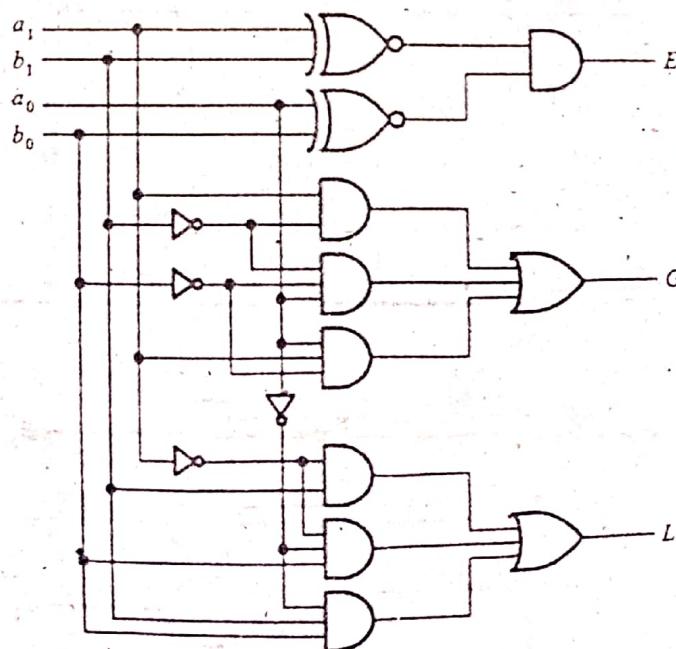
⊕ → K-NOR

K-map for L :

$a_1 \backslash a_0$	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

$$L = \bar{a}_1 b_1 + \bar{a}_0 b_1 b_0 + \bar{a}_1 \bar{a}_0 b_0$$

a) K-maps for the 2-bit comparator



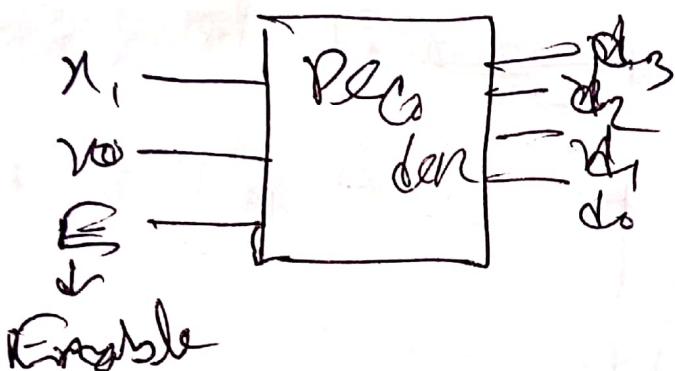
b) Logic Diagram of the 2-bit comparator

FIGURE 4.14 Design of a 2-bit comparator

Decoder!

Input: n

Output: 2^n



Truth Table:

<u>R</u>	<u>d₁</u>	<u>d₂</u>	<u>d₃</u>	<u>d₄</u>	<u>d₅</u>	<u>d₆</u>	<u>d₇</u>	<u>d₈</u>
0	X	X		0	0	0	0	0
1	0	0		1	0	0	0	0
1	0	1		0	1	0	0	0
1	1	0		0	0	1	0	0
1	1	1		0	0	0	1	0

4.5.3 Decoders

An n-bit binary number provides 2^n minterms or maxterms. For example, an n-bit binary number will generate 4 (2^2) minterms or maxterms. A decoder is a combinational logic circuit, when enabled, selects one of 2^n minterms or maxterms at the output based on the input combinations. However, a decoder sometimes may have less than 2^n outputs. For example, the BCD to seven-segment decoder has 4 inputs and 7 outputs rather than 16 (2^4) outputs.

The block diagram of a 2-to-4 decoder is shown in Figure 4.15. Table 4.8 provides its truth table.

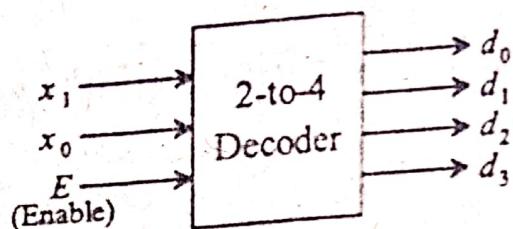


FIGURE 4.15 Block diagram of the 2-to-4 decoder

TABLE 4.8 Truth Table of the 2-to-4 Decoder

Inputs			Outputs			
E	x_1	x_0	d_0	d_1	d_2	d_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Equivalent decimal values

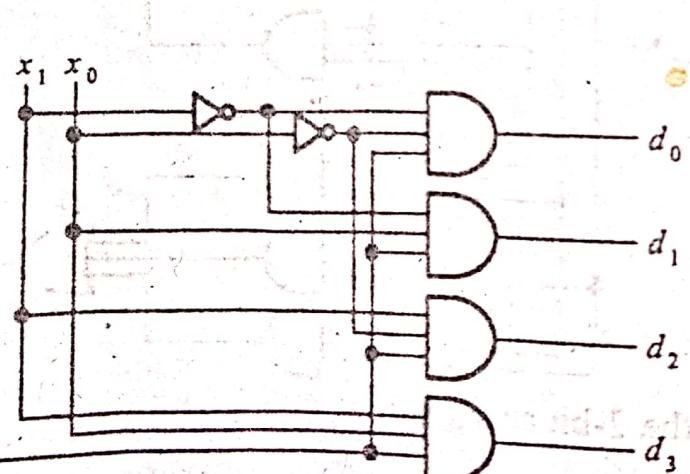


FIGURE 4.16 Logic diagram of the 2-to-4 decoder

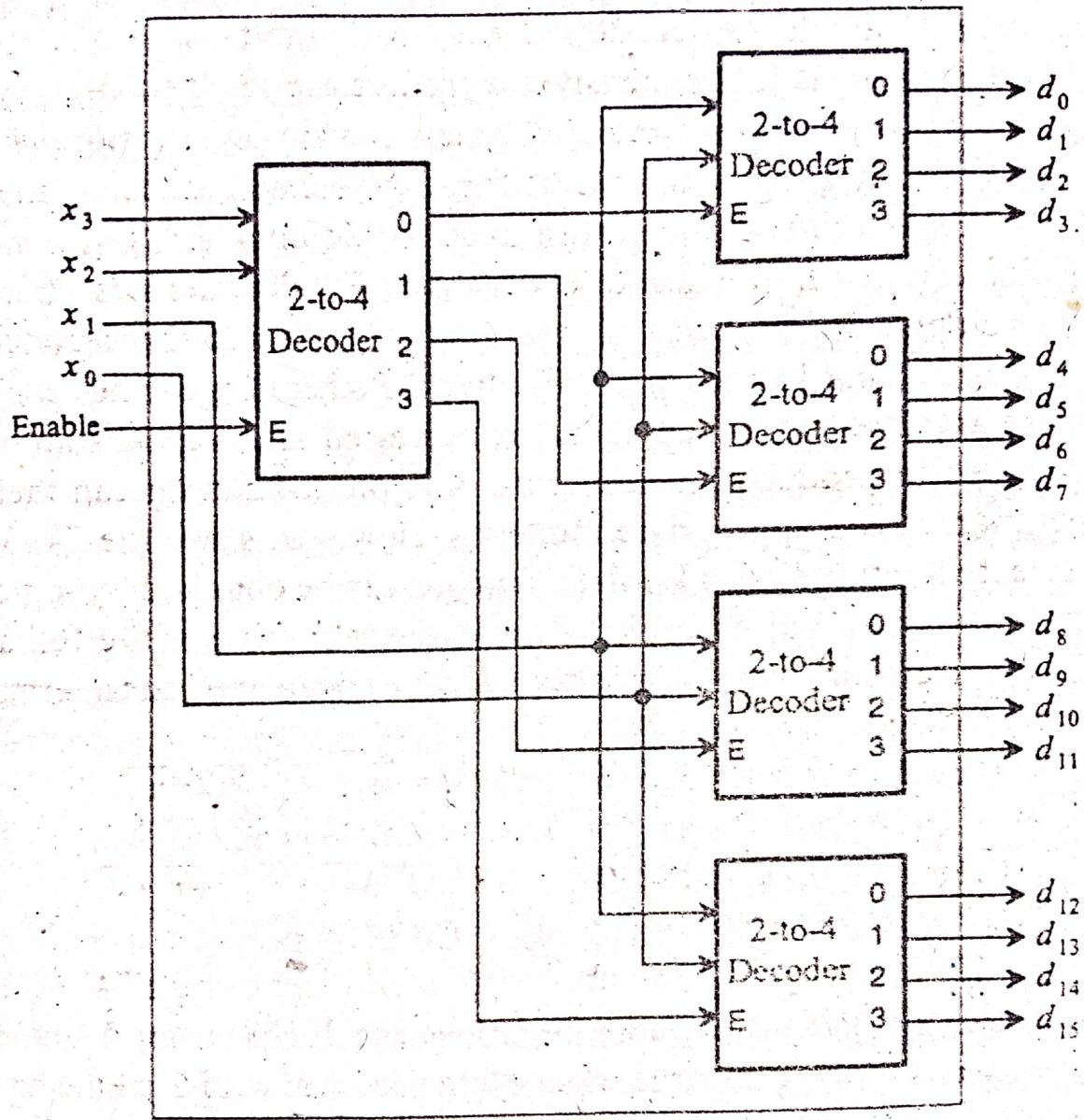


FIGURE 4.11 Implementation of a 4-to-16 Decoder Using 2-to-4 decoders

the truth table. In the truth table, the symbol X is the don't care condition, which can be 0 or 1. Also, $E = 0$ disables the decoder. On the other hand, the decoder is enabled when $E = 1$. For example, when $E = 1$, $x_1 = 0$, $x_0 = 0$, and the output d_0 is HIGH while the other outputs d_1, d_2 , and d_3 are zero. Note that $d_0 = \overline{Ex_1} \overline{x_0}$, $d_1 = \overline{Ex_1} x_0$, $d_2 = Ex_1 \overline{x_0}$, and $d_3 = Ex_1 x_0$. Therefore, the 2-to-4 line decoder outputs one of the four minterms of the two input variables x_1 and x_0 when $E = 1$. In general, for n inputs, the n -to 2^n decoder when enabled selects one of 2^n minterms or maxterms at the output based on the input combinations. The decoder actually provides binary to decimal conversion operation. Using the truth table

the appropriate minterms.

Table 4.6. The inverted sum and the inverted carry are as follows:

$$\overline{SUM} = \sum m(0, 3, 5, 6), \quad SUM = \overline{m_0} \cdot \overline{m_3} \cdot \overline{m_5} \cdot \overline{m_6}$$

$$\overline{CARRY} = \sum m(0, 1, 2, 4), \quad CARRY = \overline{m_0} \cdot \overline{m_1} \cdot \overline{m_2} \cdot \overline{m_4}$$

Figure 4.18 shows the implementation of a full adder using a 74138 decoder ($C=X$, $B=Y$, $A=Z$) and two 4-input AND gates. Note that the 74138 in the Manufacturer's data book uses the symbols C , B , A as three inputs to the decoder with C as the most significant

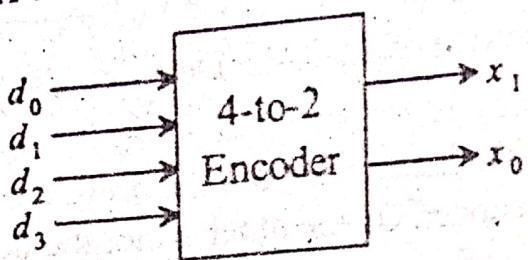


FIGURE 4.19 Block diagram of a 4-to-2 encoder

TABLE 4.9 Truth Table of the 4-to-2 Encoder

Inputs				Outputs	
d_0	d_1	d_2	d_3	x_1	x_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

TABLE 4.10 Truth Table of the 4-to-2 Priority Encoder

Inputs				Outputs	
d_0	d_1	d_2	d_3	x_1	x_0
1	0	0	0	0	0
X	1	0	0	0	1
X	X	1	0	1	0
X	X	X	1	1	1

X means don't care

$d_2 d_3$	00	01	11	10
$d_0 d_1$	X	1	1	0
00	1	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

$d_2 d_3$	00	01	11	10
$d_0 d_1$	X	1	1	1
00	0	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

 a) K-map for \bar{x}_0

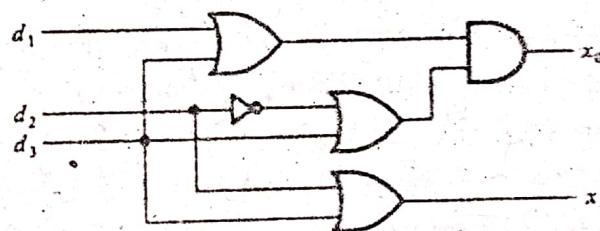
$$\bar{x}_0 = \overline{d_1} \overline{d_3} + d_2 \overline{d_3}$$

$$x_0 = (d_1 + d_3)(\overline{d_2} + d_3)$$

 b) K-map for \bar{x}_1

$$\bar{x}_1 = \overline{d_2} \overline{d_3}$$

$$x_1 = d_2 + d_3$$



c) Logic diagram

FIGURE 4.20 K-maps and logic diagram of a 4-to-2 priority encoder

bit and A as the least significant bit.

4.5.4 Encoders

An encoder is a combinational circuit that performs the reverse operation of a decoder. A encoder has a maximum of 2^n inputs and n outputs. Figure 4.19 shows the block diagram of a 4-to-2 encoder. Table 4.9 provides the truth table of the 4-to-2 encoder.

From the truth table, it can be concluded that an encoder actually performs

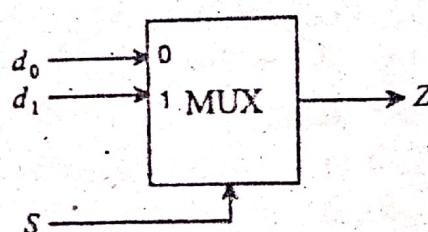


FIGURE 4.21 Block diagram of a 2-to-1 multiplexer

Priorität encoden:

$d_3 d_2 d_1 d_0$

	X	0	0	0
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

$$x_1 = d_2 + d_3$$

$d_3 d_2 d_1 d_0$

	X	0	0	0
-	-	0	0	0
-	0	-	0	0
-	0	0	-	0

$$x_0 = d_3 + d_2$$

1 1 1 1 0

$$\textcircled{1} \quad AB + (1 \oplus A) + A\bar{C} + A\bar{B} + AC$$

$$= AB + \bar{A} + A\bar{C} + A\bar{B} + AC$$

$$[1 \oplus A = \bar{A}]$$

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

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

$$[C + \bar{C} = 1]$$

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

$$= A + 1$$

$$= 1$$

Ans

$$\textcircled{2} \quad (\bar{x} + \bar{y})(\bar{x}\bar{y}) + xy\bar{z} + x\bar{y}\bar{z}$$

$$= (\bar{x} + \bar{y})(\bar{x} + \bar{y}) + xy(z + \bar{z})$$

$$= \bar{x}\bar{x} + \bar{x}\bar{y} + \bar{x}\bar{y} + \bar{y}\bar{y} + xy$$

$$= \bar{x} + \bar{x}\bar{y} + \bar{y} + xy$$

$$= \bar{x}(1 + \bar{y}) + \bar{y} + xy$$

$$= \bar{x} + \bar{y} + xy$$

$$= \bar{x}\bar{y} + xy$$

$$= \cancel{\bar{x}\bar{y} + xy}$$

Answer

(2)

$$ABC + B\bar{B}(C\bar{D})$$

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

$$+ A\bar{B}D + ABD$$

$$+ ABC + A\bar{B}D$$

$$ABC +$$

+

$$B\bar{B}(C\bar{D})$$

$$+ A\bar{B}D + ABD$$

$$+ ABC + A\bar{B}D$$

$$ABC +$$

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

$$+ ABC + A\bar{B}D$$

$$+ ABC + A\bar{B}D$$

$$ABC + B\bar{B}(C\bar{D}) + B\bar{B}\bar{D}$$

$$ABC + A\bar{B}D$$

"

Ans

$$\begin{aligned}
 F &= \overline{\overline{C}(AB + A\bar{B}D + \bar{A}B\bar{D})} \\
 &= \overline{\overline{C} + (AB + A\bar{B}D + \bar{A}B\bar{D})} \\
 &= \overline{C + (\overline{AB} \cdot \overline{A\bar{B}D} \cdot \overline{\bar{A}B\bar{D}})} \\
 &= \overline{C + (\bar{A} + \bar{B})(\bar{A} + B + \bar{D}) \cdot (A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A}\bar{A} + \bar{A} \cdot B + \bar{A}\bar{D} + \bar{A}\bar{B} + \bar{B} \cdot B + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{A}\bar{B} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= \overline{C + (\bar{A} + \bar{B}\bar{D})(A + \bar{B} + D)} \\
 &= C + (\bar{A}\bar{A} + \bar{A}\bar{B} + \bar{A}\bar{D} + \bar{B}\bar{D}A + \bar{B}\bar{D}\cdot\bar{B} + \bar{B}\bar{D}\cdot D)
 \end{aligned}$$

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

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

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

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

~~C, $\bar{A}\bar{B}$, $\bar{A}D$, $\bar{B}\bar{D}$~~

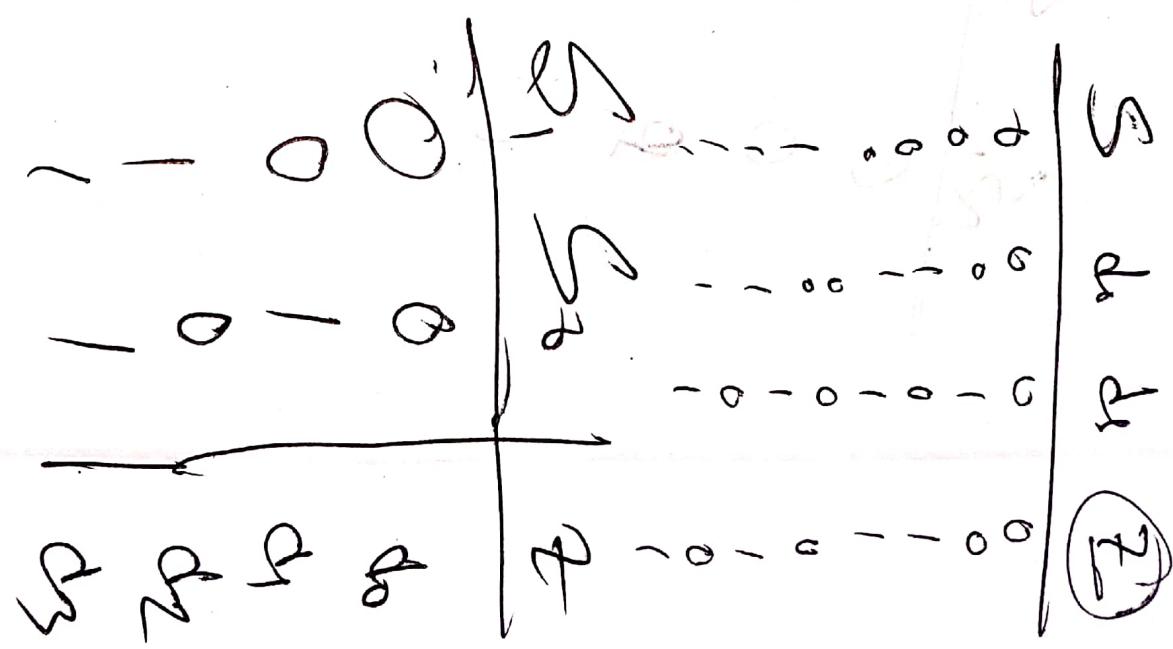
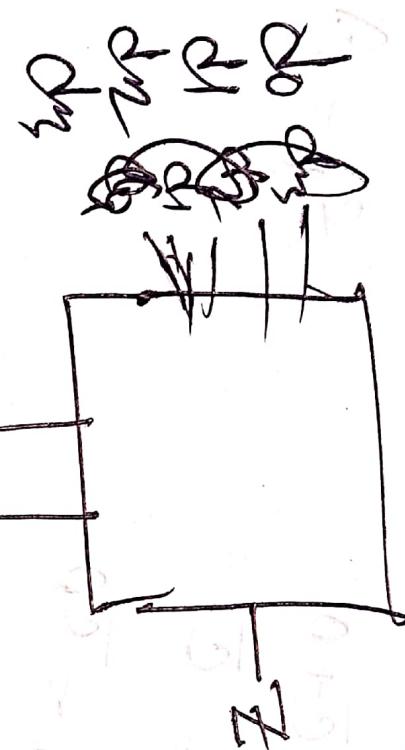
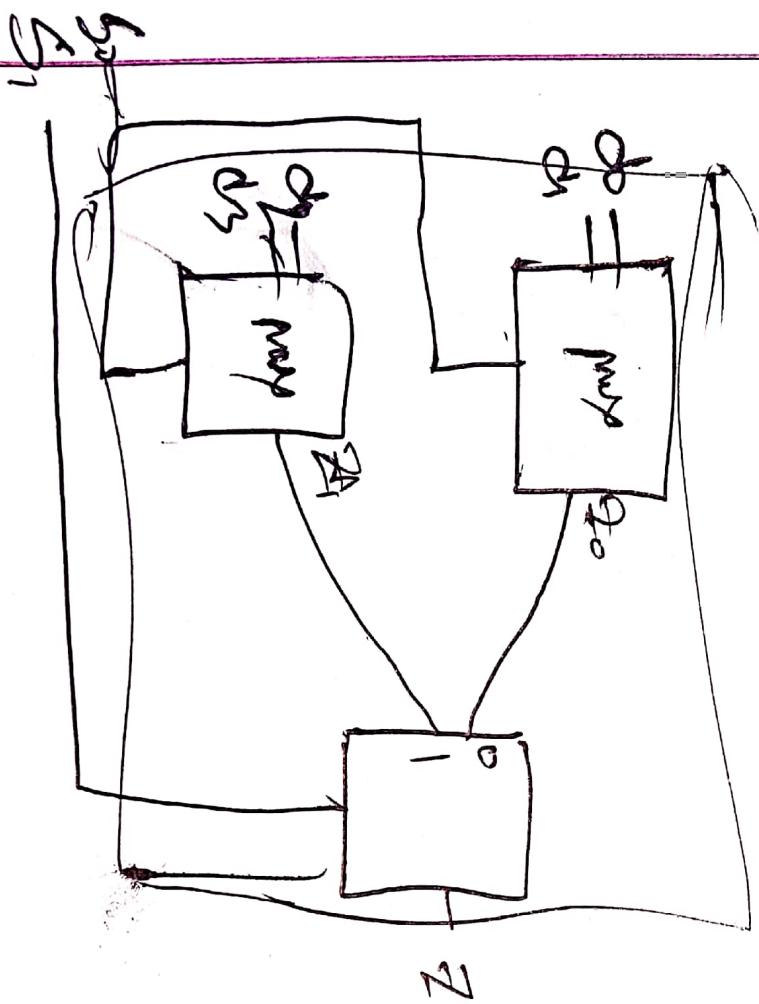
$= \bar{C} \cdot (\bar{A} + \bar{B})(A + \bar{D})(B + D)$

~~A, \bar{B} , \bar{D}~~

~~Final solution, a full row of blank space~~

(1) ~~Working~~

~~Final working~~



of a 4-to-2 encoder. Table 4.9 provides the truth table of the 4-to-2 encoder.

From the truth table, it can be concluded that an encoder actually performs

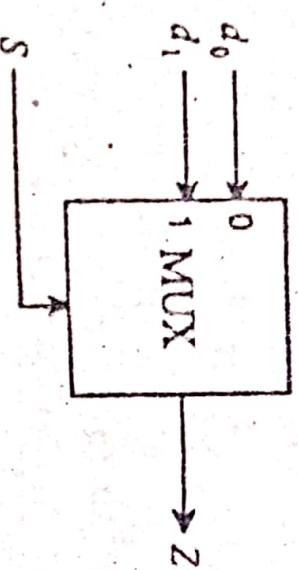


FIGURE 4.21 Block diagram of a 2-to-1 multiplexer

TABLE 4.11 Truth Table of the 2-to-1 Multiplexer

S	d_0	d_1	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$S \backslash d_0 d_1$	00	01	11	10
0			1	1
1		1	1	

FIGURE 4.22 (a) K-map for the 2-to-1 MUX

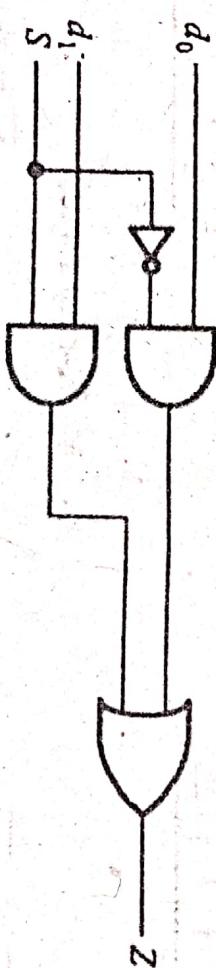


FIGURE 4.22 (b) Logic diagram of the 2-to-1 MUX

decimal-to-binary conversion. In the encoder defined by Table 4.9, it is assumed that only one of the four inputs can be HIGH at any time. If more than one input is 1 at the same time, an undefined output is generated. For example, if d_1 and d_2 are 1 at the same time, both x_0 and x_1 are 1. This represents binary 3 rather than 1 or 2. Therefore, in an encoder in which more than one input can be active simultaneously, a priority scheme must be implemented in the inputs to ensure that only one input will be encoded at the output.

A 4-to-2 priority encoder will be designed next. Suppose that it is assumed that inputs with higher priority A,

shows the logic diagram. In general, a multiplexer will "select" data inputs. Hence, multiplexers are sometimes referred to as "data selectors."

A large multiplexer can be implemented using a small multiplexer as the building block. For example, consider the block diagram and the truth table of a 4-to-1 multiplexer shown in Figure 4.23 and Table 4.12 respectively. The 4-input multiplexer c

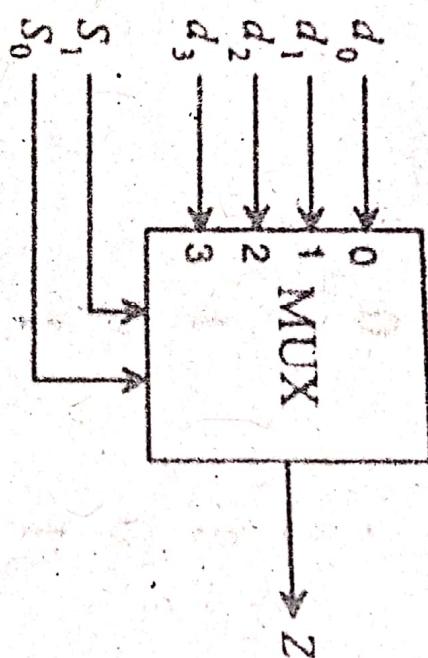


FIGURE 4.23 Block-diagram Representation of a Four-input Multiplexer

Combinational Logic Design

117

TABLE 4.24 Truth Table of the 4-to-1 Input Multiplexer

S_1	S_0	Z
0	0	d_0
0	1	d_1
1	0	d_2
1	1	d_3

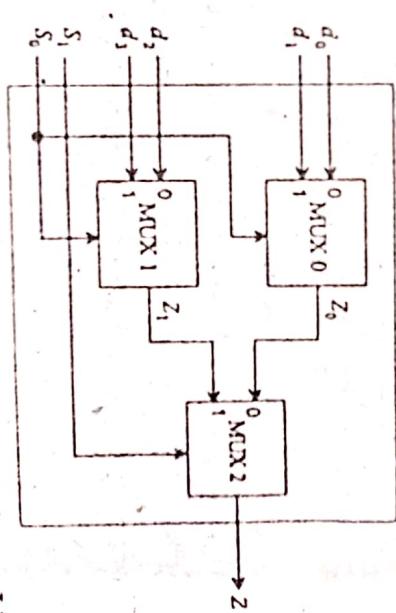
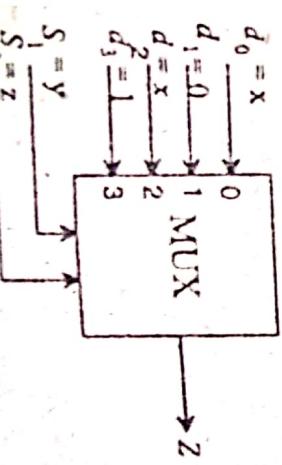


FIGURE 4.24 Implementation of a Four-Input Multiplexer Using Only Two-input Multiplexers



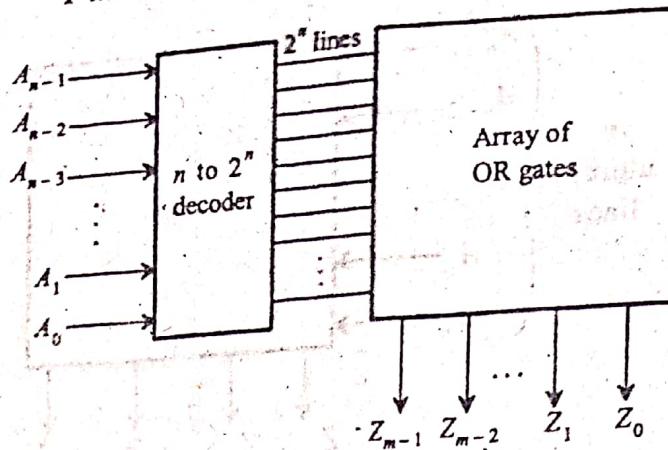


FIGURE 4.32 Internal Structure of a ROM

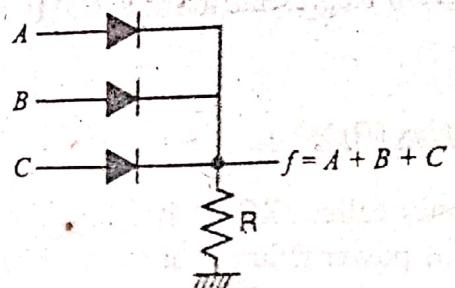


FIGURE 4.33 Diode-OR Gate

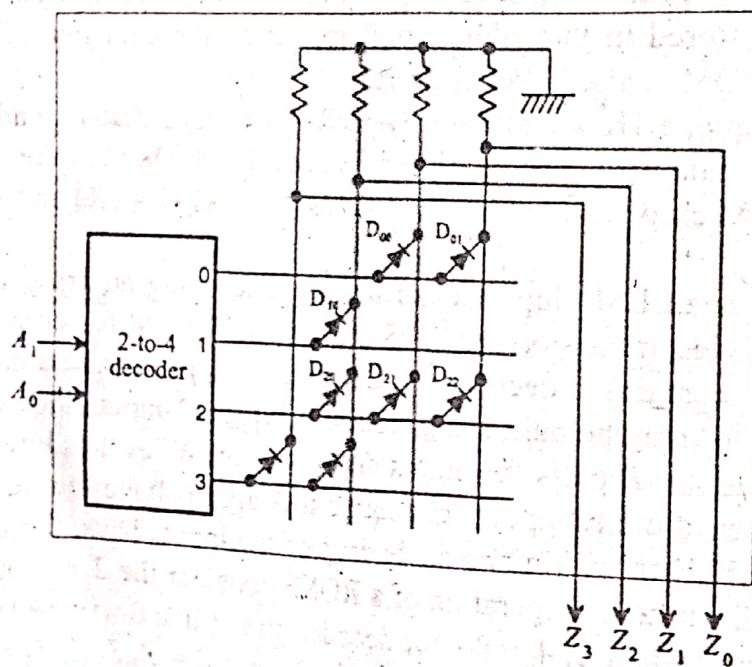


FIGURE 4.34 Hardware Organization of a Typical 2×4 ROM

TABLE 4.14 Truth Table implemented by the ROM of Figure 4.34

A_1	A_0	Z_3	Z_2	Z_1	Z_0
0	0	0	0	1	1
0	1	0	1	0	0
1	0	0	1	1	1
1	1	1	1	0	0

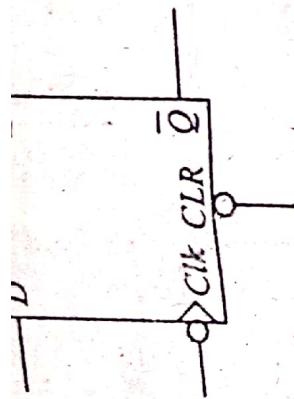


FIGURE 5.9 D Flip-Flop with Clear Input

S	Q	R	Q^+	R
Clk				
0	0	0	Q	Unchanged
0	1	0	0	Reset
1	0	1	1	Set
1	1	?	?	Invalid

(a) Symbolic Representation (b) Characteristic Table

FIGURE 5.10 RS flip-flop

J	K	Q^+	J	K
Clk				
0	0	Q	0	0
0	1	0	0	1
1	0	1	1	0
1	1	\bar{Q}	1	1

(a) Symbolic Representation (b) Characteristic Table

FIGURE 5.11 JK flip-flop

D	Q^+	D	D
Clk			
0	0	Reset	0
1	1	Set	1
1	1	Complement	0

(a) Symbolic Representation (b) Characteristic Table

FIGURE 5.12 D flip-flop

T	Q^+	T	T
Clk			
0	0	Unchanged	0
1	\bar{Q}	Complement	1
1	1	0	1
1	1	1	0

(a) Symbolic Representation (b) Characteristic Table

FIGURE 5.13 T flip-flop

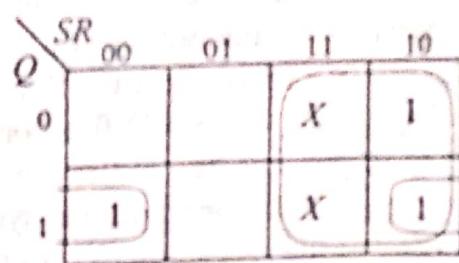
T	Q^+	T	T
Clk			
0	0	Unchanged	0
1	\bar{Q}	Complement	1
1	1	0	1
1	1	1	0

(a) Symbolic Representation (b) Characteristic Table

FIGURE 5.14 T flip-flop

(c) Excitation Table

Q	S	R	Q^+
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	invalid
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	invalid

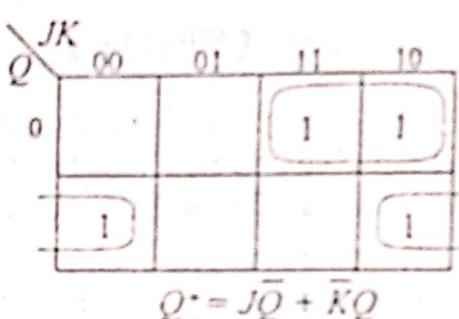


(a) Truth Table for RS-FF

(b) K-map for characteristic equation of RS-FF

FIGURE 5.14 Truth table and K-map for the characteristic equation of RS flip-flop

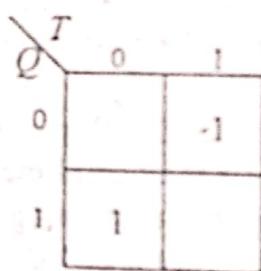
Q	J	K	Q^+
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



(a) Truth Table for JK-FF

(b) K-map for characteristic equation of JK-FF

Q	T	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

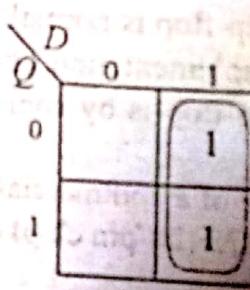


(c) Truth Table for T-FF

(d) K-map for characteristic equation of T-FF

FIGURE 5.15 Truth table and K-map for the characteristic equation of JK and T flip-flops

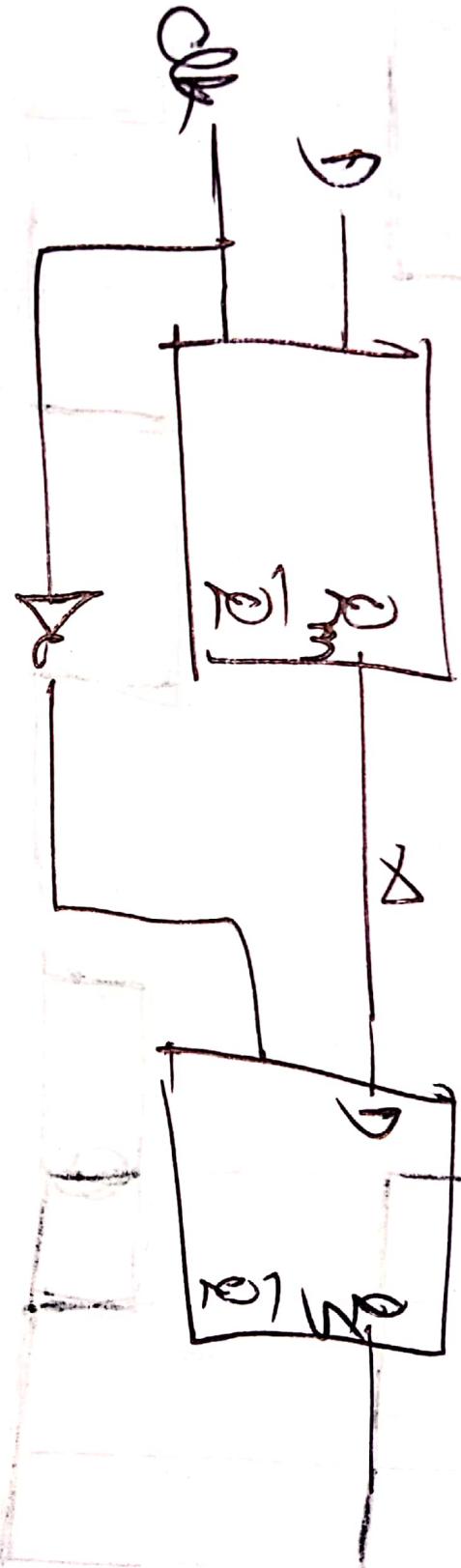
Q	D	Q^+
0	0	0
0	1	1
1	0	0
1	1	1



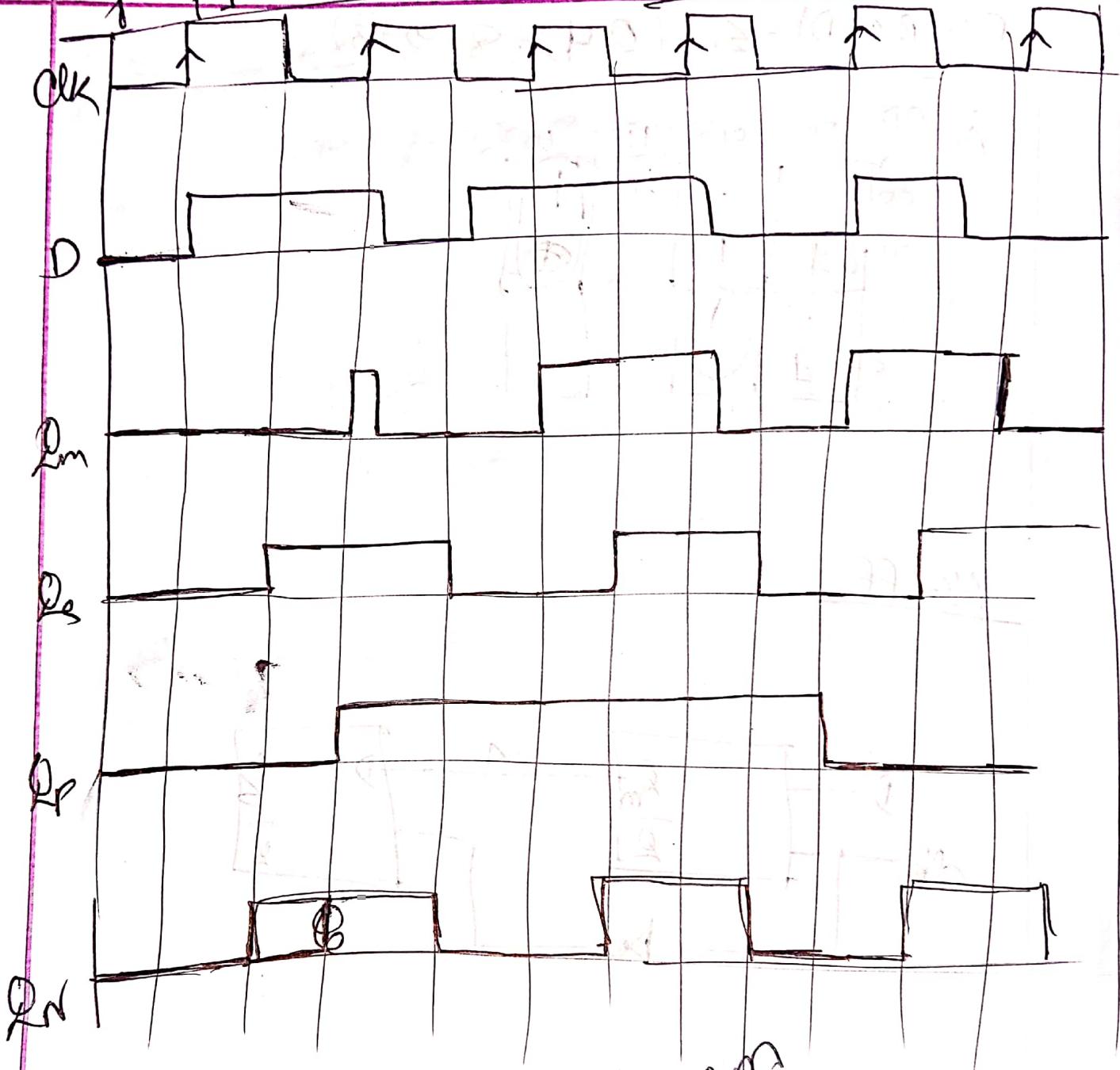
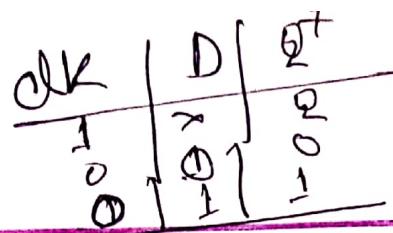
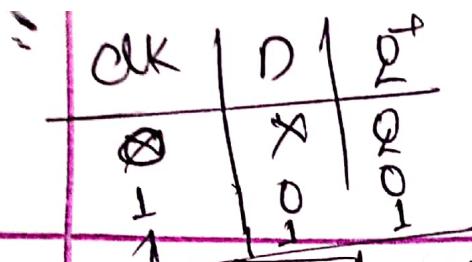
(a) Truth Table for D-FF

(b) K-map for characteristic equation of D-FF

FIGURE 5.16 Truth table and K-map for the characteristic equation of D flip-flop



M
ff

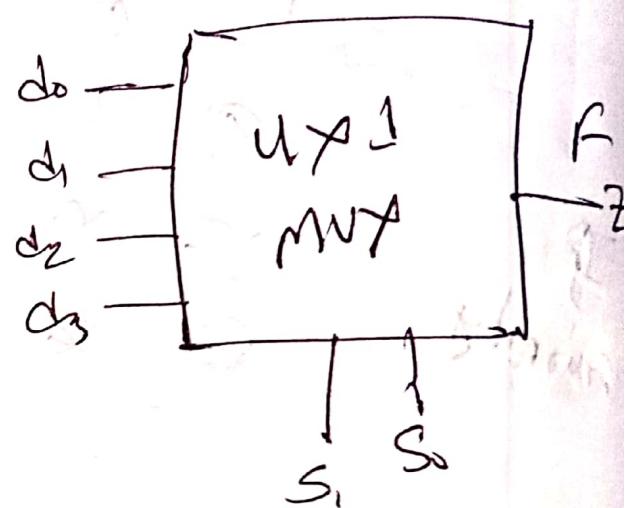


Timing Diagram

MUX Design

$$F = \Sigma m(1, 4, 5, 7, 9, 12, 13)$$

A_3	C_D	00	01	11	10	\bar{C}_D
0	1	1	1	1		$C + D$
1	1	1	1			\bar{C}
2	1	1				$\bar{C}D$
3		1				

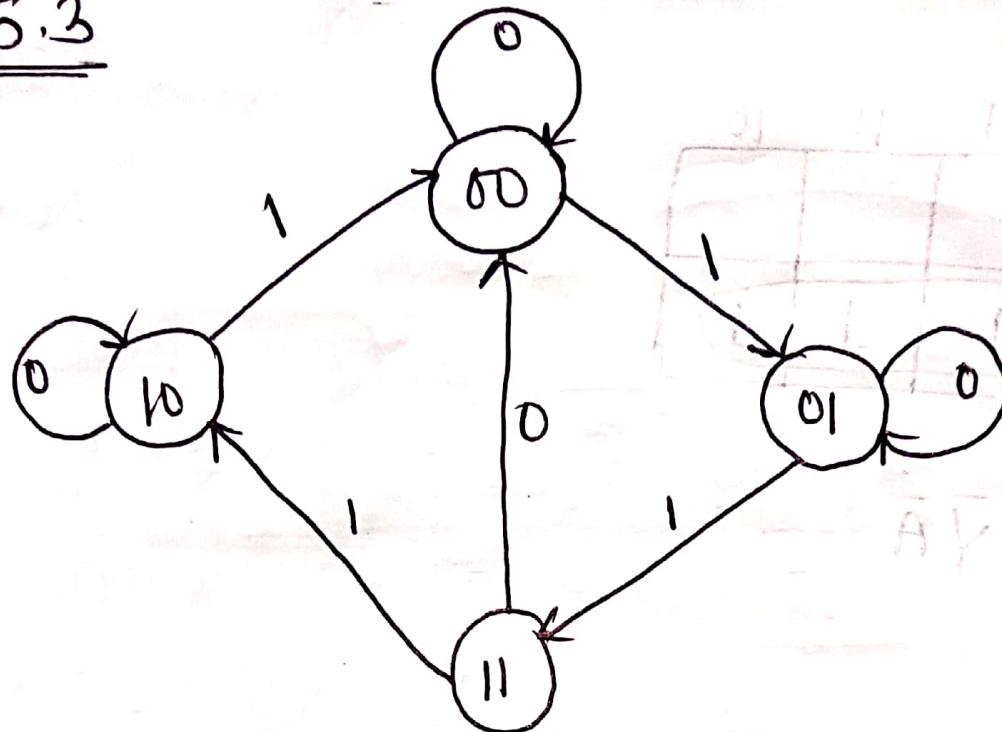


S_1	S_0	Z
0	0	$d_0 = \bar{C}D$
0	1	$d_1 = \bar{C} + D$
1	0	$d_2 = \bar{C}$
1	1	$d_3 = \bar{C}D$

$$S_0 = B$$

$$S_1 = A$$

Ex: 5.3



Design S.S.C using JK FF.

Excitation Table

Q	Q^+	J	K
0	0	0	x
0	1	1	x
1	0	x	1
0	1	x	0

Merged Diagram

Present States		Inputs	Next States		ff Inputs			
X	Y	A	X ⁺	Y ⁺	J _x	K _x	J _y	K _y
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	0	1	0	X	X	0
0	1	1	1	1	1	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	0	0	X	1	0	X
1	1	0	0	0	X	1	X	1
1	1	1	1	0	X	0	X	1

K-map(J_x)

X\YA	00	01	11	10
0			(1)	
1	X	X	(X)	X

$$J_x = YA$$

X\YA	00	01	11	10
0		(1)	X	X
1			X	X

$$J_y = \bar{X}A \quad \bar{X}A$$

X\YA	00	01	11	10
0	X	(X)	X	(X)
1		(1)		1

$$K_x = \bar{Y}A + Y\bar{A} = Y \oplus A$$

X\YA	00	01	11	10
0	X	X		
1	(X)	X	1	1

$$K_y = X$$

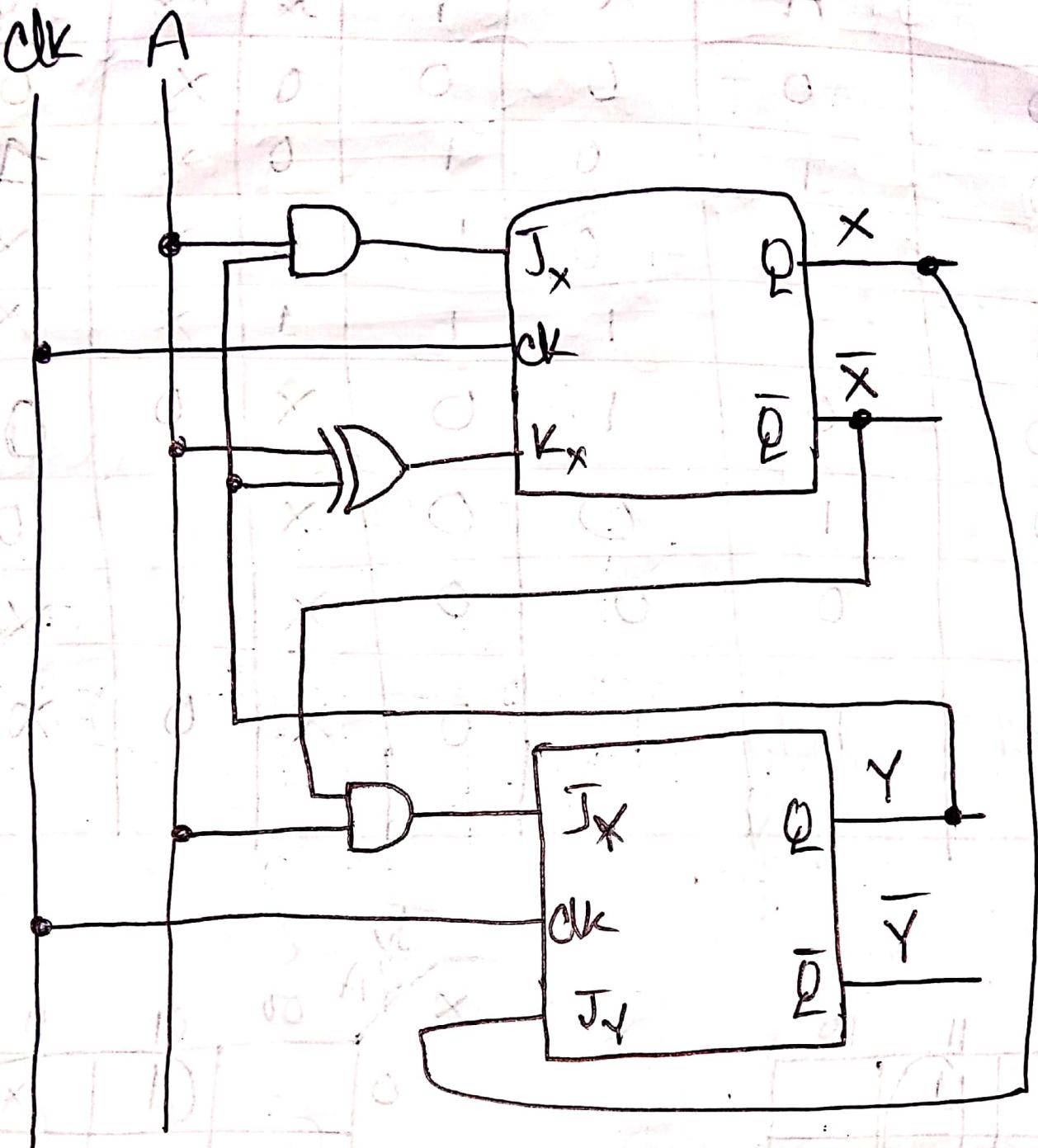
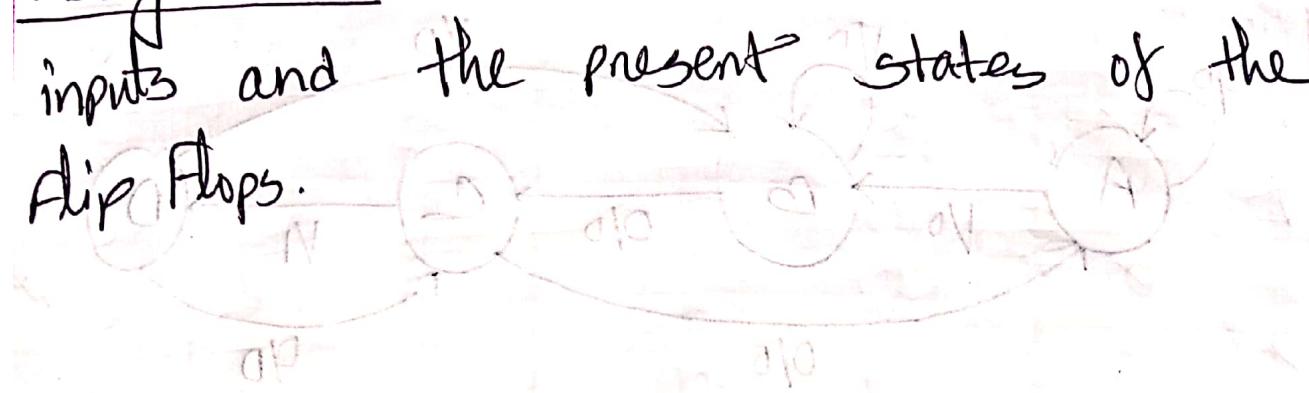


Fig: Logic Diagram.

$$A\bar{X} = \bar{X}C$$

Mealy Circuit: Output depends on both the inputs and the present states of the flip-flops.



Output depends on present state and input.

Moore Circuit: Output depends on present states of flip-flop.

Input should be bypassed
flip-flop only bypassed

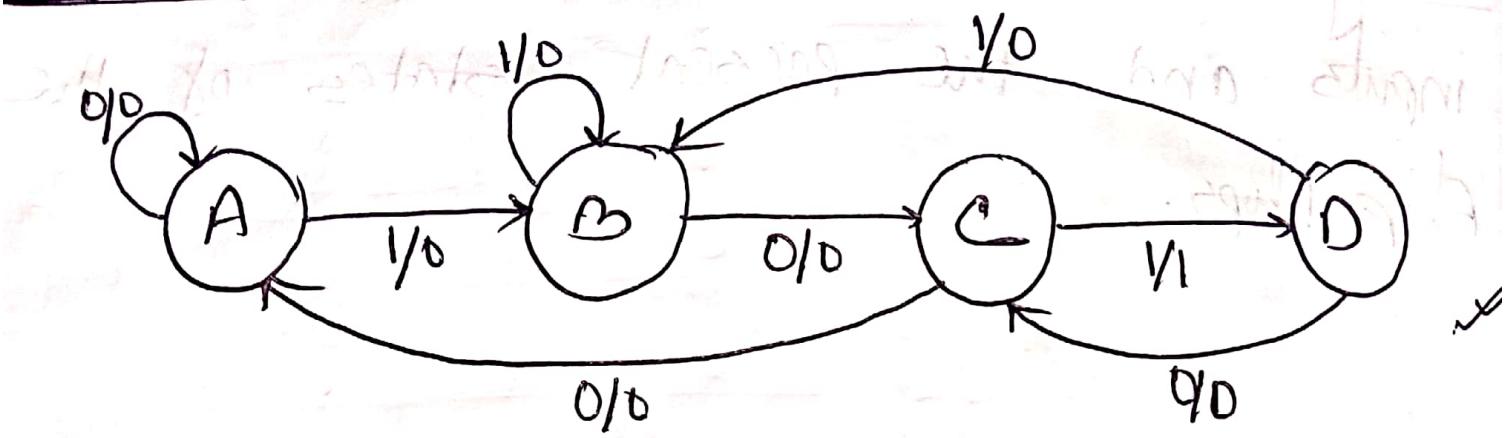
Output depends on state of flip-flop

Initial state

Input X1

Input X2

Ex: 5.4



Begin S.S.C using TFF.

Solution:

<u>State</u>	<u>Interpretation</u>
A	Look for a new pattern
B	Received the first 1
C	Received a 1 followed by a 0
D	Recognized the pattern 11

<u>Sybolic State</u>	<u>Binary Value</u>
A	0 0
B	0 1
C	1 1
D	1 0

State Table

Present State	Next State		Output Z	
	$x=0$	$x=1$	$x=0$	$x=1$
A	A	B	0	0
B	C	B	0	0
C	A	D	0	1
D	C	B	0	0

Modified State Table

Present State y_1 y_0	Next State		Output Z	
	$y_1 + y_0$	$y_1 + y_0$	Input	Input
	$x=0$	$x=1$	$x=0$	$x=1$
0 0	00	1 01	0	1 0
0 1	11	0 01	0	1 0
1 1	00	0 10	0	1 0
1 0	11	1 01	0	1 0

Excitation Table

<u>Q</u>	<u>Q⁺</u>	<u>T</u>
0	0	0
0	1	0
1	0	1
0	1	0

Present State		Input	Next State		FF Input	Output
y_1	y_0	x	y_1^+	y_0^+	\bar{T}_{y_1}	\bar{T}_{y_0}
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	1	1	0
0	1	1	0	1	0	0
1	0	0	1	0	0	0
1	0	1	0	1	1	0
1	1	0	0	0	1	0
1	1	1	0	1	0	1

K-Map

	\bar{Y}_1	$\bar{Y}_0 X$	00	01	11	10
\bar{Y}_1	0	1	1	1	1	1
1	1	1	1	1	1	1

\bar{T}_{Y_0}

	\bar{Y}_1	$\bar{Y}_0 X$	00	01	11	10
\bar{Y}_1	0	1	1	1	1	1
1	1	1	1	1	1	1

$$\bar{T}_{Y_1} = \bar{Y}_1 \bar{Y}_0 X + Y_0 \bar{X}$$

$$\bar{T}_{Y_0} = Y_1 + \bar{Y}_0 X$$

\bar{Z}

	\bar{Y}_1	$\bar{Y}_0 X$	00	01	11	10
\bar{Y}_1	0	1	1	1	1	1
1	1	1	1	1	1	1

$$Z = Y_1 Y_0 X$$

maximal sign : 9

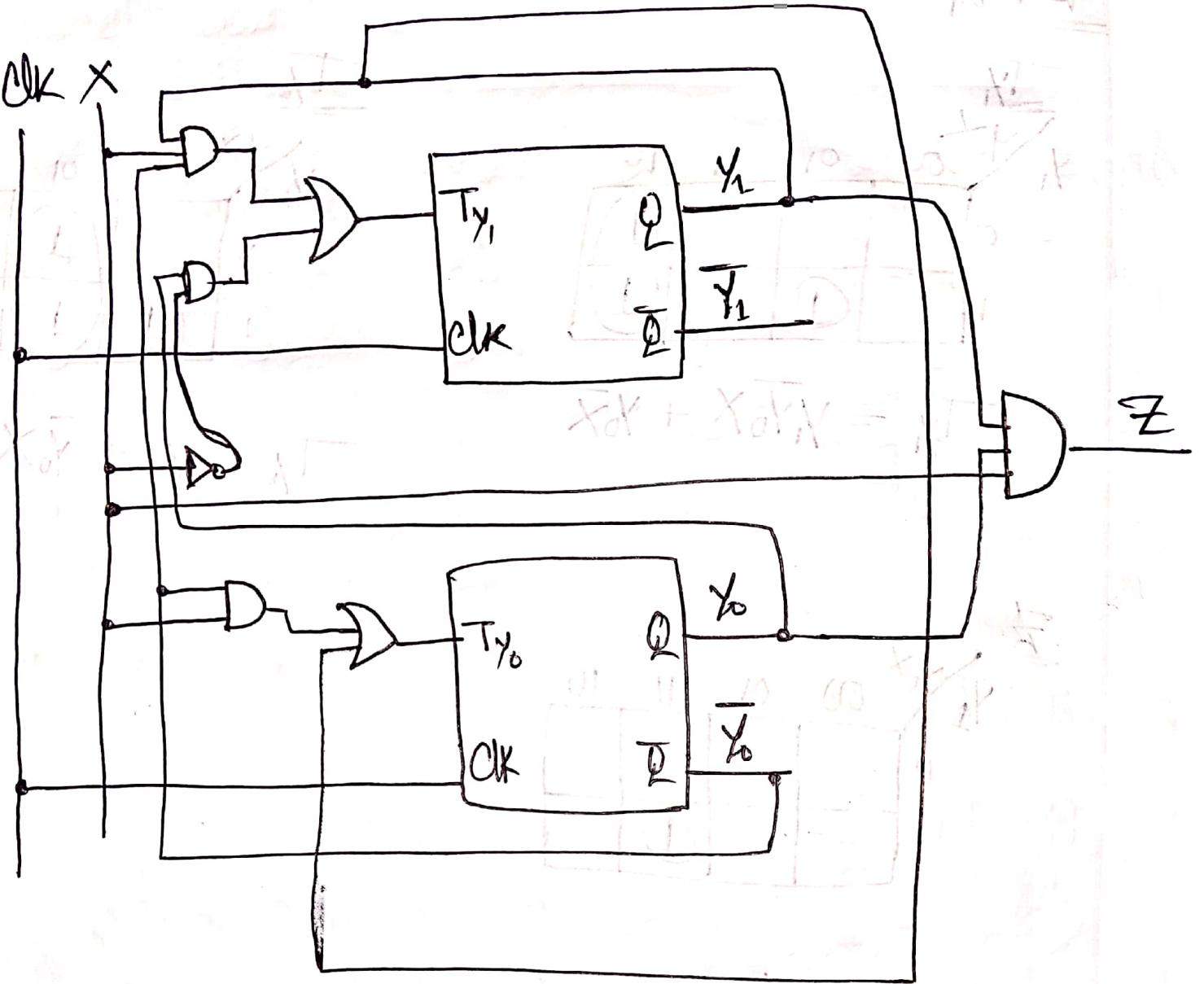
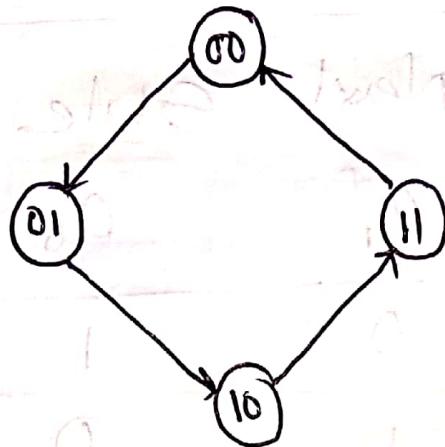


Fig. Logic Diagram

Ex: 5.5



Design 2bit Counter using T FF.

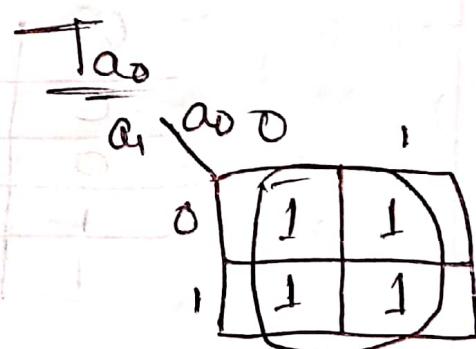
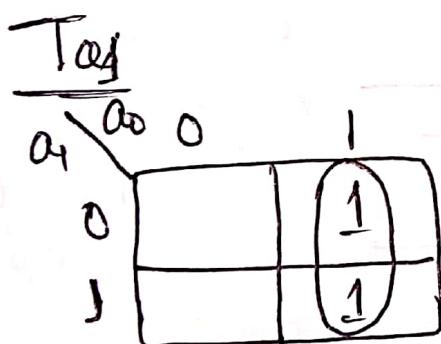
Excitation Table

Q	Q^+	T
0	0	0
0	1	1
1	0	1

State Table

Present State		Next State		FF Inputs	
a_1	a_0	a_1^+	a_0^+	T_{a_1}	T_{a_0}
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

K-map



$$T_{a_1} = a_0$$

$$T_{a_0} = 1$$

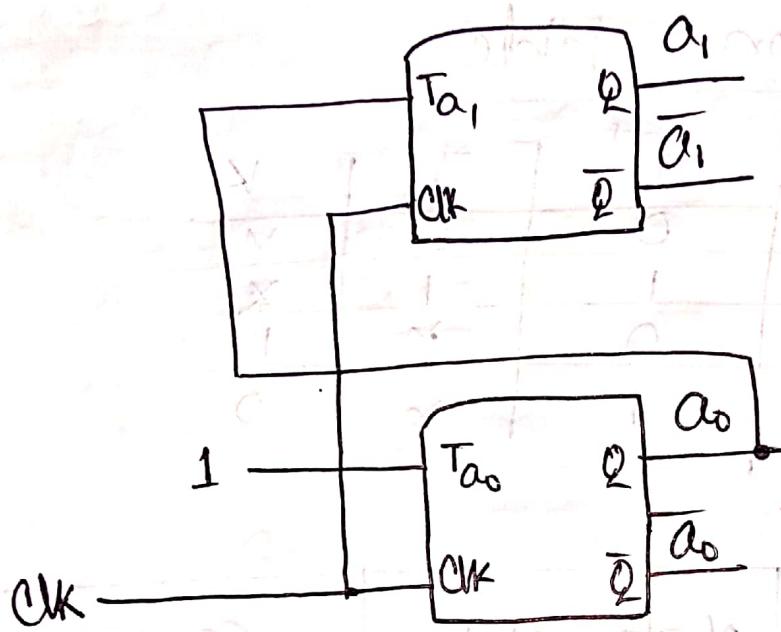
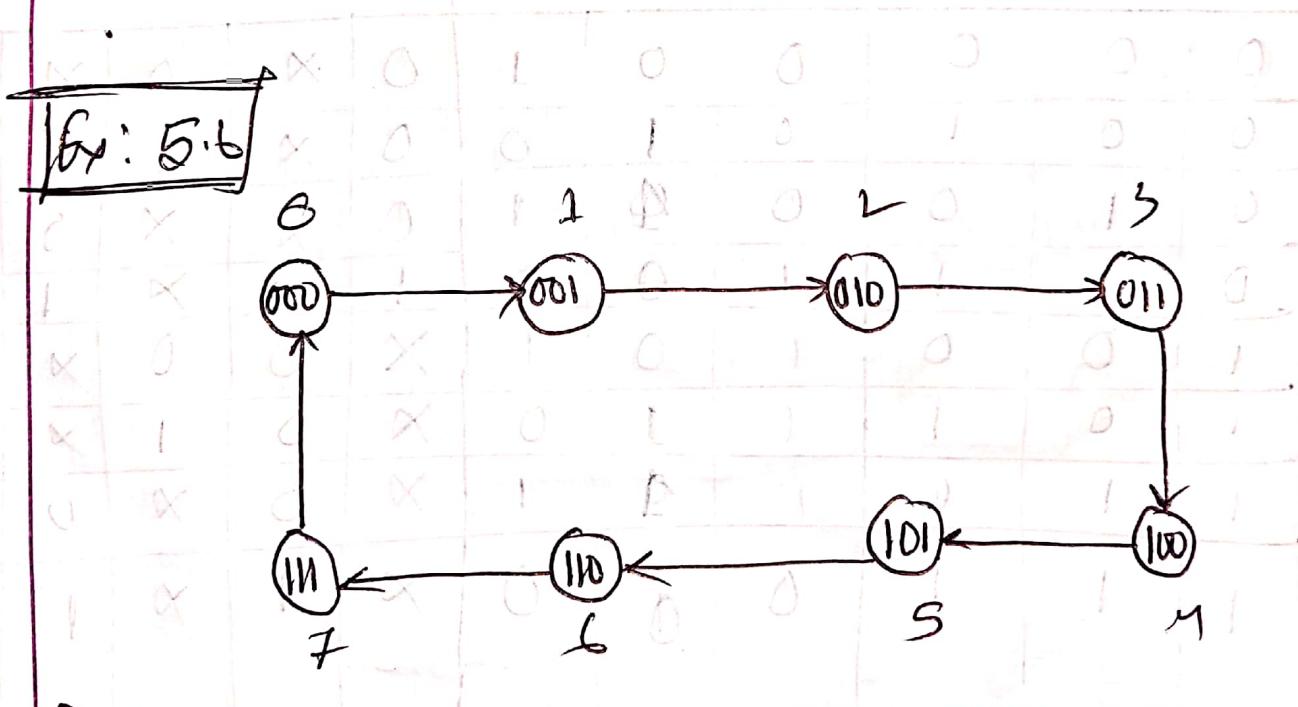


Fig. Logic Diagram for 2-bit Counter.



Design 3 bit counters using JK FF.

Soln:

Excitation Table

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

Present States			Next States			FF Input		
Q_2	Q_1	Q_0	Q_2'	Q_1'	Q_0'	J_{Q_2}	K_{Q_2}	J_{Q_1} , J_{Q_0} , K_{Q_0}
0	0	0	0	0	1	0	X	0 X 1 X
0	0	1	0	1	0	0	X	1 X X 1
0	1	0	0	1	1	0	X	X 0 1 X
0	1	1	1	0	0	1	X	1 X X 1
1	0	0	1	0	1	X	0	0 X 1 X
1	0	1	1	1	0	X	0	1 X X 1
1	1	0	1	1	1	X	0	X 0 1 X
1	1	1	0	0	0	X	1	1 X X 1

K-Map

		J ₀₂			
		00	01	11	10
a ₂	0	X		1	
	1	X	X	X	X

		K ₀₂			
		00	01	11	10
a ₂	0	X	X	1	X
	1			1	

$$J_{02} = a_1 a_0$$

$$K_{02} = a_1 a_0$$

J₀₁

		J ₀₁			
		00	01	11	10
a ₂	0	1	X	X	X
	1	1	X	X	X

		K ₀₁			
		00	01	11	10
a ₂	0	X	X	1	
	1	X	X	1	

$$J_{01} = a_0$$

$$K_{01} = a_0$$

J₀₀

		J ₀₀			
		00	01	11	10
a ₂	0	1	X	X	1
	1	1	X	X	1

		K ₀₀			
		00	01	11	10
a ₂	0	X	1	1	X
	1	X	1	1	X

$$J_{00} = 1$$

$$K_{00} = 1$$

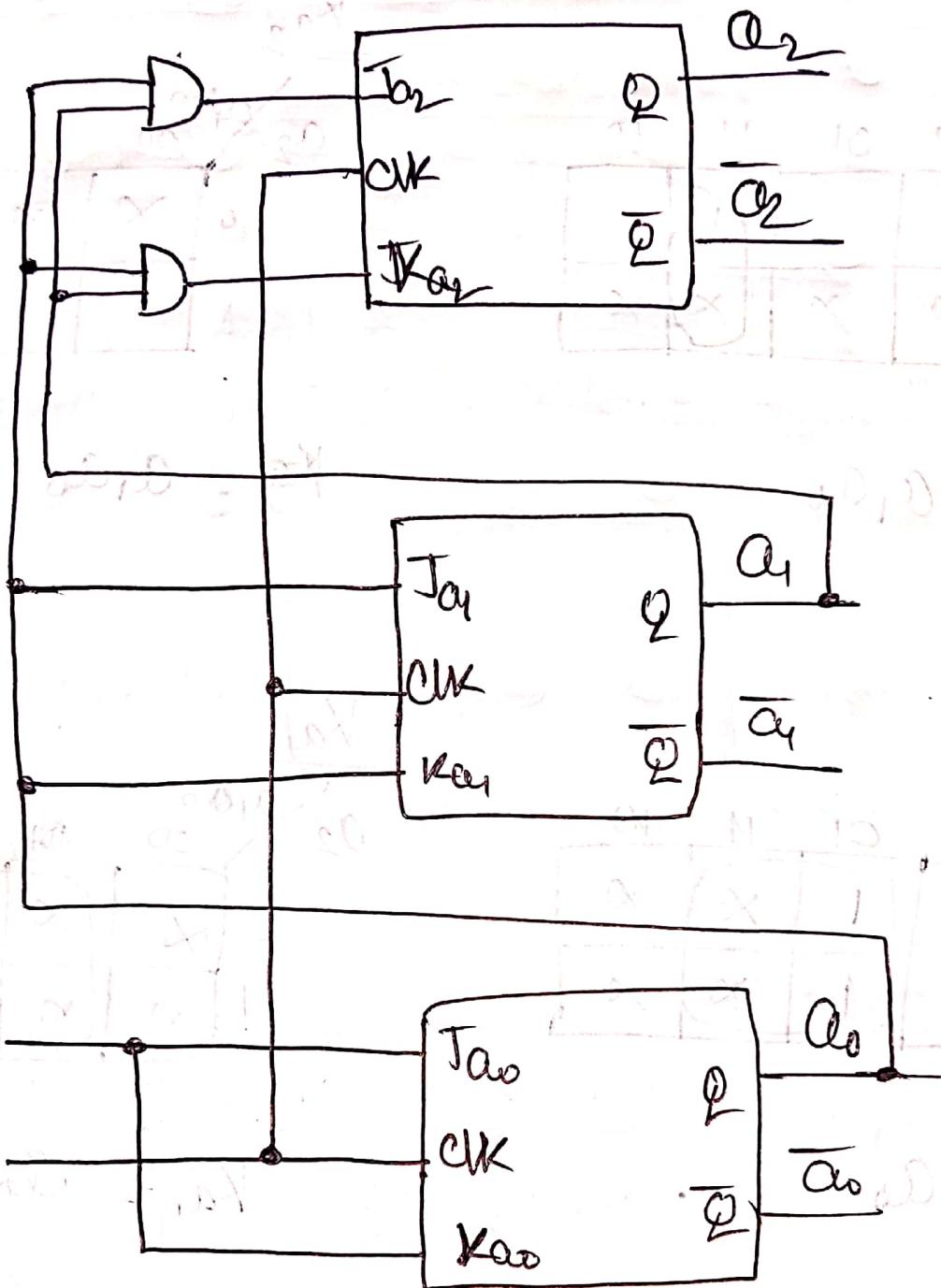
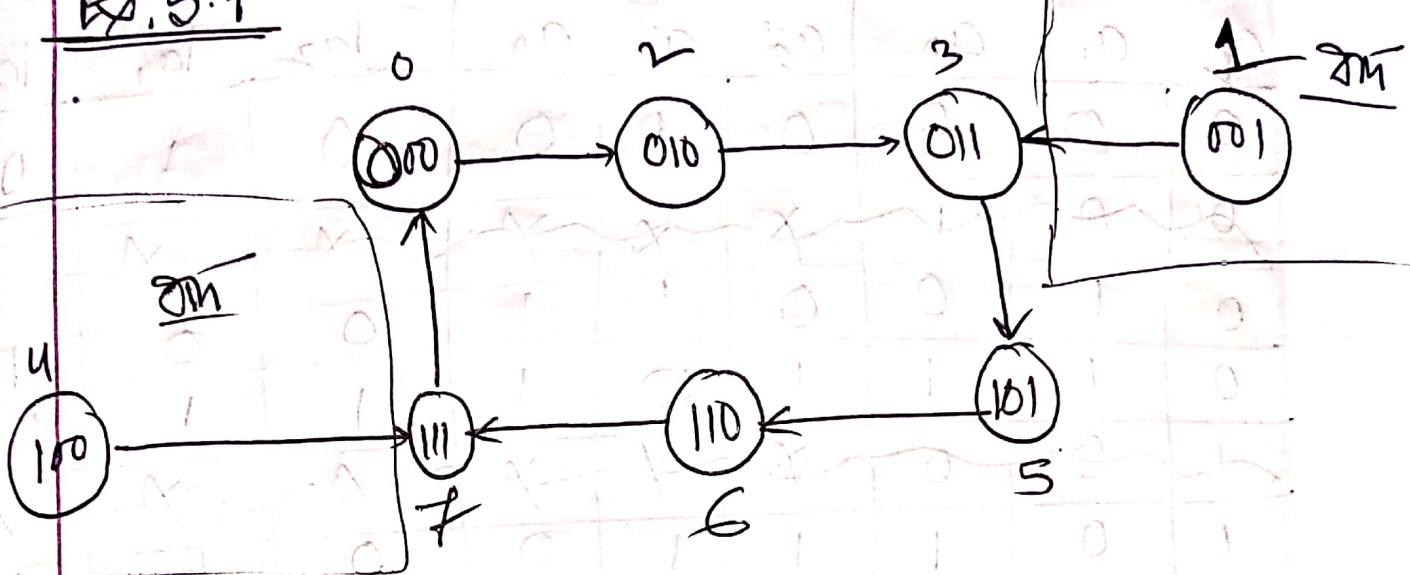


fig. 3 Bit Counter

1	x	x	1	0
1	x	x	1	1

Ex: 5.7



Design 3 bit counters using T FF.
(few conditions)

Soln:

Excitation table

α	11	10	0	Q	Q^+	T	\bar{T}	X	0
	1	0	1	0	0	0	1	1	1
	0	1	0	0	1	1	0	0	0
	1	0	0	1	0	1	0	0	0

$$D + \bar{Q} = \text{set}$$

α	11	10	0	Q	Q^+	T	\bar{T}	X	0
	1	0	1	0	0	0	1	1	1
	0	1	0	0	1	1	0	0	0
	1	0	0	1	0	1	0	0	0

$$\bar{D}_P + Q = \text{set}$$

Present State			Next State			FF Inputs		
a_2	a_1	a_0	a_2^+	a_1^+	a_0^+	T_{a2}	T_{a1}	T_{a0}
0	0	0	0	1	0	0	1	0
0	0	0	X	X	X	X	X	X
0	1	0	0	1	1	0	0	1
0	1	1	1	0	1	1	1	0
1	0	0	X	X	X	X	X	X
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

K-Map

T_{a2}

a_2	a_1	a_0	00	01	11	10
0	0	X	1			
1	X		1			

T_{a1}

a_2	a_1	a_0	00	01	11	10
0	1	X	1			
1	X		1	1	1	1

$$T_{a2} = a_1 a_0$$

$$T_{a1} = \bar{a}_1 + a_0$$

T_{a0}

a_2	a_1	a_0	00	01	11	10
0	0	X				
1	X	1	1	1	1	1

$$T_{a0} = a_2 + a_1 \bar{a}_0$$

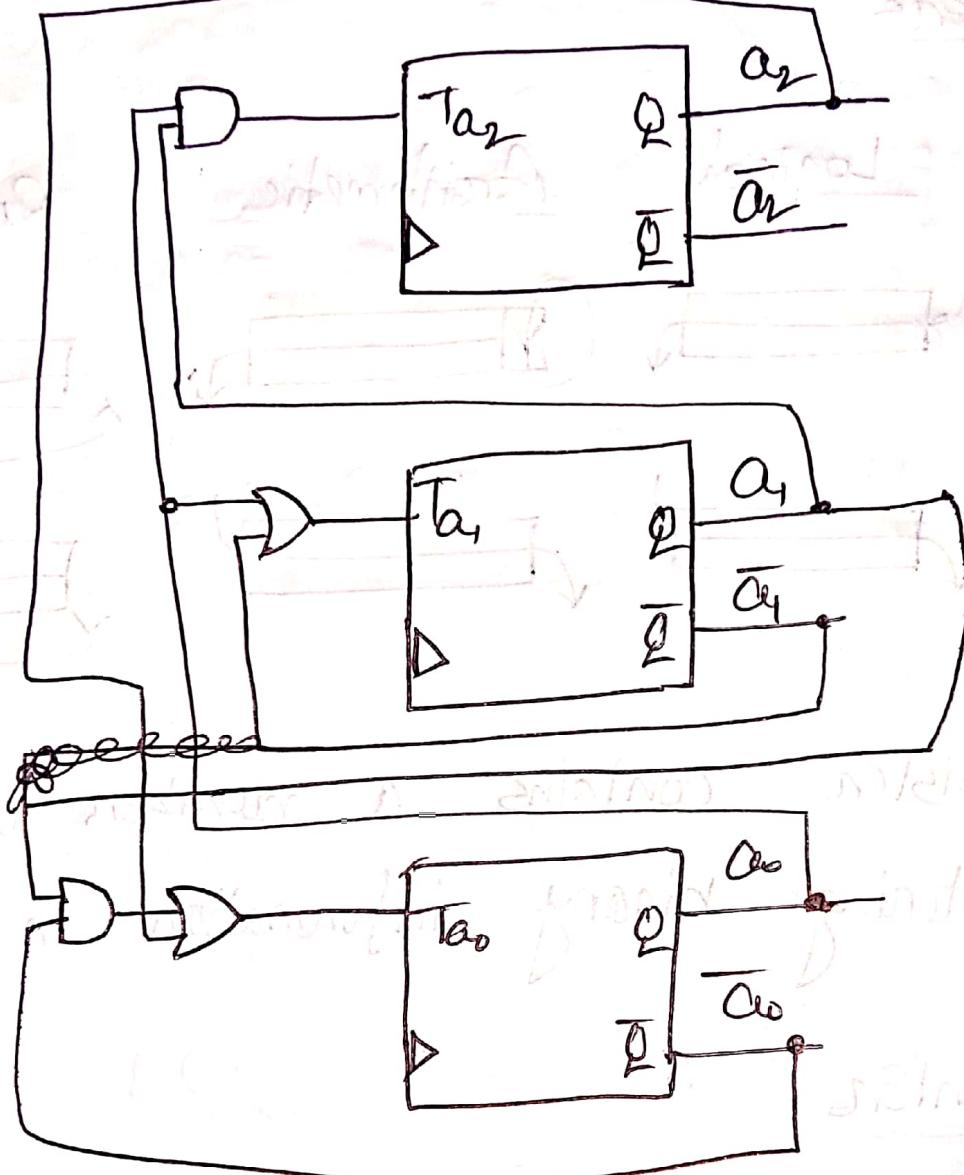


fig: Logic Diagram.

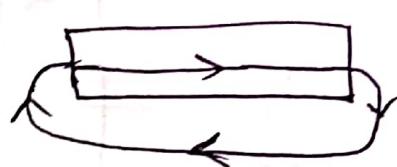
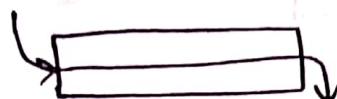
Registers

Logical

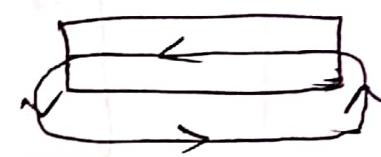
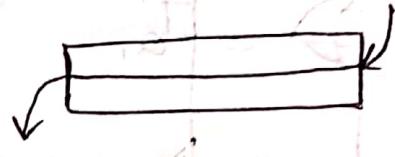
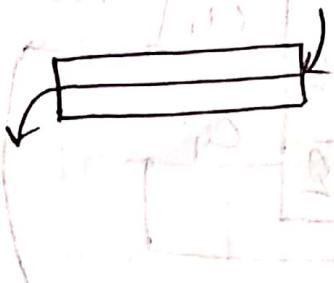
Arithmetic

Rotate

Right



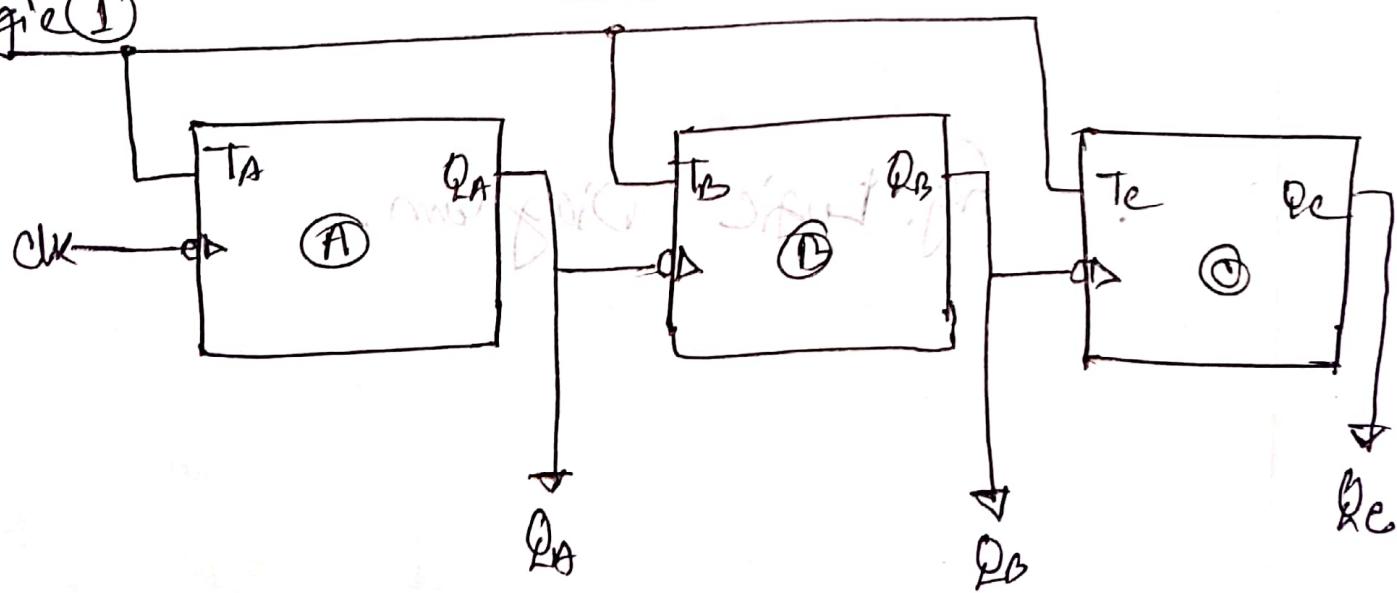
Left



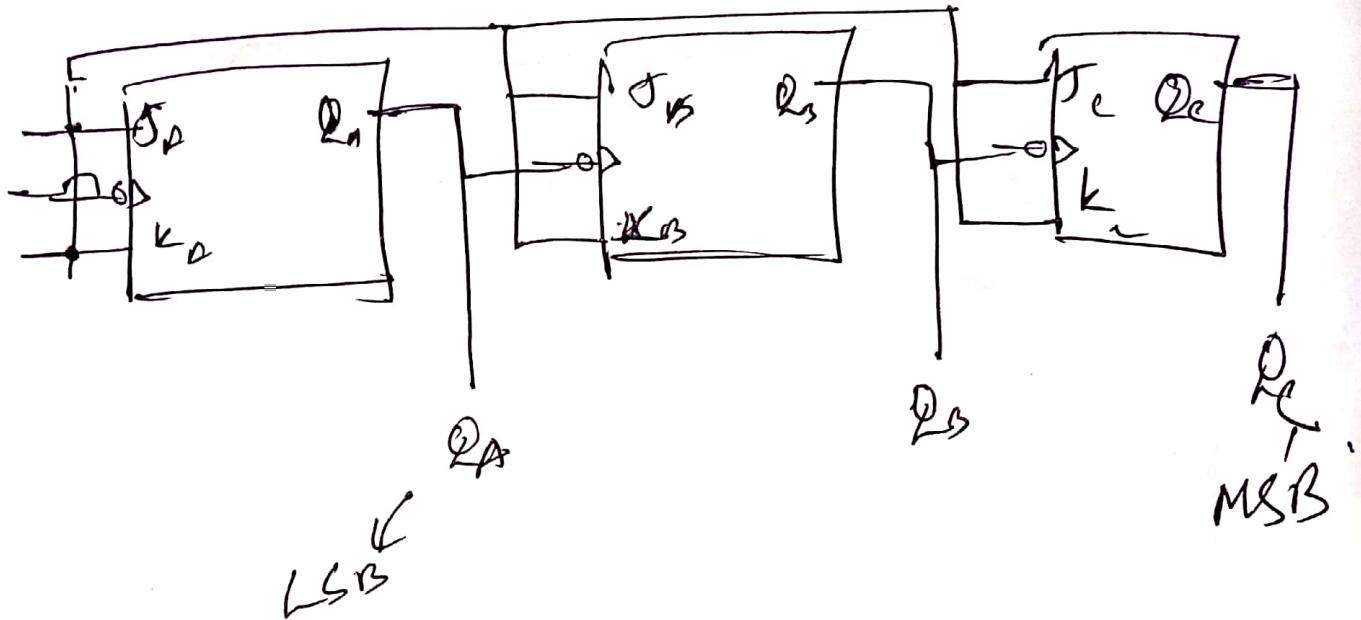
A register contains a number of flip-flop for storing binary information in a computer.

Up Counter

Logic ①



UP Counter:



Q₃ Q₂ Q₁

000

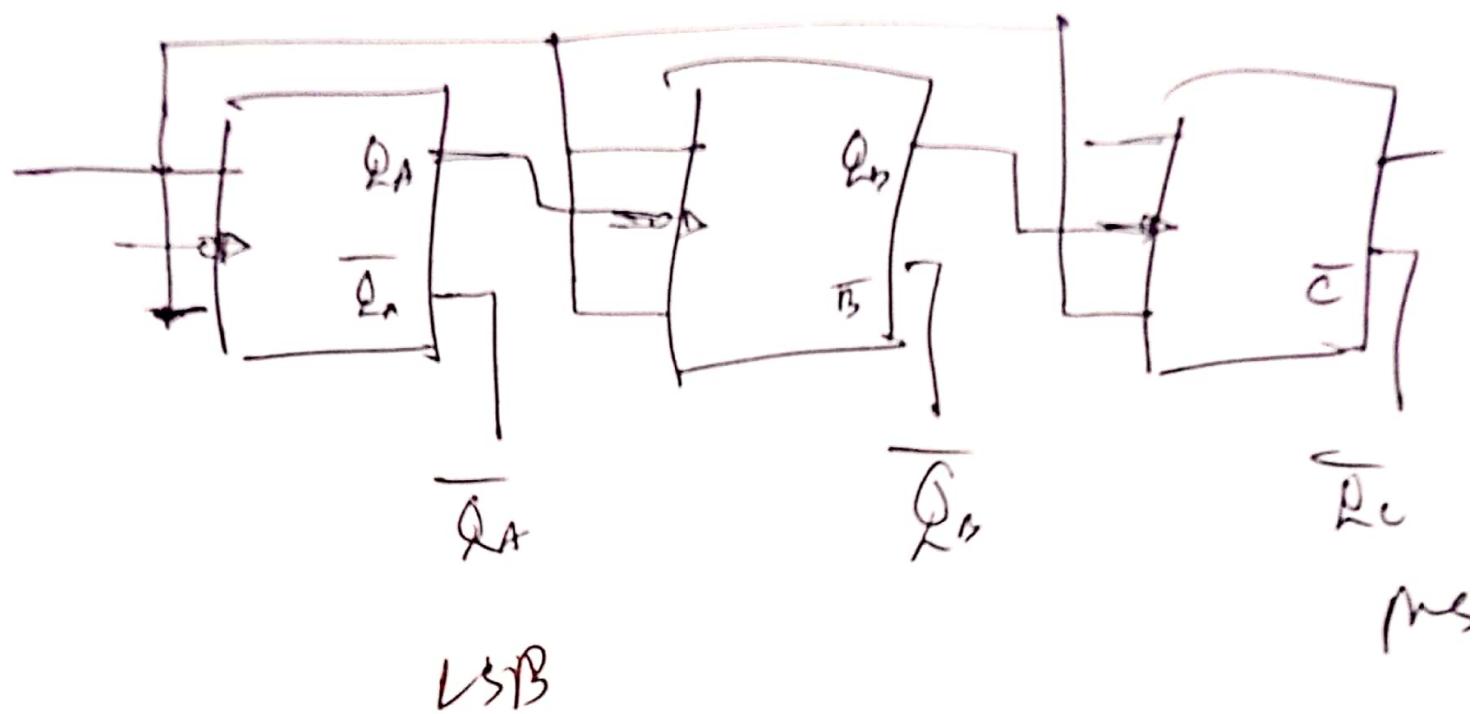
001

...

...

111

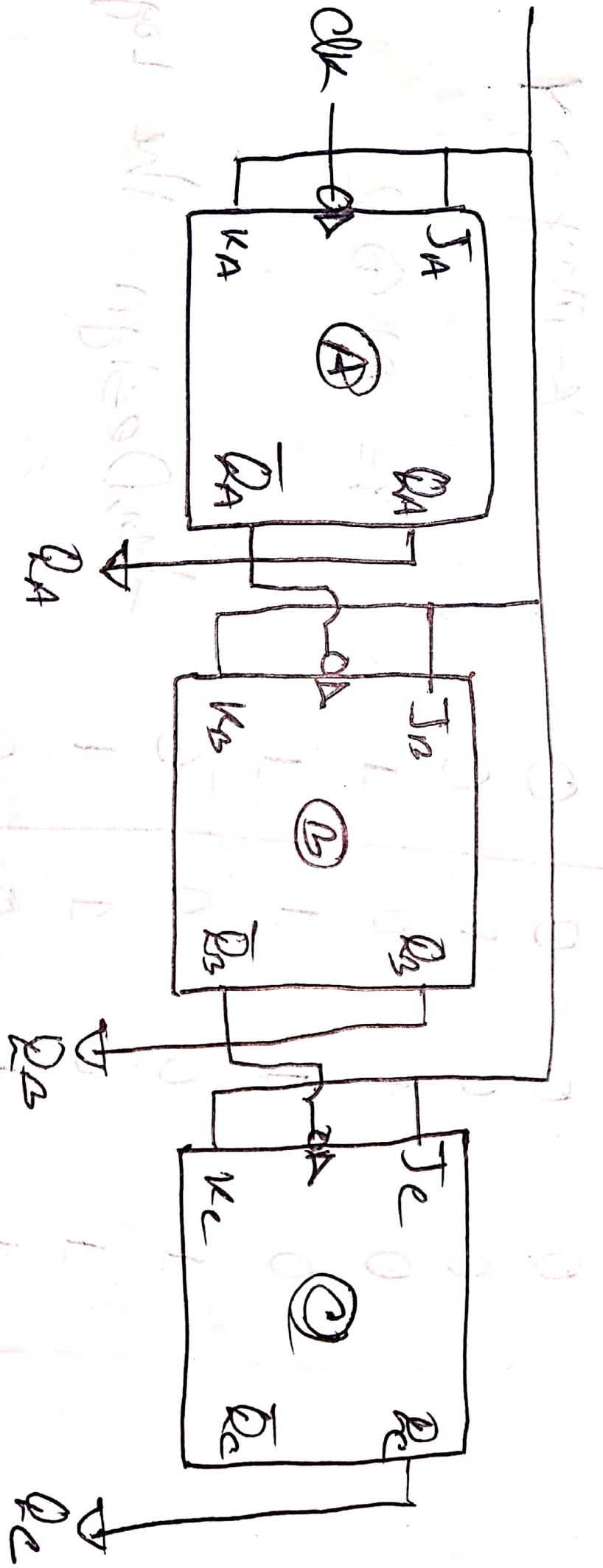
Down



LSPB

clk	Q_C	\bar{Q}_C	Q_B	\bar{Q}_B	Q_A	\bar{Q}_A
Initial	0	1	0	1	0	1
1st	0	1	1	0	0	1
2nd	0	1	0	1	1	0
3rd	0	1	1	0	0	1
4th	-	-	-	-	-	-

Down Counter



Up/Down Ripple Counter

M	P	\bar{P}	Y
0	0	0	0
0	0	0	0
0	-	-	0
-	0	-	0
-	0	0	1
-	0	0	0

Up/Down selection = sign bit where \bar{P} is connected to \bar{Q}_N

$$M =$$

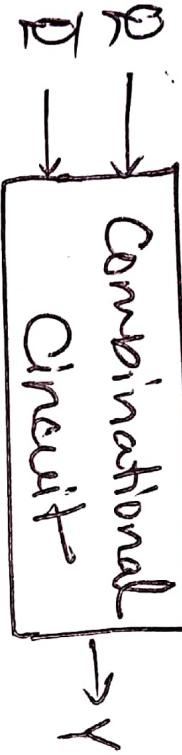
$$M = 0$$

Up counting

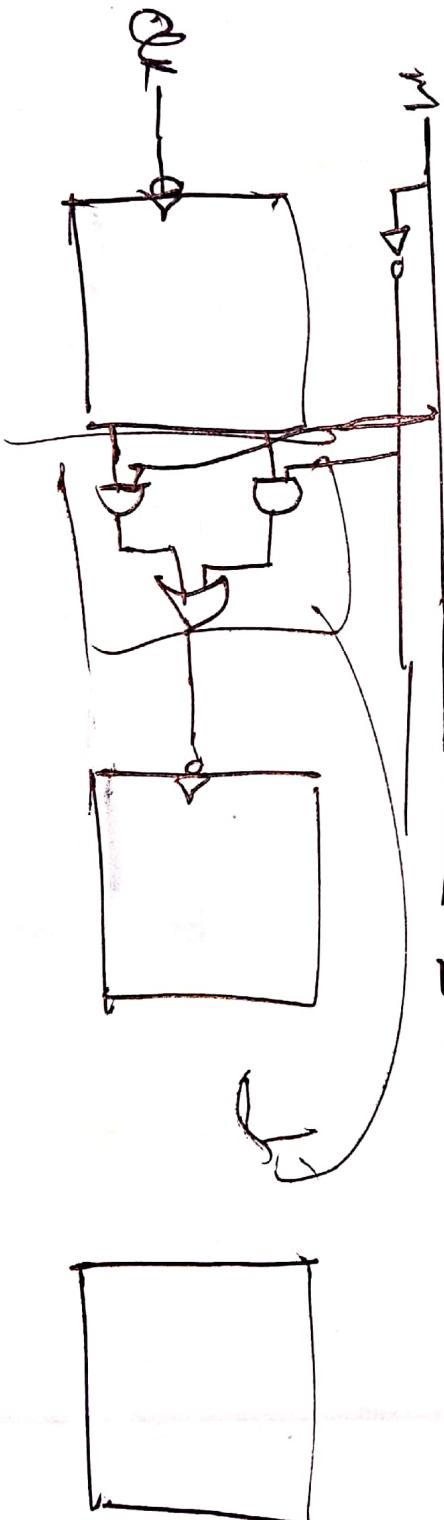
$$M = 1$$

down in

\bar{Q} is connected to \bar{Q}_N .



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



2bit ripple counters = modern counters

	m	Q	\bar{Q}	
	0	0	0	Y
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

$$R\text{-Max} \rightarrow Y$$

$$Y = m \oplus Q$$

Now Design the logic circuit

MOD-6 Counter

and Like BCD Counter

States = 6

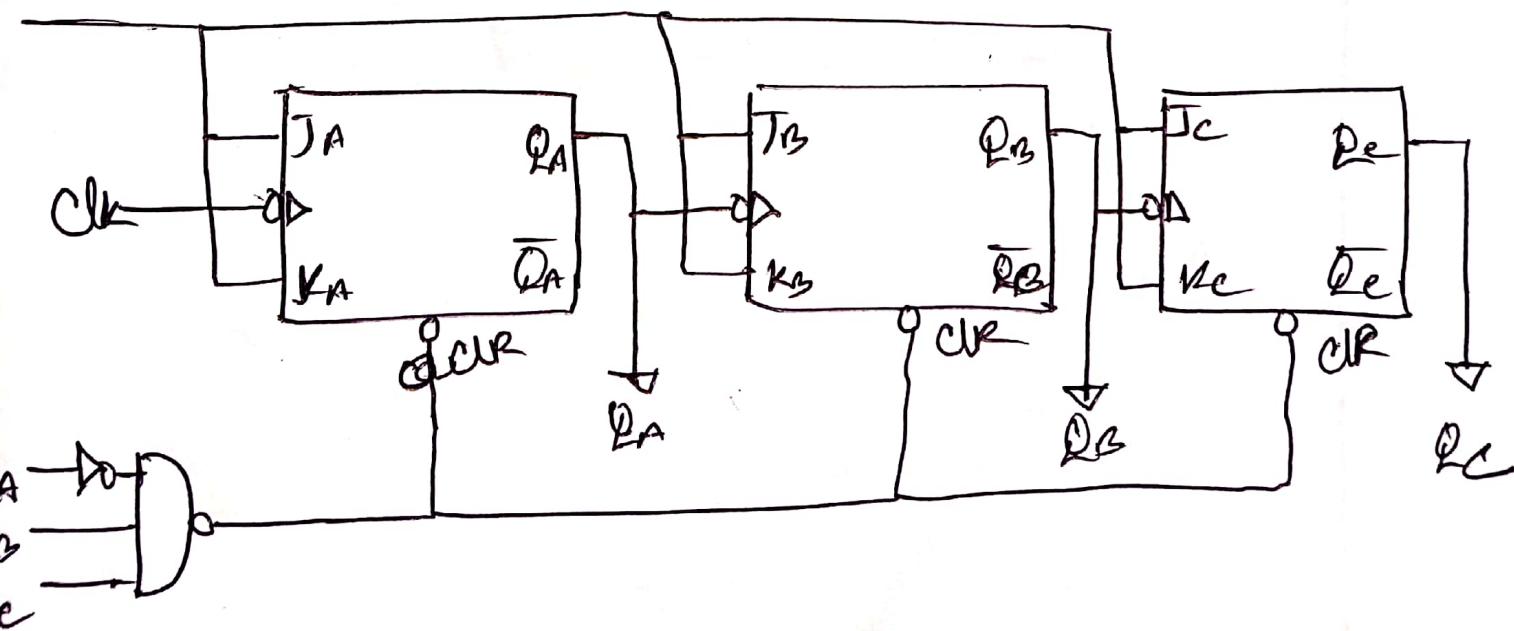
Max. count = $6 - 1$

= 5

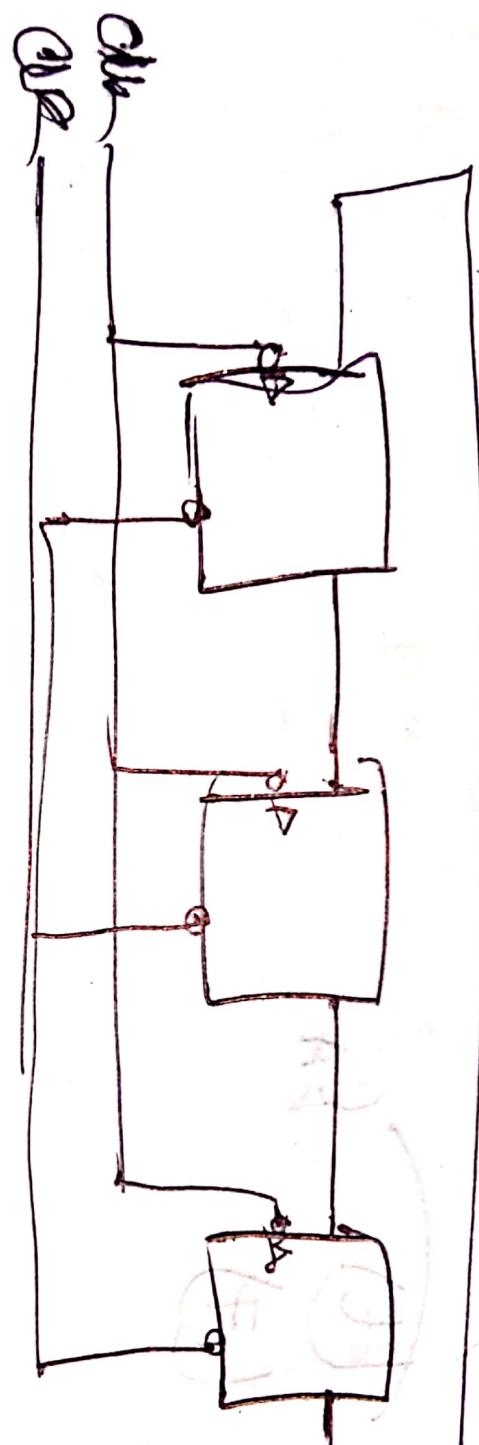
$Q_C \ Q_B \ Q_A$

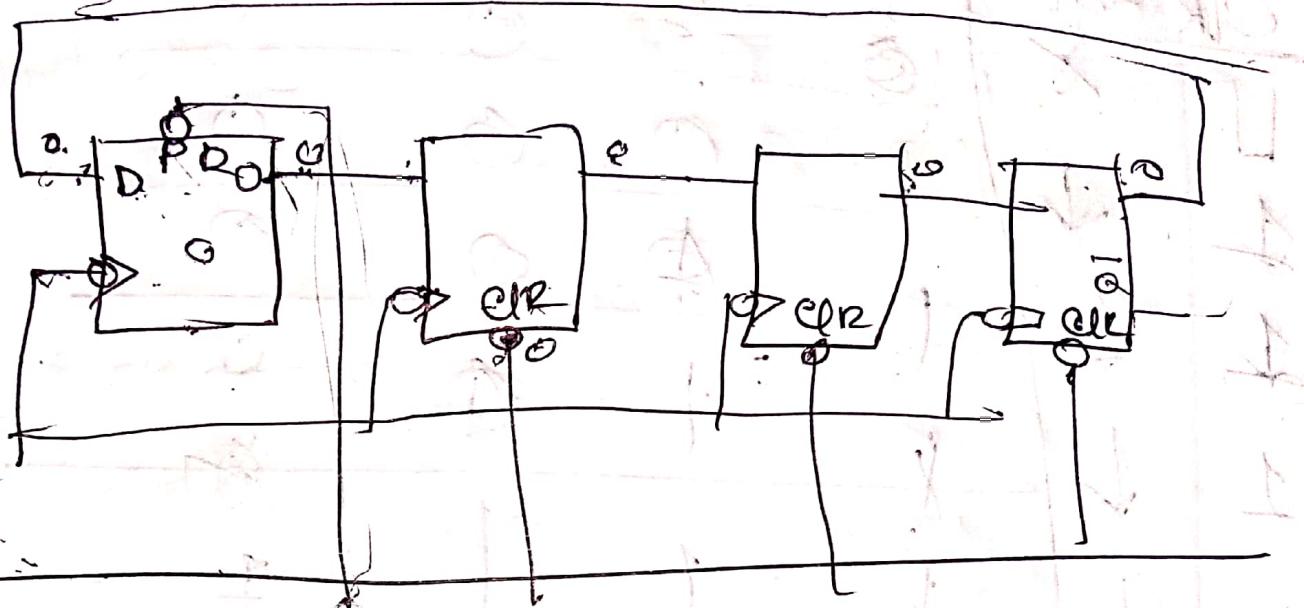
- (0) 000 } MOD-6
- (1) 001 }
- (2) 010 }
- (3) 011 }
- (4) 100 }
- (5) 101 }

$Q_C \leftarrow 1 \ 1 \ 0$ (6) Q_A
 $Q_B \leftarrow 1 \ 1 \ 1$ (7)



Ring Counter



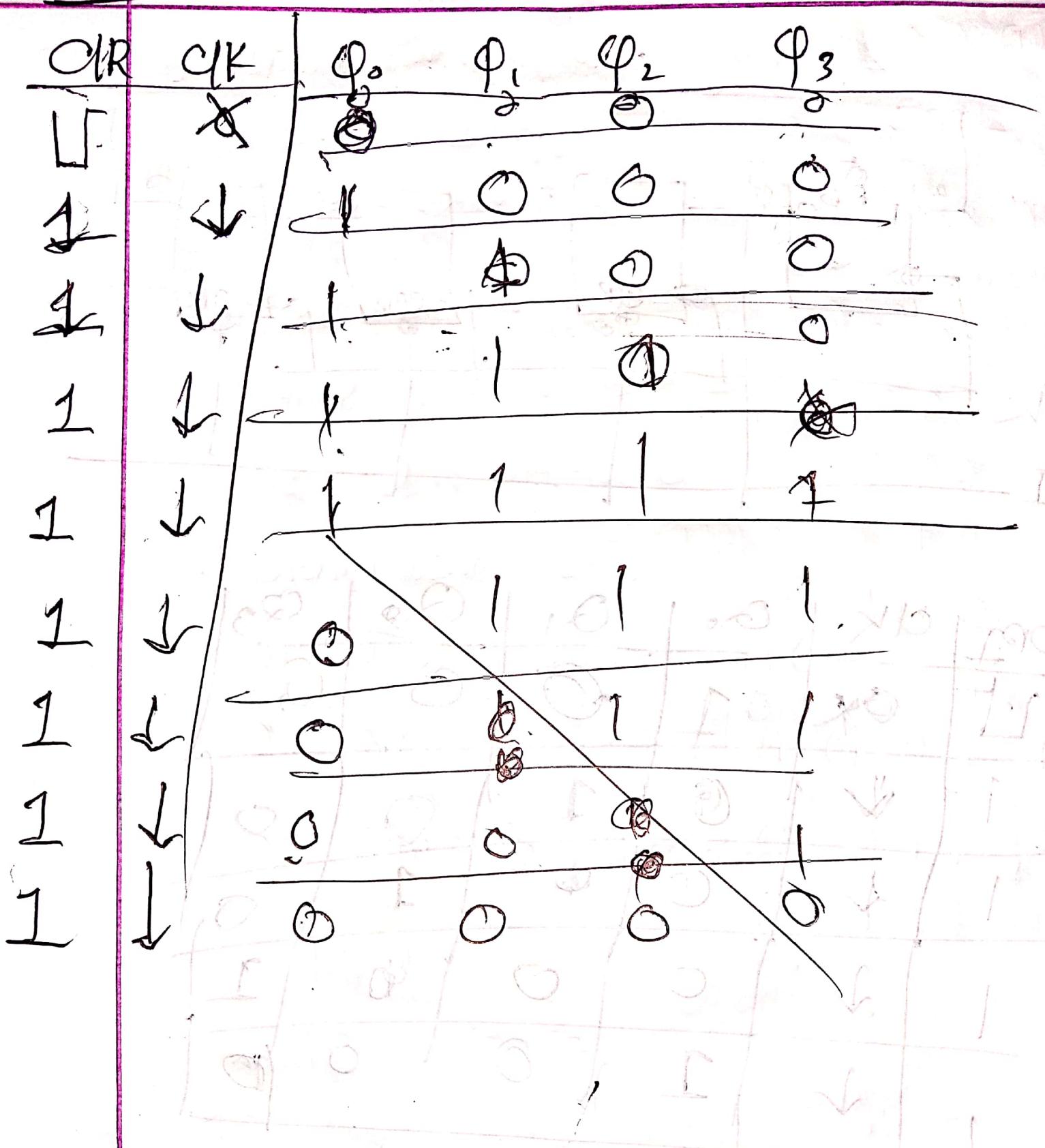


clk

ORI

ORI	clk	Q ₀	Q ₁	Q ₂	Q ₃
1	0x	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	1	0	0	0

Johnson Counter:



Propagation Delay: Propagation delay is the time required for a signal to travel from input to output when the binary output changes its value.

Boolean Algebra: Boolean algebra is the algebra of binary variables and functions.

A system of algebra in which there are only three possible values for a variable:

Hold Time: Minimum time up to which the signal on data should not change after the active edge of the clock.

Rise time: Rise time is the time taken for a signal to cross a specified lower voltage(0) threshold followed by a specified upper voltage threshold(1).

Fall Time: Rise time \rightarrow fall time