[Unit-2] Digital components

Integration circuit (IC)

An integration circuit is a small silicon semiconductor crystal, call a chip. Which containing the electronic components for the digital gates. The various gates are interconnection inside the chip to form the required circuit.

[A] Small scale integration (SSI)

Small scale integration devices contain several independent gates in a single package. The number of gates is usually less than 10.

[B] Medium scale integration (MSI)

Medium scale integration devices contain 10 to 200 gates in a single package. They usually perform specific elementary digital functions such as decoders, adders and register.

[C]Large scale integration (LSI)

Large scale integration devices contain between 200 and a few thousand gates in a single package. They include digital systems, such as processors, memory chips and programmable modules.

[D]Very large scale integration (VLSI)

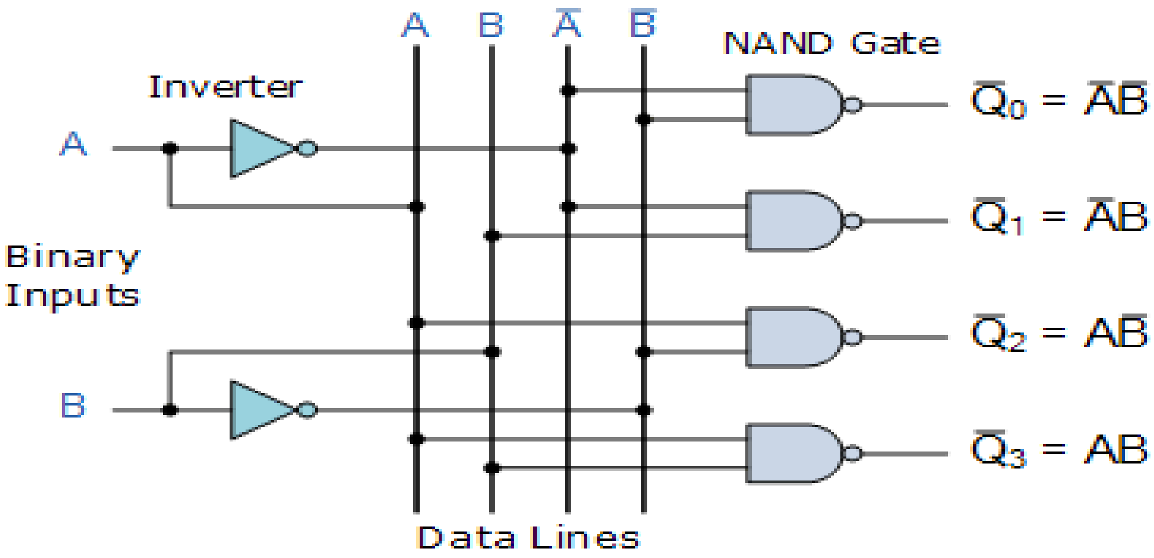
Very large scale integration devices contain thousand of gates in a single package. Example of VLSI is large memory array and complex microcomputer chips.

DECODERS

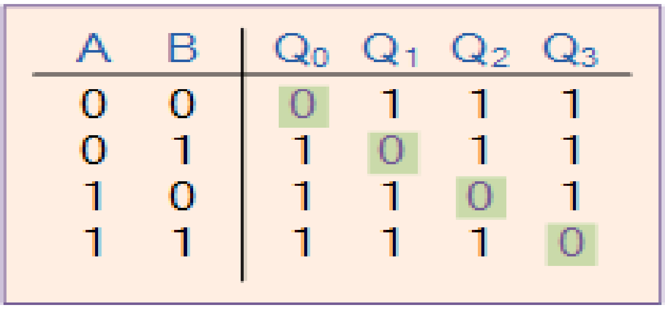
It is a combination circuit that converts binary information in from n input lines to maximum 2n unique output lines. It has n inputs and m outputs and also referred to as n-to-m line decoders.(where m=2n). A decoder translates each possible input combination of values to a distinct output line.

[A] 2 – to – 4 line Decoder (NAND gate Decoder)

A 2-to-4 line decoder with an enable input constructed with NAND gate is show in the fig. the decoder is enable when E is equal to 0. As show in the truth table at any given time ,any one output is equal to 0 and other three output are equal to 1.


A
B
$\overline{A}$
$\overline{B}$
NAND Gate
Inverter
A
Binary Inputs
B
$\overline{Q_0} = \overline{A}\,\overline{B}$
$\overline{Q_1} = \overline{A}B$
$\overline{Q_2} = A\overline{B}$
$\overline{Q_3} = AB$
Data Lines

Truth Table

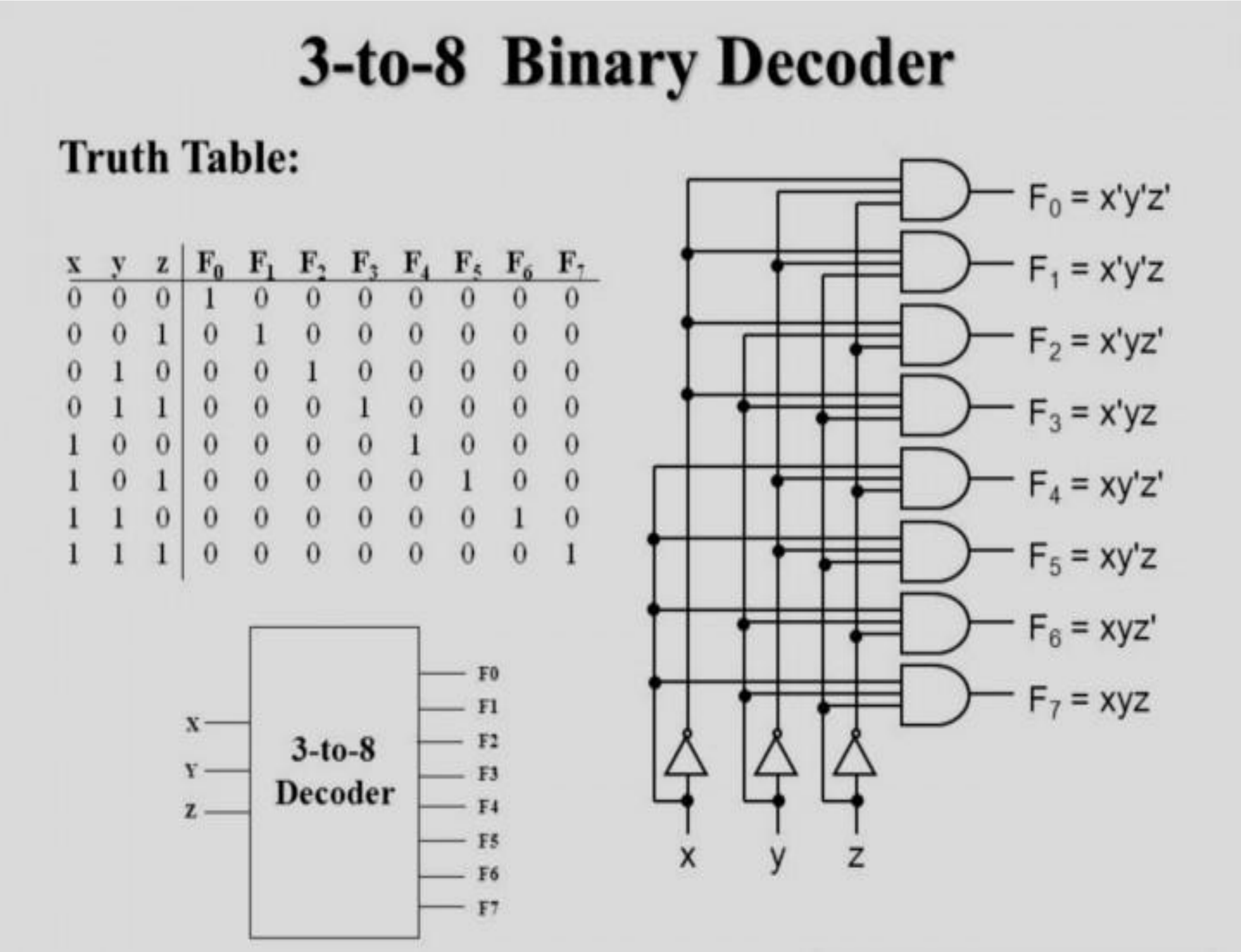


A	B	Q_0	Q_1	Q_2	Q_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

[B] 3 – to – 8 line Decoder

This decoder has three input lines and eight output lines, hence it is known as 3-to-8 line decoder. It is also called binary-to-octal decoder because it takes a 3-bit binary input code and activates one of the eight (octal) output corresponding to that code.

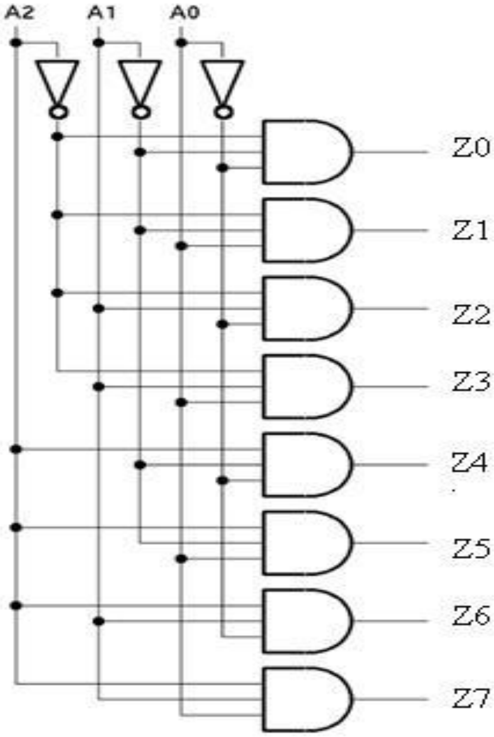
The logic diagram of 3-to-8 line decoder is show in the fig. the three data inputs, A0,A1 and A2 are decoded into 8 output. Each output representing one of the combinations of the 3 binary input variables. The three NOT gates provide the complement of the inputs and each of the eight AND gates generates one of the binary combination.



3-to-8 Binary Decoder

Truth Table:

x	y	z	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Application of Decoder

- Decoders are widely used in memory systems of computers.
- 2-to-4 decoder used in address decoding.
- With addition input line, decoder can be used as a demultiplexer.
- BCD-to-Seven segment decoder are useful for display any decimal number(i.e. 0 to 9)
- The **Decoders** were used in analog to digital conversion in analog **decoders**.
- Used in electronic circuits to convert instructions into CPU control signals.
- They mainly used in logical circuits, data transfer.

ENCODER

An encoder is combinational logic circuit that perform reverse operation of decoder.

Encoding is exactly opposite process to decoding.

It is a process of converting familiar number into a coded format.

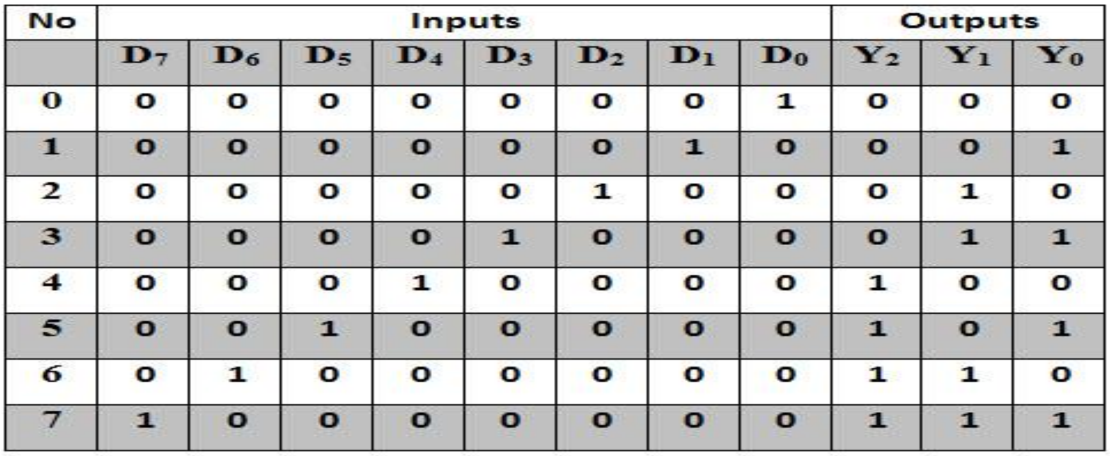
Decoder has number of output lines and at a time only one output is high, while encoder has number of input lines and only one of its activated (high) at a given time.

An encoded has 2n (or less) input lines and m output lines. The output lines generate the binary code corresponding to the input values.

Octal-to-Bibary ENCODER (3 – to – 8 line Decoder)

it is referred as 8- to- 3 encoder, because it has 8 input line s and 3 output lines. This circuit accept 8 input lines and produces a 3-bit output code corresponding to be activated input. The circuit and truth table of octal to binary encoder is displayed in the fig.

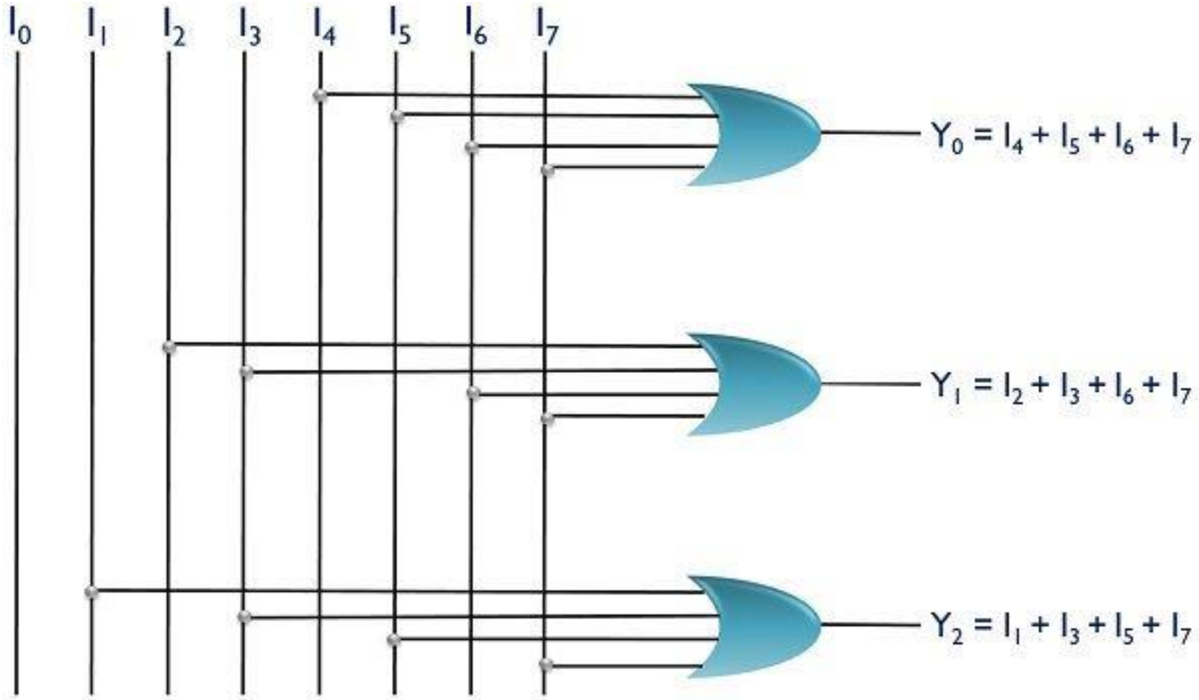
Truth table



No	Inputs								Outputs		
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	1	0	0	0	1	0
3	0	0	0	0	1	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	0	0	0	0	1	0	1
6	0	1	0	0	0	0	0	0	1	1	0
7	1	0	0	0	0	0	0	0	1	1	1

.

Circuit in 3 – to – 8 line Decoder


I_0
I_1
I_2
I_3
I_4
I_5
I_6
I_7
$Y_0 = I_4 + I_5 + I_6 + I_7$
$Y_1 = I_2 + I_3 + I_6 + I_7$
$Y_2 = I_1 + I_3 + I_5 + I_7$

Logic Circuit for Octal to Binary Encoder

Electronics Desk

MULTIPLEXER

Multiplexer is a combinational circuit that has maximum of 2^n data inputs, ‘n’ selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

Since there are ‘n’ selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input. Multiplexer is also called as **Mux**.

4x1 Multiplexer

4x1 Multiplexer has four data inputs I_3, I_2, I_1 & I_0, two selection lines s_1 & s_0 and one output Y. The **block diagram** of 4x1 Multiplexer is shown in the following figure.

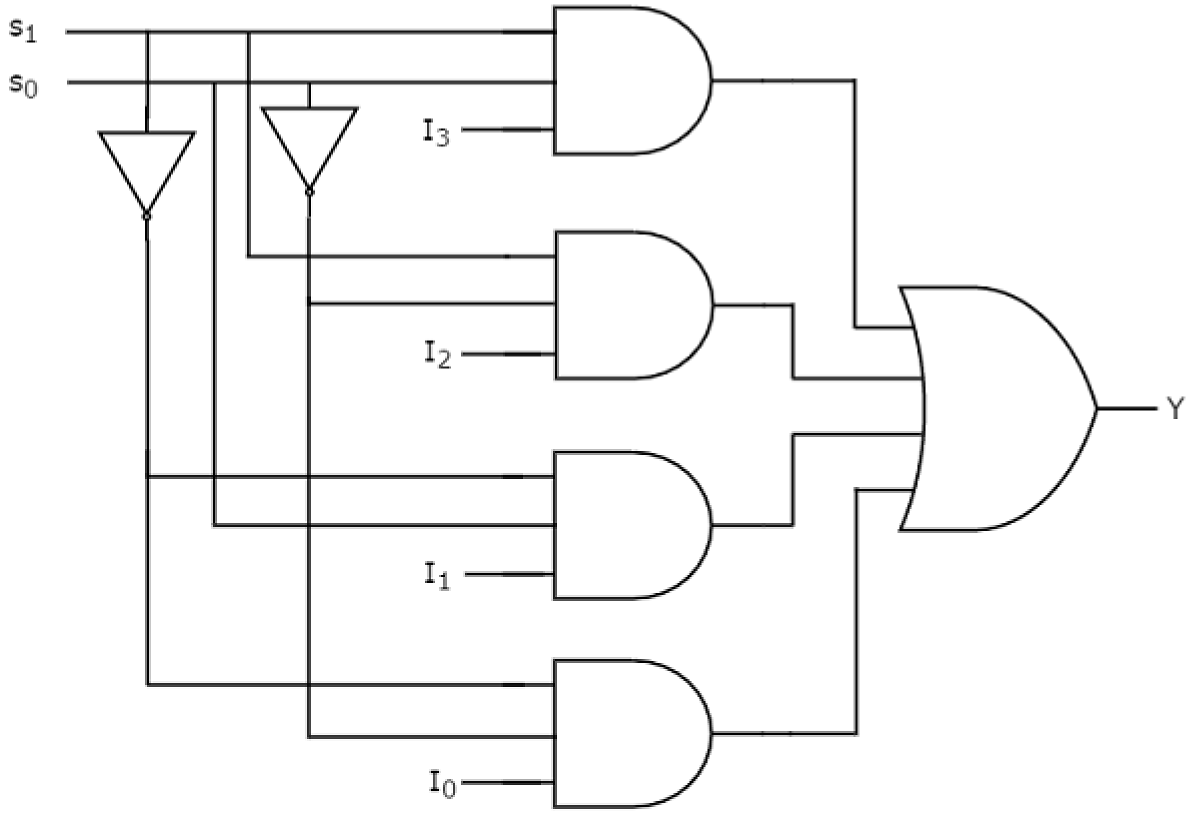
One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines. **Truth table** of 4x1 Multiplexer is shown below.

Selection Lines		**Output**
S_1	**S_0**	**Y**
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

From Truth table, we can directly write the **Boolean function** for output, Y as

$$Y=S1'S0' I0 + S1'S0 I1 + S1S0' I2 + S1S0 I3$$

We can implement this Boolean function using Inverters, AND gates & OR gate. The **circuit diagram** of 4x1 multiplexer is shown in the following figure.



DE-MULTIPLEXER

1x4 De-Multiplexer

1x4 De-Multiplexer has one input I, two selection lines, s_1 & s_0 and four outputs Y_3, Y_2, Y_1 &Y_0. The **block diagram** of 1x4 De-Multiplexer is shown in the following figure.

The single input 'I' will be connected to one of the four outputs, Y_3 to Y_0 based on the values of selection lines s_1 & s0. The **Truth table** of 1x4 De-Multiplexer is shown below.

Selection Inputs		**Outputs**			
S_1	**S_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
0	0	0	0	0	**I**
0	1	0	0	**I**	0
1	0	0	**I**	0	0
1	1	**I**	0	0	0

From the above Truth table, we can directly write the **Boolean functions** for each output as

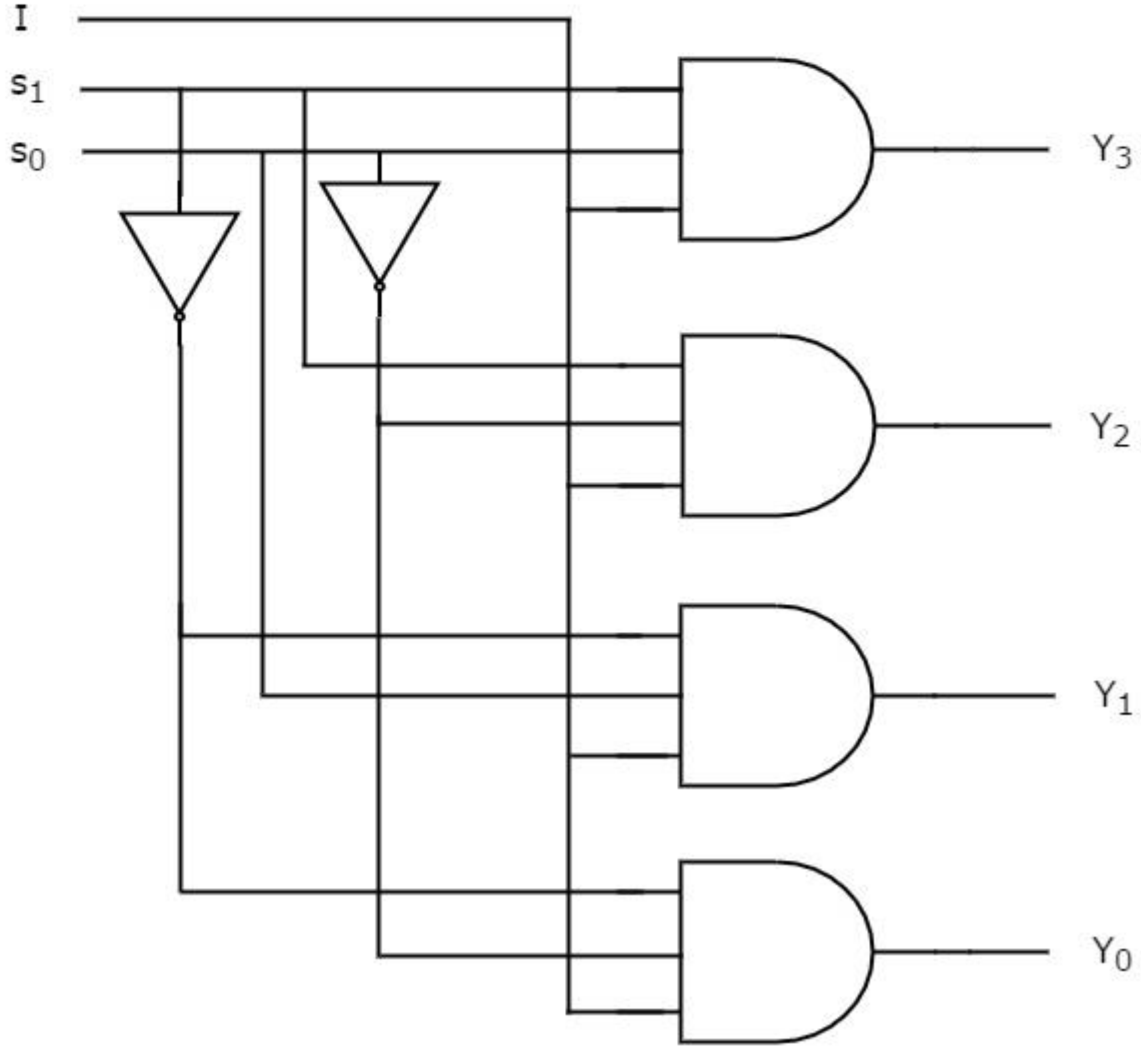
$$Y3=s1s0\ I$$

$$Y2=s1s0'\ I$$

$$Y1=s1's0\ I$$

$$Y0=s1's0'\ I$$

We can implement these Boolean functions using Inverters & 3-input AND gates. The **circuit diagram** of 1x4 De-Multiplexer is shown in the following figure.



[a] REGISTERS

(Register is a group of flip-flop with each flip-flop capable of storing 1 bit of digital information.)As discussed in Chapter-2, a flip-flop can store 1-bit of information. Thus an array of flip-flops is required to store binary information, the number of flip-flops is required being equal to the number of bits in the binary word. Therefore n-bit register has a group of n flip-flop, and is capable of storing any binary information of n-bits.

In addition to flip-flop, register may have combination gates that perform data processing tasks. *In broad definition register consists of group of* flip-flop *and gates where* flip-flop *hold the binary information and gates control the information.*

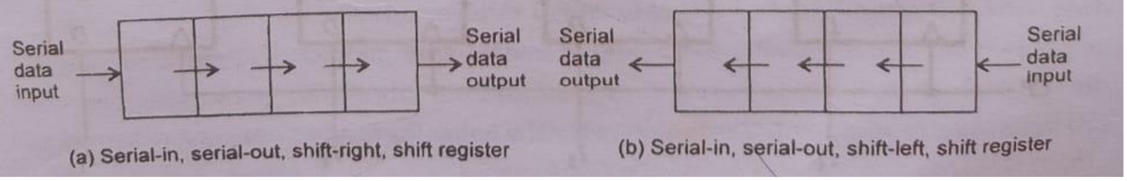
(Registers find application in a variety of digital systems including microprocessors. For example, 8085 Intel microprocessor chip contains seven 8-bit registers and five 1-bit registers.)

→ ➤ Types of Registers

(In registers the data can be entered in serial form (i.e. one bit at a time) or in parallel form (i.e. all the bits at the same time). Same way data can be retrieved either in serial or parallel form. As we know that register are made of number of Flip-Flops and these Flip-Flops connected in such a way that data may be shifted into or shifted out of them. Such registers are known as shift registers. Hence we can classify the registers based on data shifted and the way data entered and retrieved. There are four basic types of shift registers as follows:

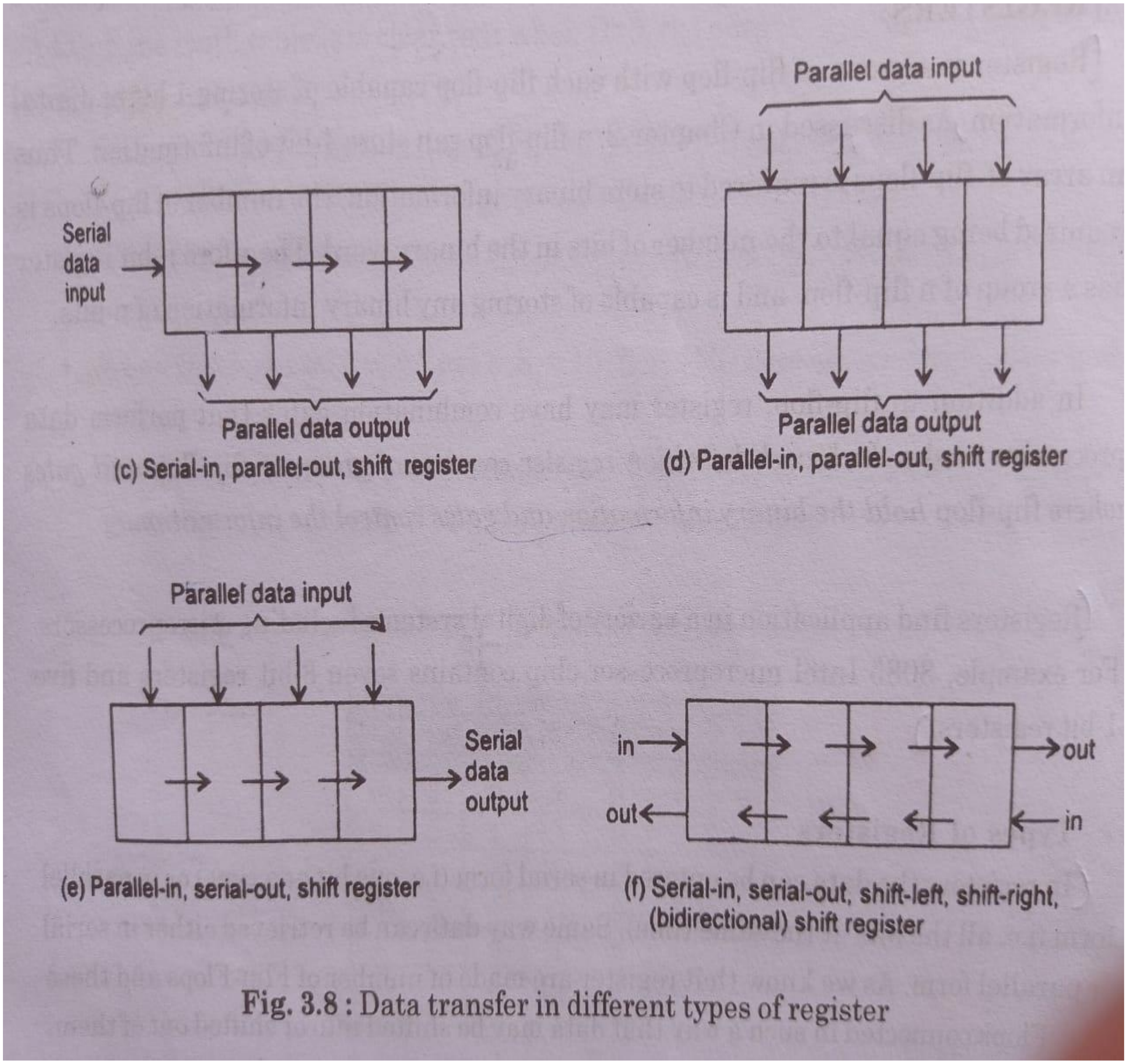
- Serial –in, serial –out.
- Serial –in, parallel –out.
- Parallel –in, serial –out.
- Parallel –in, Parallel –out.

The process of data shifting in these registers is illustrated in Fig-3.8.



(a) Serial-in, serial-out, shift-right, shift register

(b) Serial-in, serial-out, shift-left, shift register



(c) Serial-in, parallel-out, shift register

(d) Parallel-in, parallel-out, shift register

(e) Parallel-in, serial-out, shift register

(f) Serial-in, serial-out, shift-left, shift-right, (bidirectional) shift register

Fig. 3.8 : Data transfer in different types of register

[ii] Shift Register (Serial-in, Serial-out)

(A register is capable of shifting its binary information in one or both directions is called a Shift Register. The logical configuration of a shift register consists of chain of flip-flops in cascade, with the output of one flip-flop connected to the input of the next flip-flop. All the flip-flops receive common clock pulse that causes the shift from one stage to the next.)

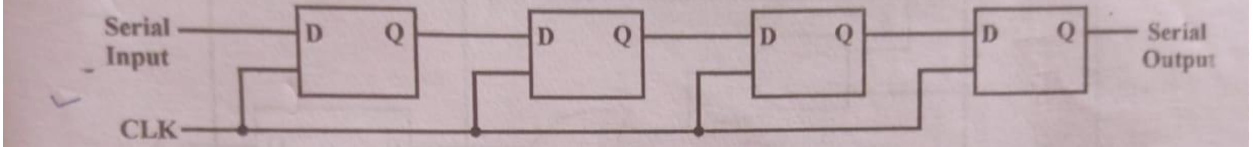


Fig-3.10 : 4 - Bit shift register

The logic diagram of 4-bit serial-in, serial-out , shift register is shown in Fig-3.10.(As shown in this figure, it has four flip-flops, thus it can store up to four bits of data. Serial data is applied at the D input of the first flip-flop. The Q output of the first flip-flop is connected to the D input of the second flip-flop, the Q output of the second flip-flop is applied to the input (D) of the third flip-flop and the Q output of the third flip-flop is connected to the D input of the fourth flip-flop. The clock is common to all flip-flops. The serial input determines what goes into the leftmost position during the shift. The serial output is taken from rightmost flip-flop.)

[iii] Register with Parallel Load

(The transfer of new information in register is referred to as *loading the register*. If all the bits of the register are loaded simultaneously with common clock pulse, we say that the loading is done in parallel.)The circuit diagram of 4-bit register is displayed in Fig-3.11. The load input in the register determines the action to be taken with each clock pulse(When the load input is 1, then I inputs are transferred into the register on the next clock pulse. When the loud input is zero, the circuit inputs I_0, I_1, I_2, I_3, are inhibited and D flip flops are reloaded with their present value, thus, maintaining the content of the register.)

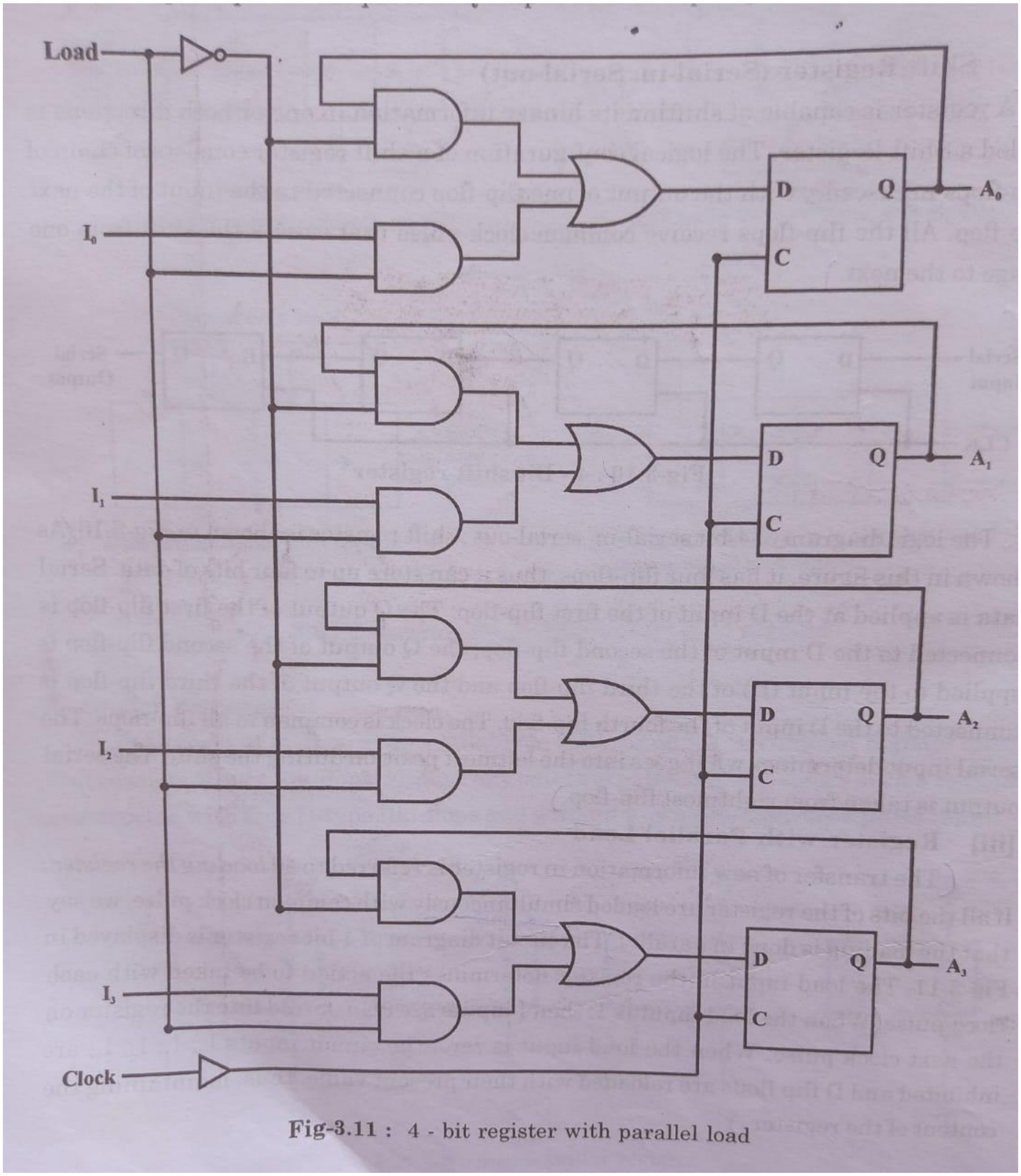
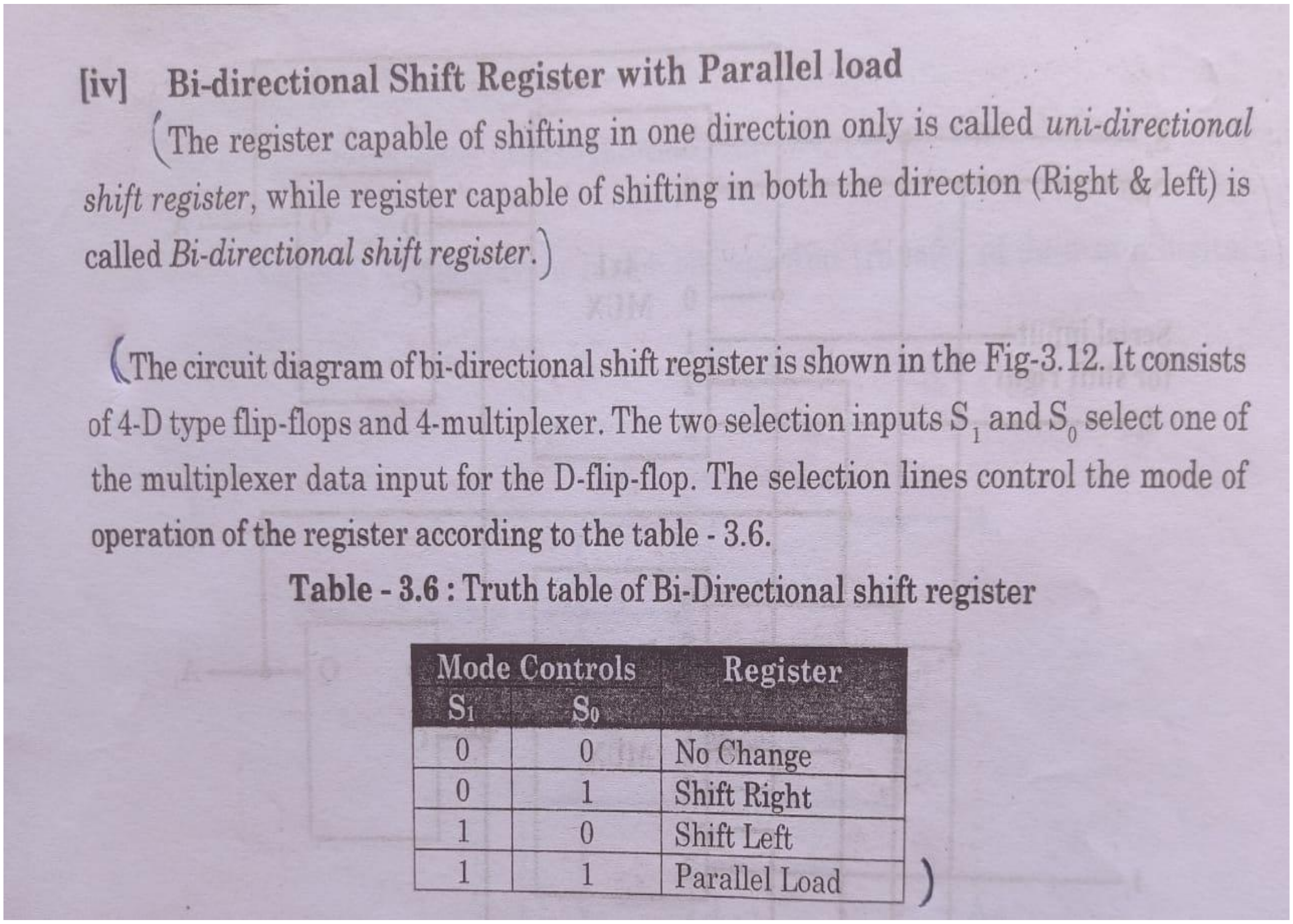


Fig-3.11 : 4 - bit register with parallel load



[iv] Bi-directional Shift Register with Parallel load

(The register capable of shifting in one direction only is called *uni-directional shift register*, while register capable of shifting in both the direction (Right & left) is called *Bi-directional shift register.*)

(The circuit diagram of bi-directional shift register is shown in the Fig-3.12. It consists of 4-D type flip-flops and 4-multiplexer. The two selection inputs S_1 and S_0 select one of the multiplexer data input for the D-flip-flop. The selection lines control the mode of operation of the register according to the table - 3.6.

Table - 3.6 : Truth table of Bi-Directional shift register

Mode Controls		Register
S_1	S_0	
0	0	No Change
0	1	Shift Right
1	0	Shift Left
1	1	Parallel Load

)

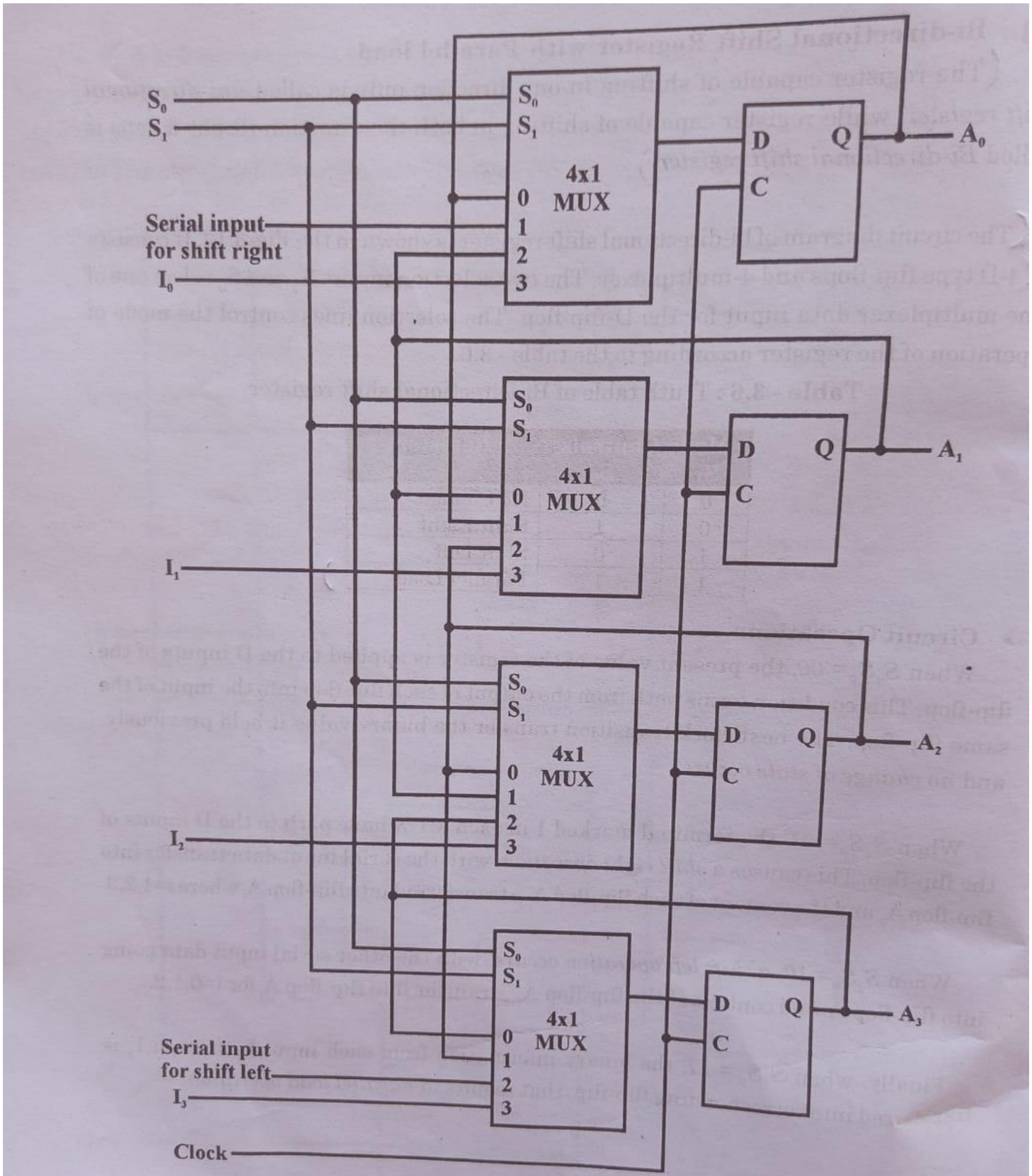


Fig - 3.12 : Bi directional shift register

[b] COUNTER

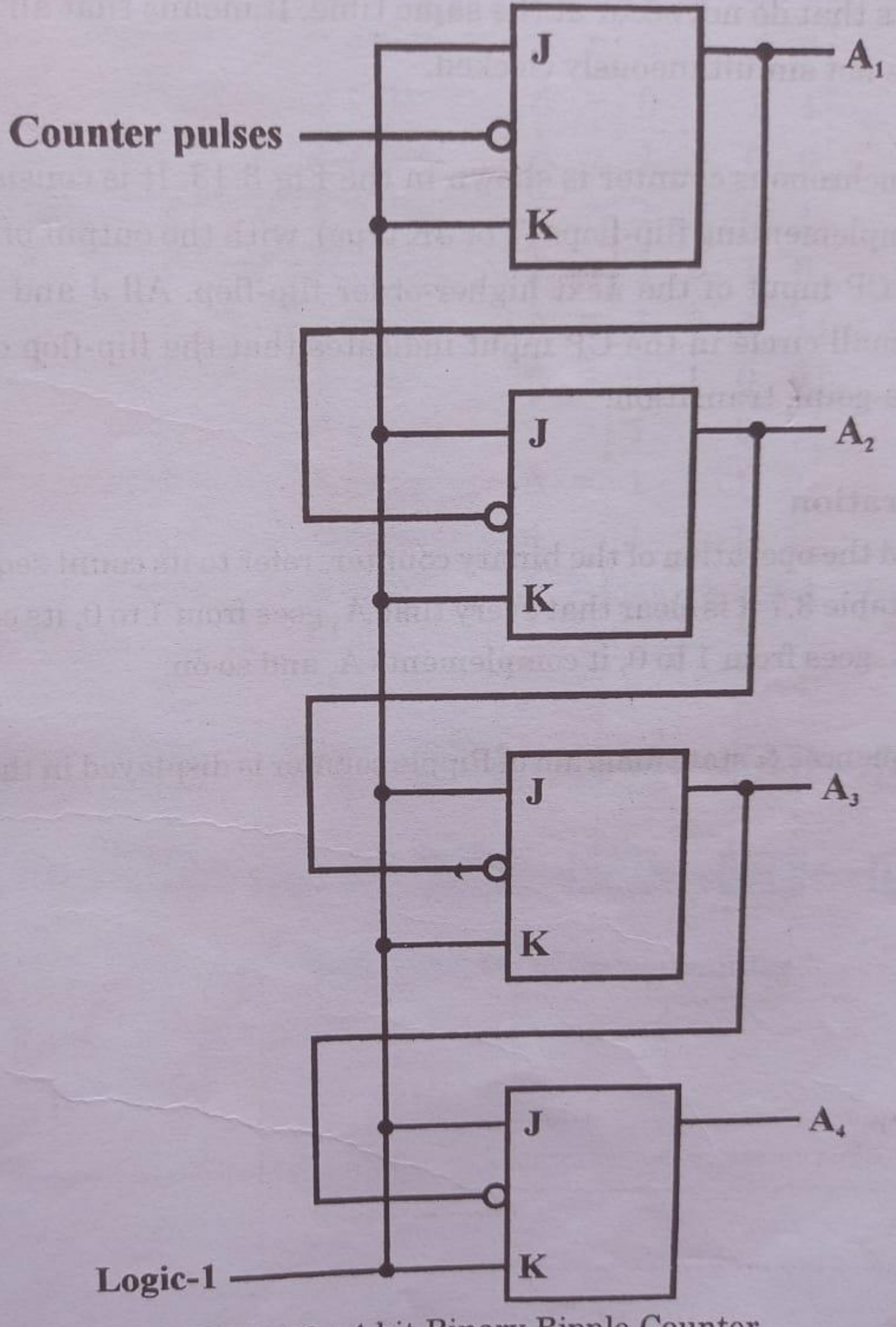
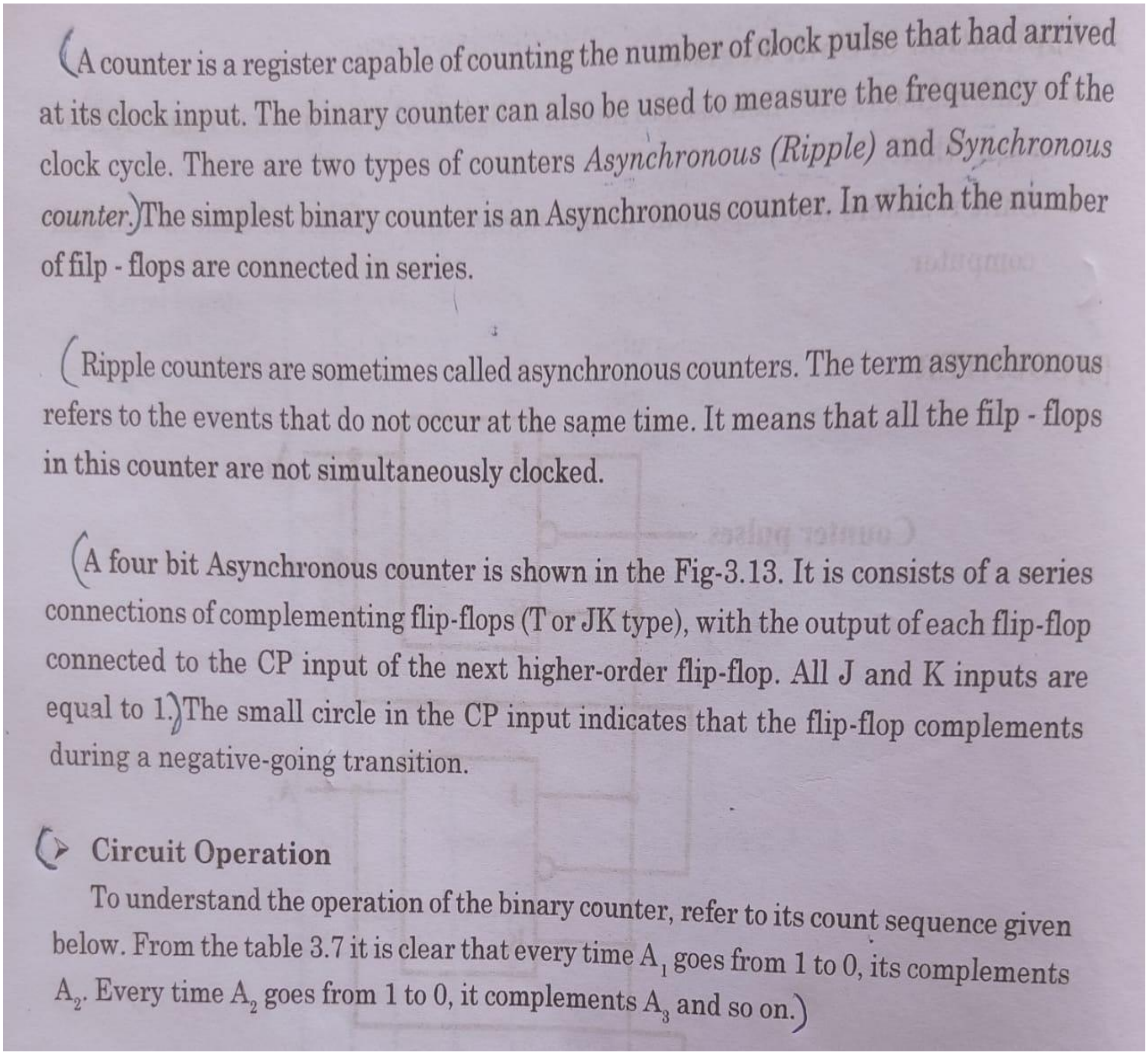


Fig-3.13 : 4-bit Binary Ripple Counter



(A counter is a register capable of counting the number of clock pulse that had arrived at its clock input. The binary counter can also be used to measure the frequency of the clock cycle. There are two types of counters *Asynchronous (Ripple)* and *Synchronous counter.*)The simplest binary counter is an Asynchronous counter. In which the number of filp - flops are connected in series.

(Ripple counters are sometimes called asynchronous counters. The term asynchronous refers to the events that do not occur at the same time. It means that all the filp - flops in this counter are not simultaneously clocked.

(A four bit Asynchronous counter is shown in the Fig-3.13. It is consists of a series connections of complementing flip-flops (T or JK type), with the output of each flip-flop connected to the CP input of the next higher-order flip-flop. All J and K inputs are equal to 1.)The small circle in the CP input indicates that the flip-flop complements during a negative-going transition.

(➢ Circuit Operation

To understand the operation of the binary counter, refer to its count sequence given below. From the table 3.7 it is clear that every time A_1 goes from 1 to 0, its complements A_2. Every time A_2 goes from 1 to 0, it complements A_3 and so on.)

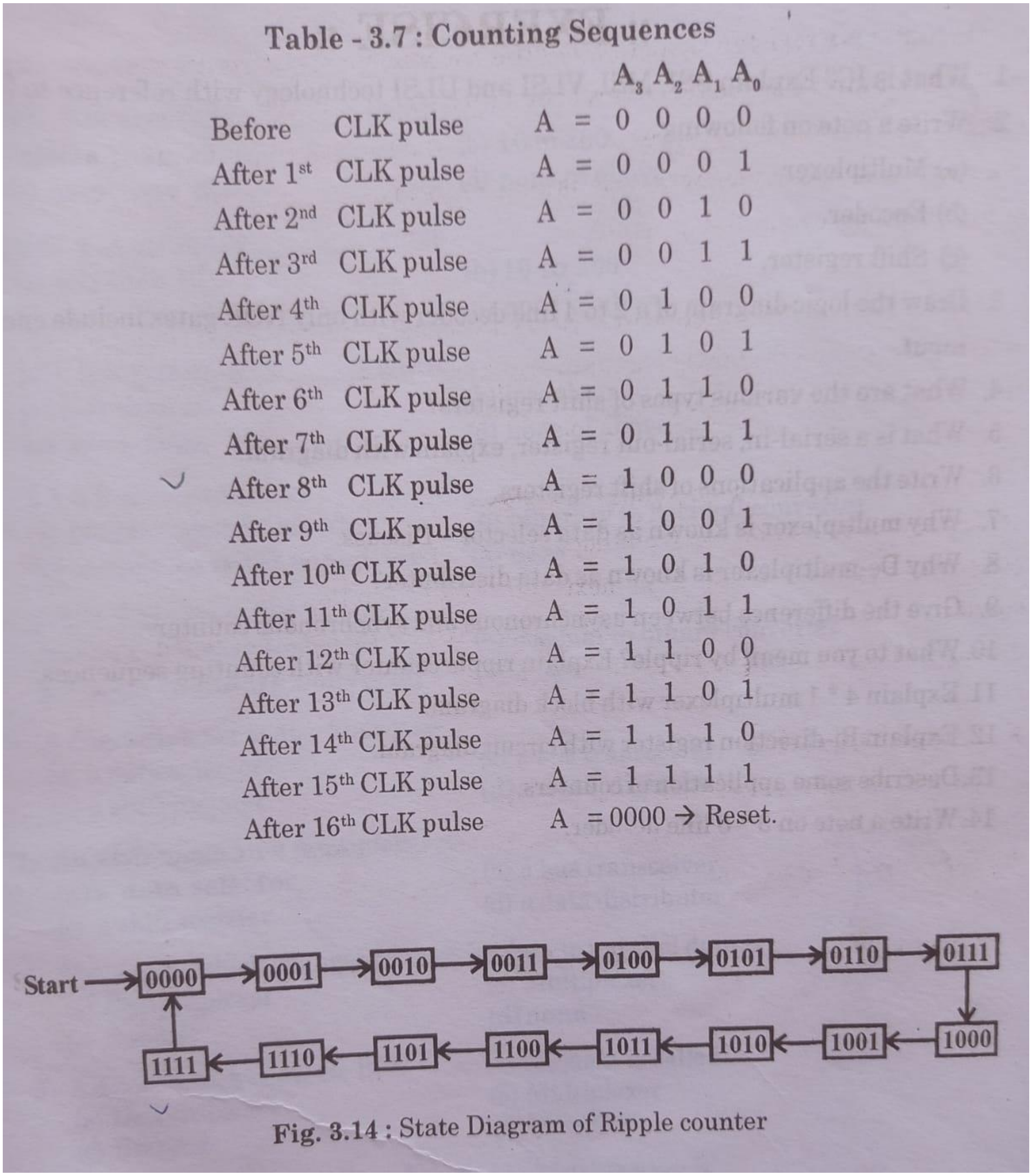


Table - 3.7 : Counting Sequences

		A_3	A_2	A_1	A_0
Before CLK pulse	A =	0	0	0	0
After 1st CLK pulse	A =	0	0	0	1
After 2nd CLK pulse	A =	0	0	1	0
After 3rd CLK pulse	A =	0	0	1	1
After 4th CLK pulse	A =	0	1	0	0
After 5th CLK pulse	A =	0	1	0	1
After 6th CLK pulse	A =	0	1	1	0
After 7th CLK pulse	A =	0	1	1	1
After 8th CLK pulse	A =	1	0	0	0
After 9th CLK pulse	A =	1	0	0	1
After 10th CLK pulse	A =	1	0	1	0
After 11th CLK pulse	A =	1	0	1	1
After 12th CLK pulse	A =	1	1	0	0
After 13th CLK pulse	A =	1	1	0	1
After 14th CLK pulse	A =	1	1	1	0
After 15th CLK pulse	A =	1	1	1	1
After 16th CLK pulse	A =0000 → Reset.				

Fig. 3.14 : State Diagram of Ripple counter