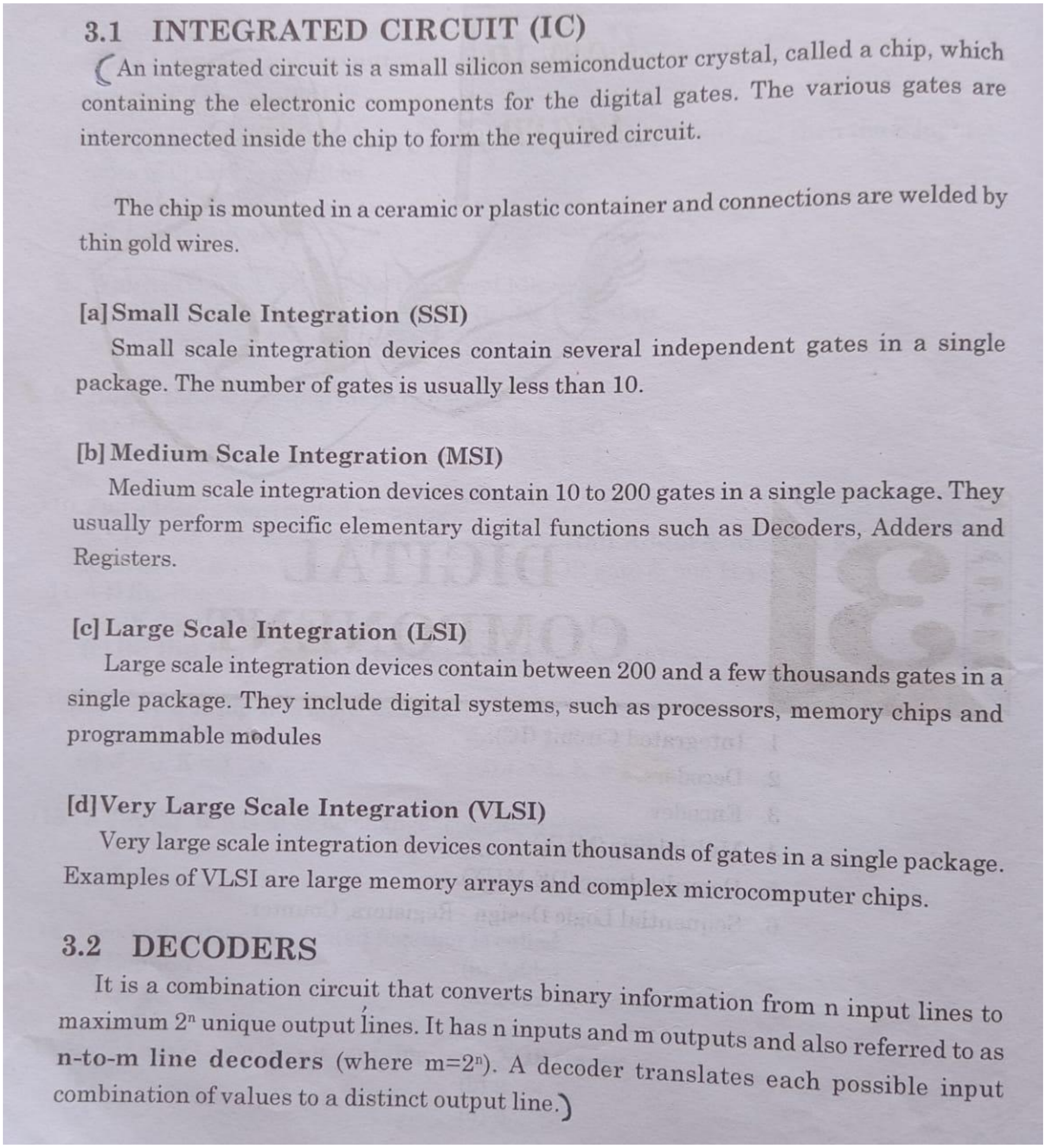
UNIT-3 DIGITAL COMPONENT



3.1 INTEGRATED CIRCUIT (IC)

(An integrated circuit is a small silicon semiconductor crystal, called a chip, which containing the electronic components for the digital gates. The various gates are interconnected inside the chip to form the required circuit.

The chip is mounted in a ceramic or plastic container and connections are welded by thin gold wires.

[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 Registers.

[c] Large Scale Integration (LSI)

Large scale integration devices contain between 200 and a few thousands 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 thousands of gates in a single package. Examples of VLSI are large memory arrays and complex microcomputer chips.

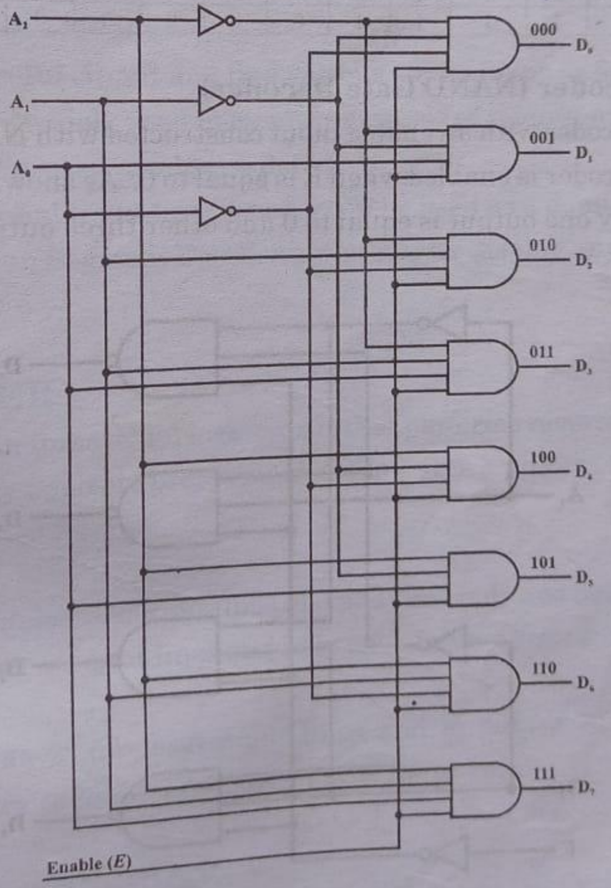
3.2 DECODERS

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

[a] 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) outputs corresponding to that code.)

The logic diagram of 3-to-8-line decoder is shown in the Fig-3.1.(The three data inputs, A_0, A_1, and A_2 are decoded into 8 outputs. 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.)



1) Small scale
2) medium scale
3) Large scale
4) very Large

Fig. - 3.1 : 3 to 8 - line Decoder

➢ **Operation**

(The operation of the decoder can be clarified using the truth table. As shown in the truth table when enable input E = 0 all the outputs are equal to zero, regardless of the values of three data inputs. This is shown by 'X' don't-care conditions in the table)- 3.1.

Table - 3.1 : Truth table of 3 to 8 - line Decoder

Enable	Inputs			Outputs							
E	A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

[b] 2-to-4-line Decoder (NAND Gate Decoder)

(A 2-to-4-line Decoder with an enable input constructed with NAND gate)is shown in the Fig-3.2.(The decoder is enabled when E is equal to 0. As shown in the truth table at any given time, any one output is equal to 0 and other three outputs are equal to 1.

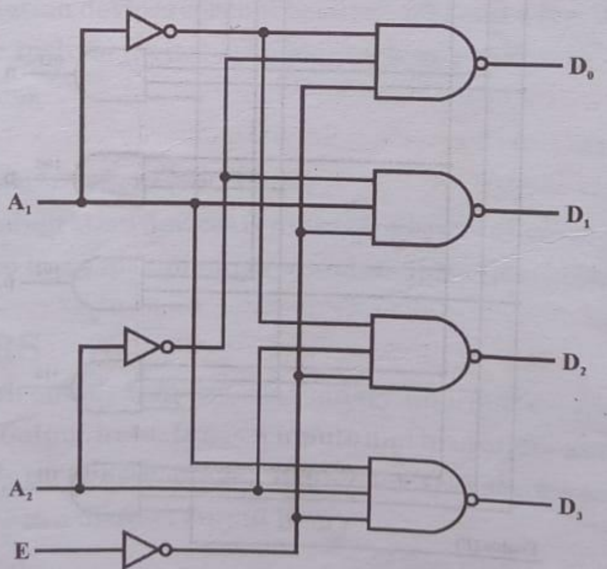


Fig. 3.2 : 2 to 4 - line NAND gate Decoder

(The decoder is disables when E equal to 1, regardless of the values of the other two inputs. 'X' shows this condition in the truth table)- 3.2.(The main application of 2-to-4 decoder is in address decoding.)

Table - 3.2 : Truth table of NAND gate decoder

Enable	Inputs		Outputs			
E	A_1	A_2	D_0	D_1	D_2	D_3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	X	X	1	1	1	1

➢ Applications of Decoder

(Decoders are useful circuit and find various applications as follows:

- Decoders are widely used in memory systems of computers.
- 2-to-4 decoder used in address decoding.
- With additional 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).)

3.3 ENCODER

(An encoder is combinational logic circuit that performs reverse operation of decoder. Encoding is exactly opposite process to decoding. It is a process of converting familiar numbers into a coded format.

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

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

➢ Octal-to-Binary ENCODER

It is also referred as 8 x 3 encoder, because it has 8 input lines and 3 output lines. This circuit accepts 8 input lines and produces a 3-bit output code corresponding to the) activated input) The circuit and truth table of octal to binary encoder is displayed in the Fig-3.3.

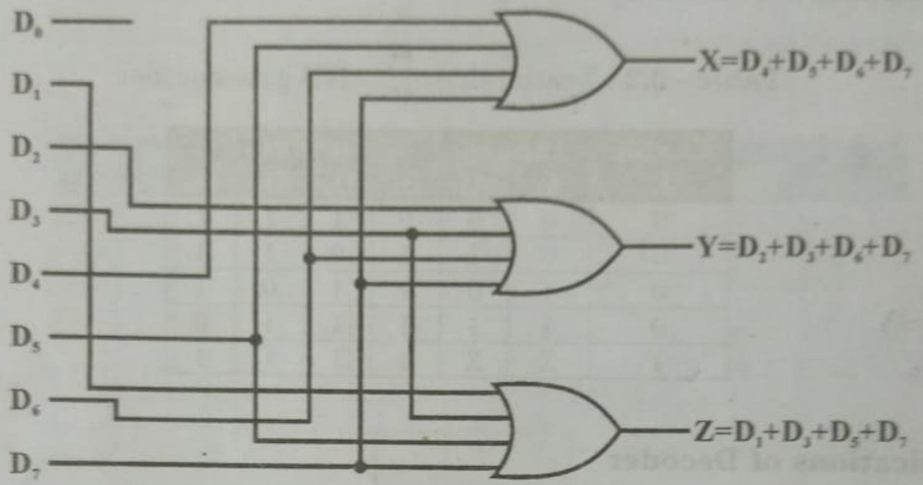


Fig. 3.3 : Logic circuit of Octal to Binary Encoder

Table - 3.3 : Truth table of Encoder

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	X	Y	Z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

(The encoder can be implemented with OR gate and their inputs are determined directly from the truth table)(See table 3.3).(Output Z = 1 when input octal digit is 1,3,4,7 output Y = 1 when input octal digit 2,3,6,7 and output X = 1 for digits 4,5,6,7. These conditions can be expressed by the following functions:

$$\boxed{\begin{aligned} Z &= D_1 + D_3 + D_5 + D_7 \\ Y &= D_2 + D_3 + D_6 + D_7 \\ X &= D_4 + D_5 + D_6 + D_7 \end{aligned}}$$)

3.4 MULTIPLEXER (MUX)

(A Multiplexer is combination circuit that is used to direct one out of 2^n input data lines to a single output line. It is also known as Data selector, because it selects one of many inputs and directs it to the output. The selection of particular input data line is controlled by a set of selection inputs. Normally there are 2^n input data lines and n input selection lines.

The block diagram of 4 to 1 line MUX is shown in fig.-3.4 and logic circuit is displayed in Fig-3.5. Each of the 4 data inputs I_0 through is I_3 applied to one input of AND gate. The two selection inputs S_1 and S_0 are decoded to select a particular AND gate. The output of the AND gates are applied to a single OR gate to provide the single output.)

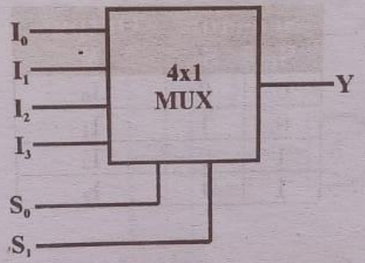


Fig. 3.4 : Block Diagram of 4 x 1 MUX

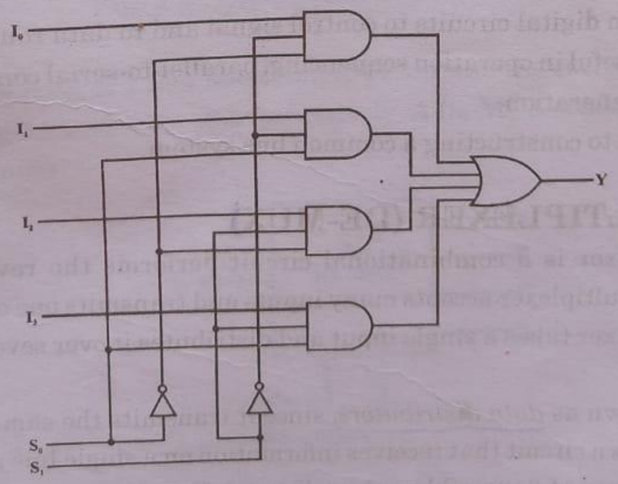


Fig. 3.5 : Logic circuit of MUX

➢ Circuit Operation

Selection lines S_1 and S_0 are decoded to select particular AND gate. To demonstrate circuit operation considers the case when S_1 and S_0 are both equal to zero. The AND gate associated with input I_0 produce high output while other three AND gates produce low output. The OR gate output is now equal to I_0.

When $S_0 = 0$ and $S_1 = 1$, then AND gate associated with I_1 produce high output while the rest of AND gates produce low output. Therefore, OR gate output is now I_1.

Similarly for the cases: $S_0 = 1$ and $S_1 = 0$, the AND gate associated with I_2 produce high output and for $S_0 = S_1 = 1$, the AND gate associated with I_3 produce high output.

Table - 3.4 : Truth table of 4 x 1 MUX

Selector		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

➢ Application of Multiplexer

- It is used to connecting two or more sources to a single destination among computer units.
- It is used in digital circuits to control signal and in data routing.
- It is also useful in operation sequencing, parallel-to-serial conversion and in logic function generation.
- It is useful to constructing a common bus system.

3.5 DEMULTIPLEXER (DE-MUX)

A demultiplexer is a combinational circuit performs the reverse operation of multiplexer. A multiplexer accepts many inputs and transmits one of it's to the output, while demultiplexer takes a single input and distributes it over several outputs.

It is also known as *data distributors,* since it transmits the same data to different destinations. It is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. A Decoder with an enable input can function as a Demultiplxer.

➢ 1-to-4 line Demultiplexer

Fig-3.6 shows a 1-to-4 line Demultiplexer circuit (As shown in this fig the input data lines goes to all of the AND gates. The two select lines S_0 and S_1 activate (enabled) only one gate at a time.)

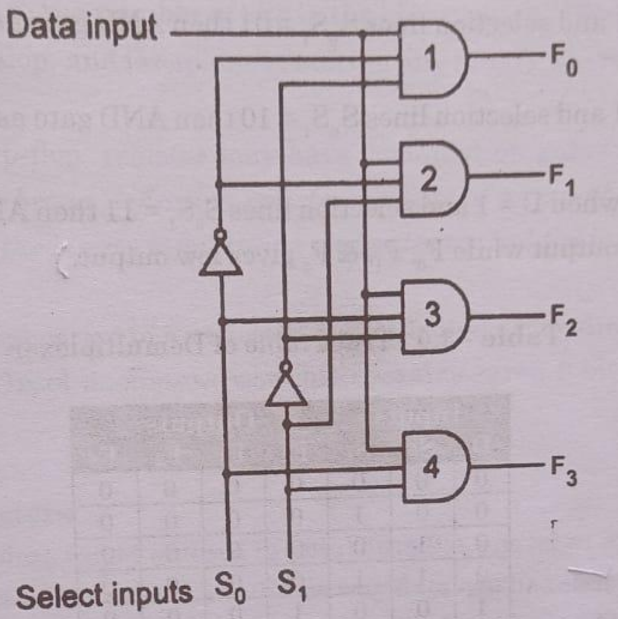


Fig. 3.6 : Logic circuit of Demultiplexer

The block diagram of 1 to 4 line Demultiplexer and the corresponding truth table are shown in Fig-3.7 and table - 3.5 respectively. A De-MUX sends its single data input to one of its outputs

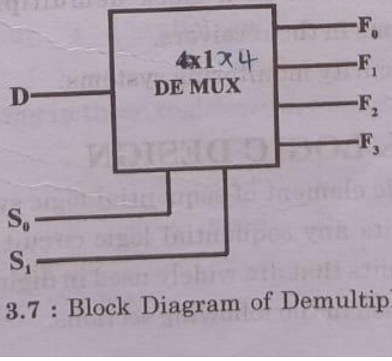
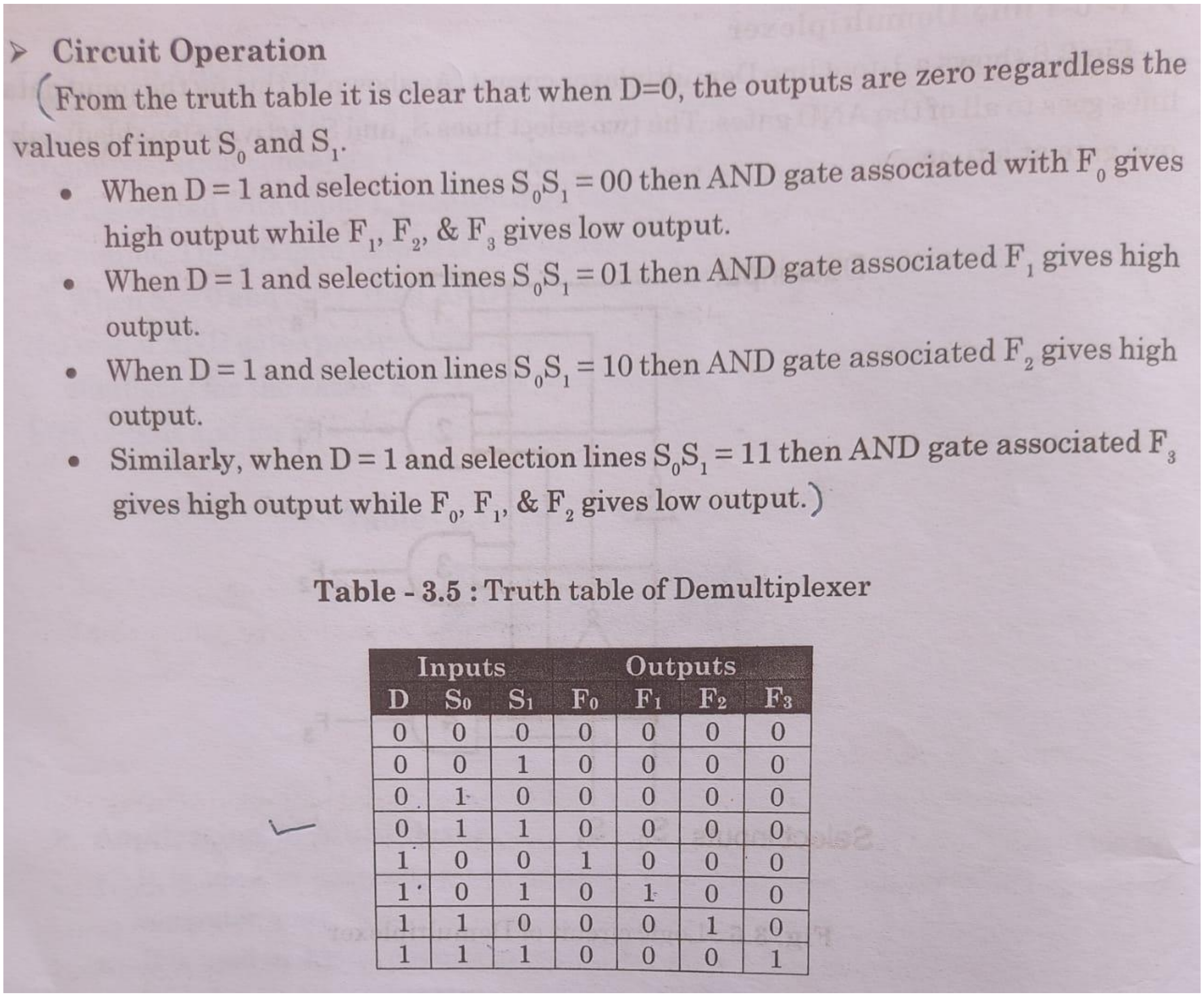


Fig. 3.7 : Block Diagram of Demultiplexer



➢ Circuit Operation

From the truth table it is clear that when D=0, the outputs are zero regardless the values of input S_0 and S_1.

- When D = 1 and selection lines S_0S_1 = 00 then AND gate associated with F_0 gives high output while F_1, F_2, & F_3 gives low output.
- When D = 1 and selection lines S_0S_1 = 01 then AND gate associated F_1 gives high output.
- When D = 1 and selection lines S_0S_1 = 10 then AND gate associated F_2 gives high output.
- Similarly, when D = 1 and selection lines S_0S_1 = 11 then AND gate associated F_3 gives high output while F_0, F_1, & F_2 gives low output.

Table - 3.5 : Truth table of Demultiplexer

Inputs			Outputs			
D	S_0	S_1	F_0	F_1	F_2	F_3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	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

[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.*

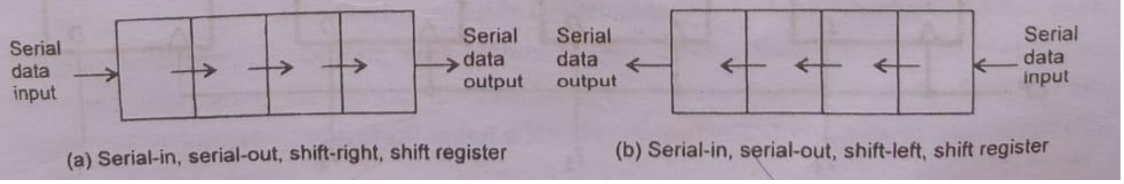
(Registers find application in a variety of digital systems including microprocessors. For example, 8085 Intel microprocessor chip contains seven 8-bit registers and five 1-bit registers.)

→ ➤ 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:

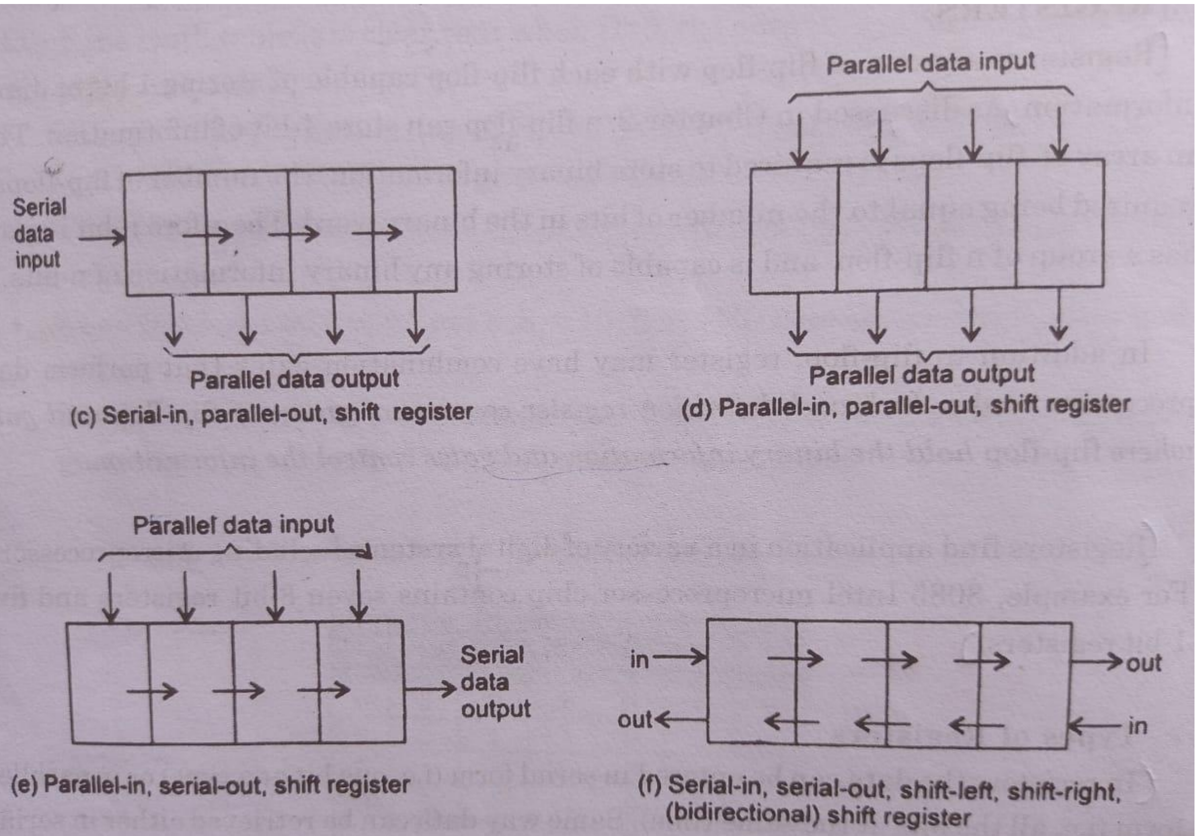
- Serial –in, serial –out.
- Serial –in, parallel –out.
- Parallel –in, serial –out.
- Parallel –in, Parallel –out.

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.)

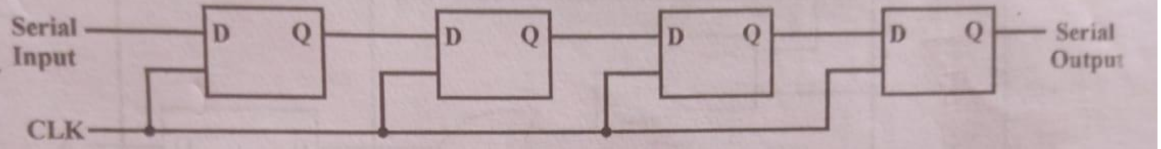


Fig-3.10 : 4 - Bit shift register

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.)

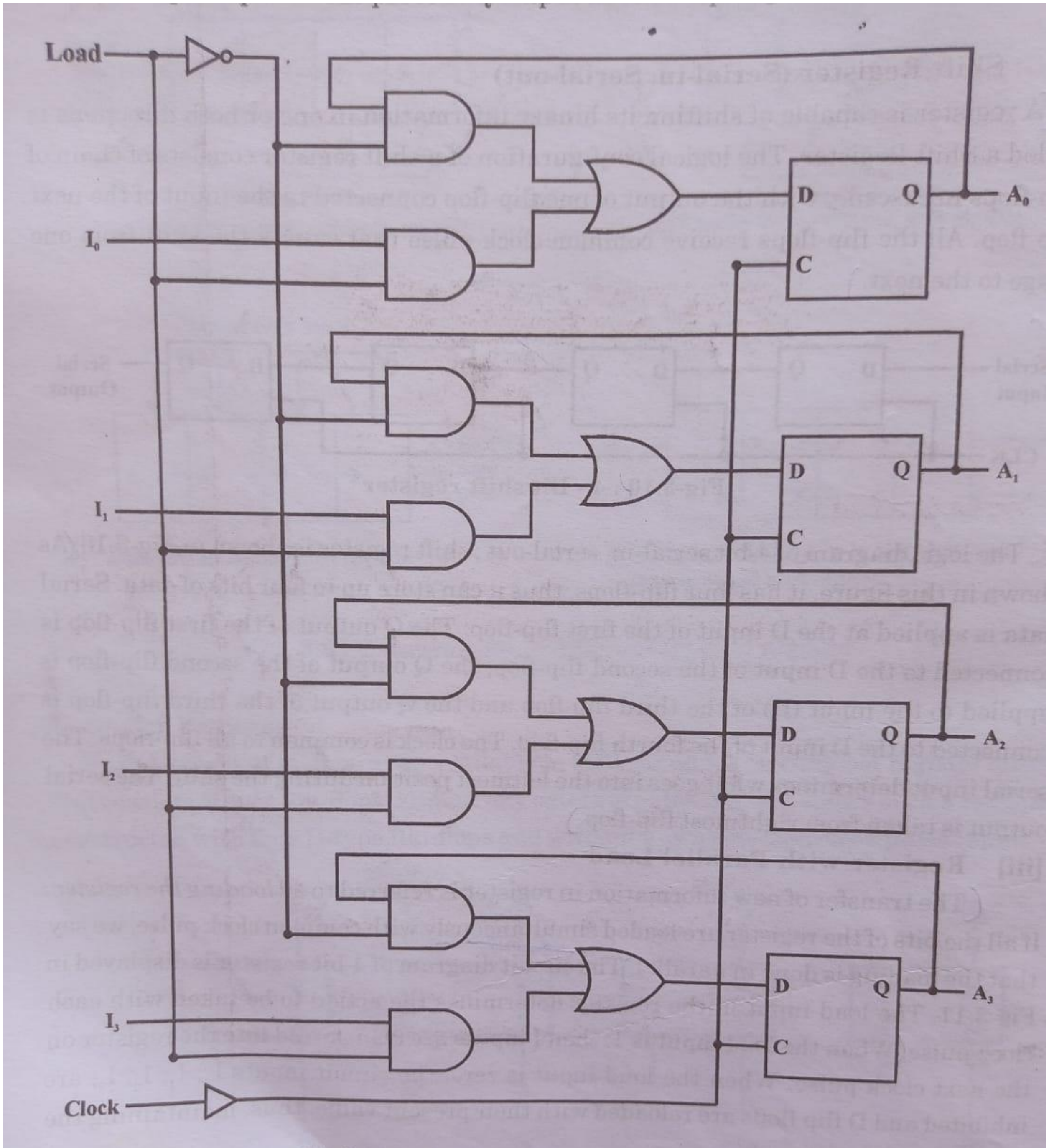
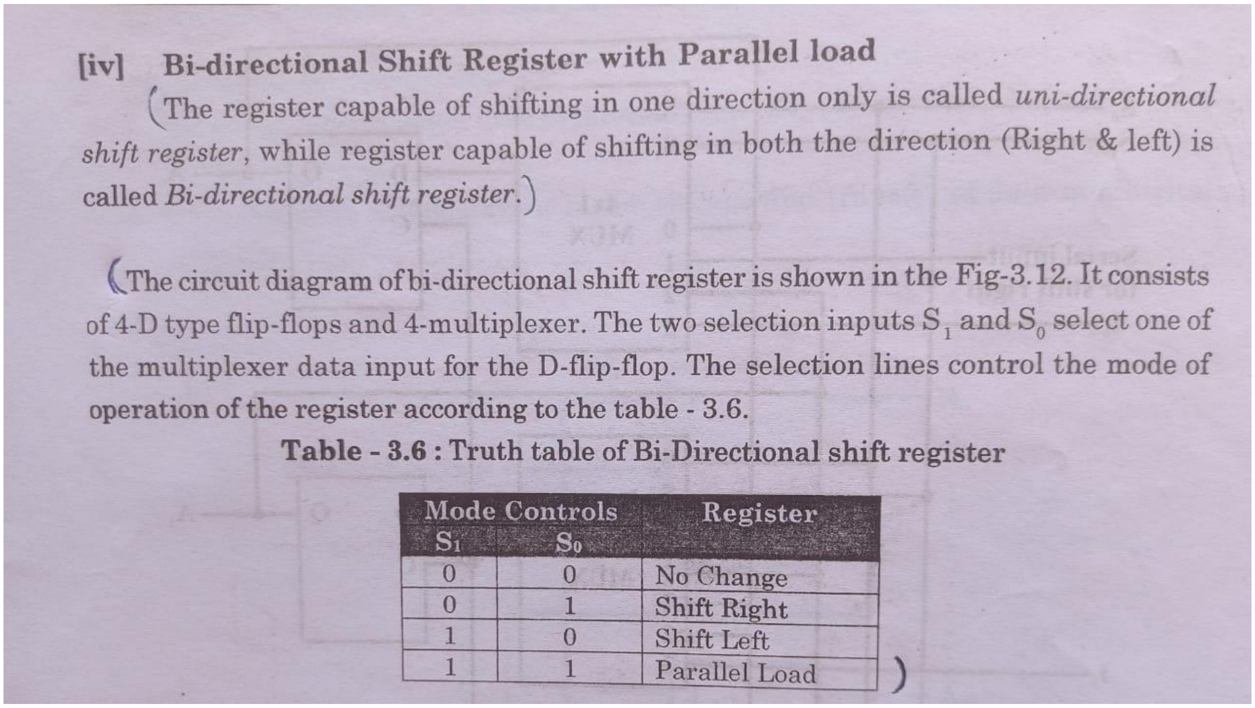


Fig-3.11 : 4 - bit register with parallel load



[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

)

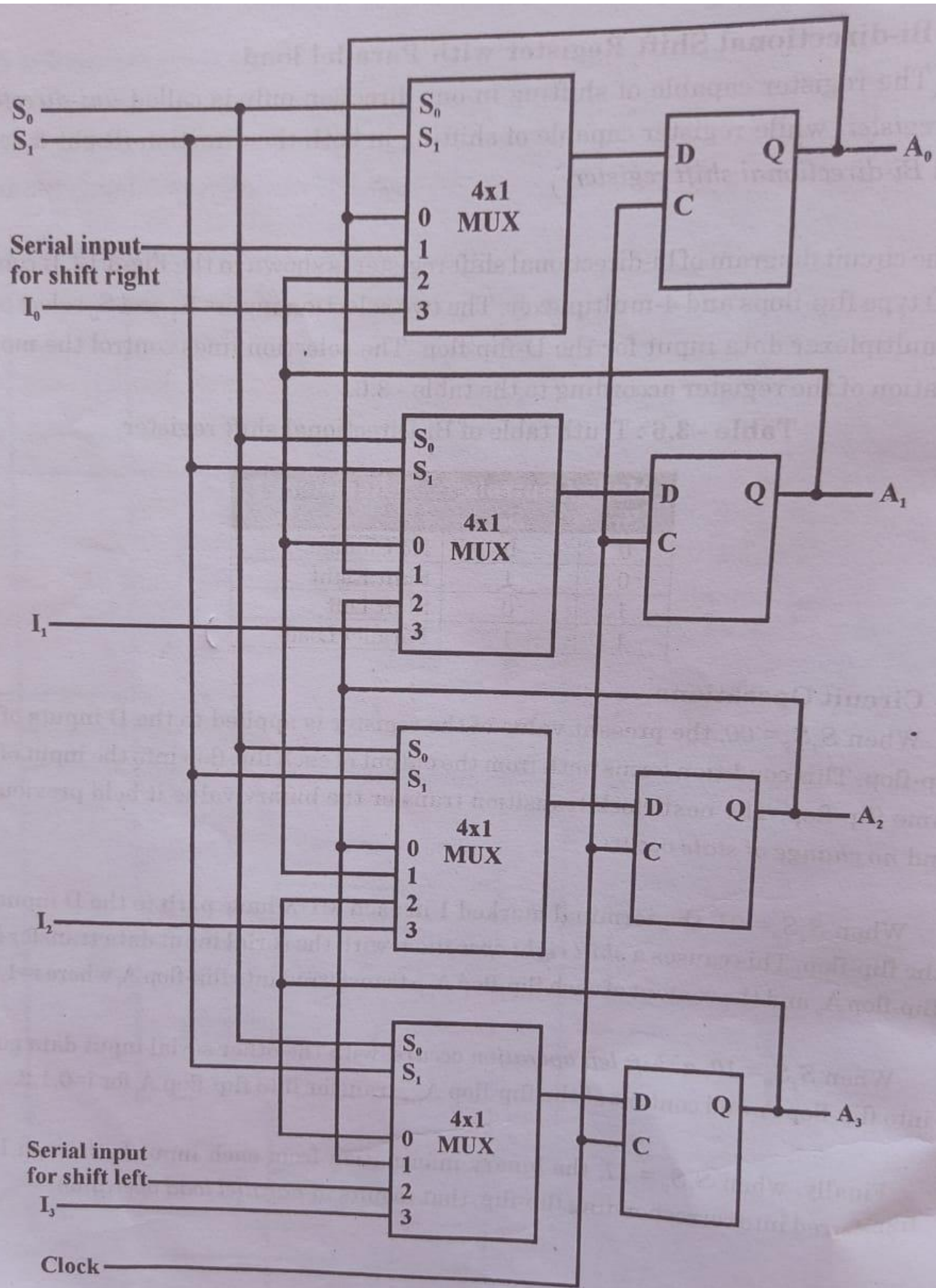


Fig - 3.12 : Bi directional shift register

[b] COUNTER

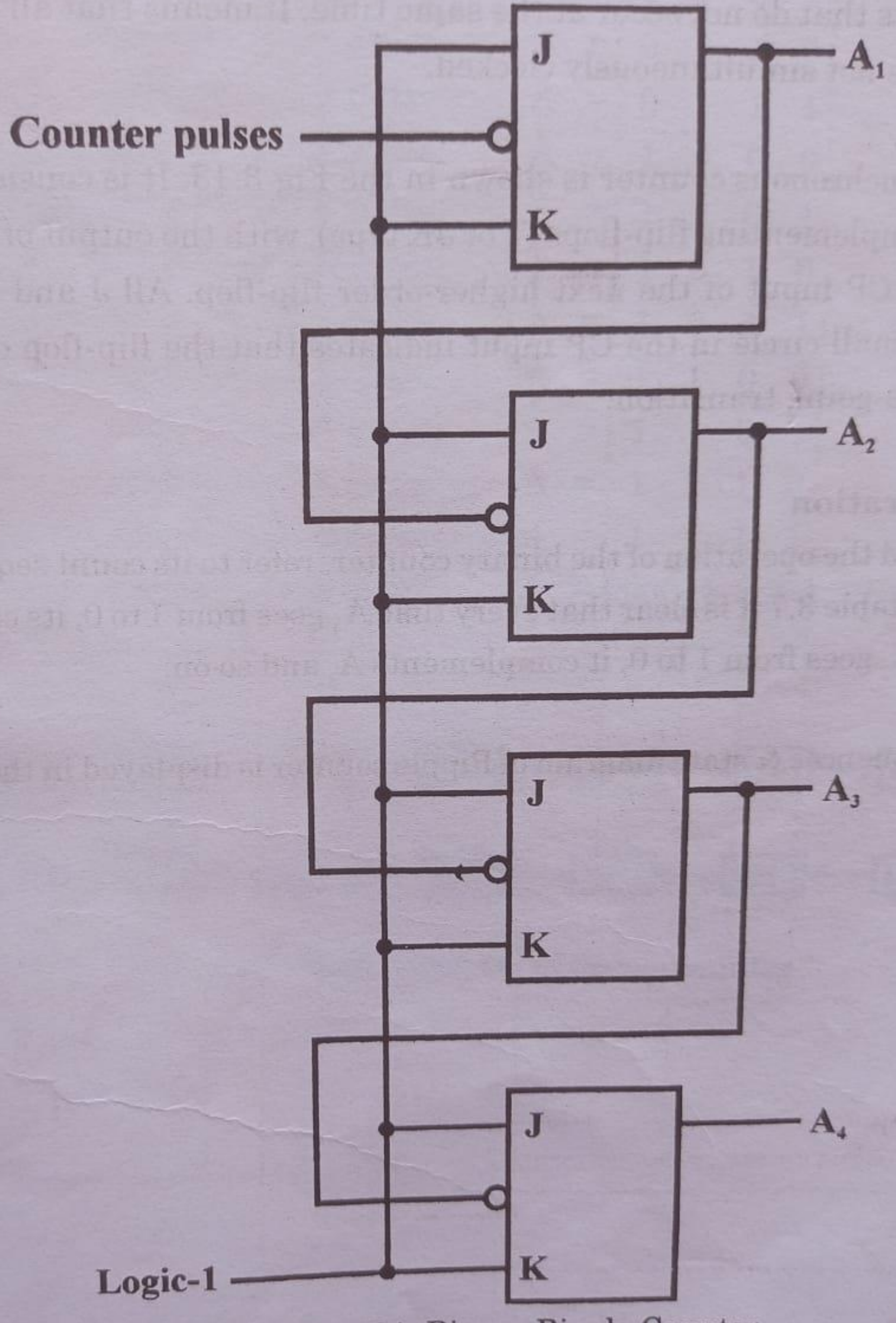
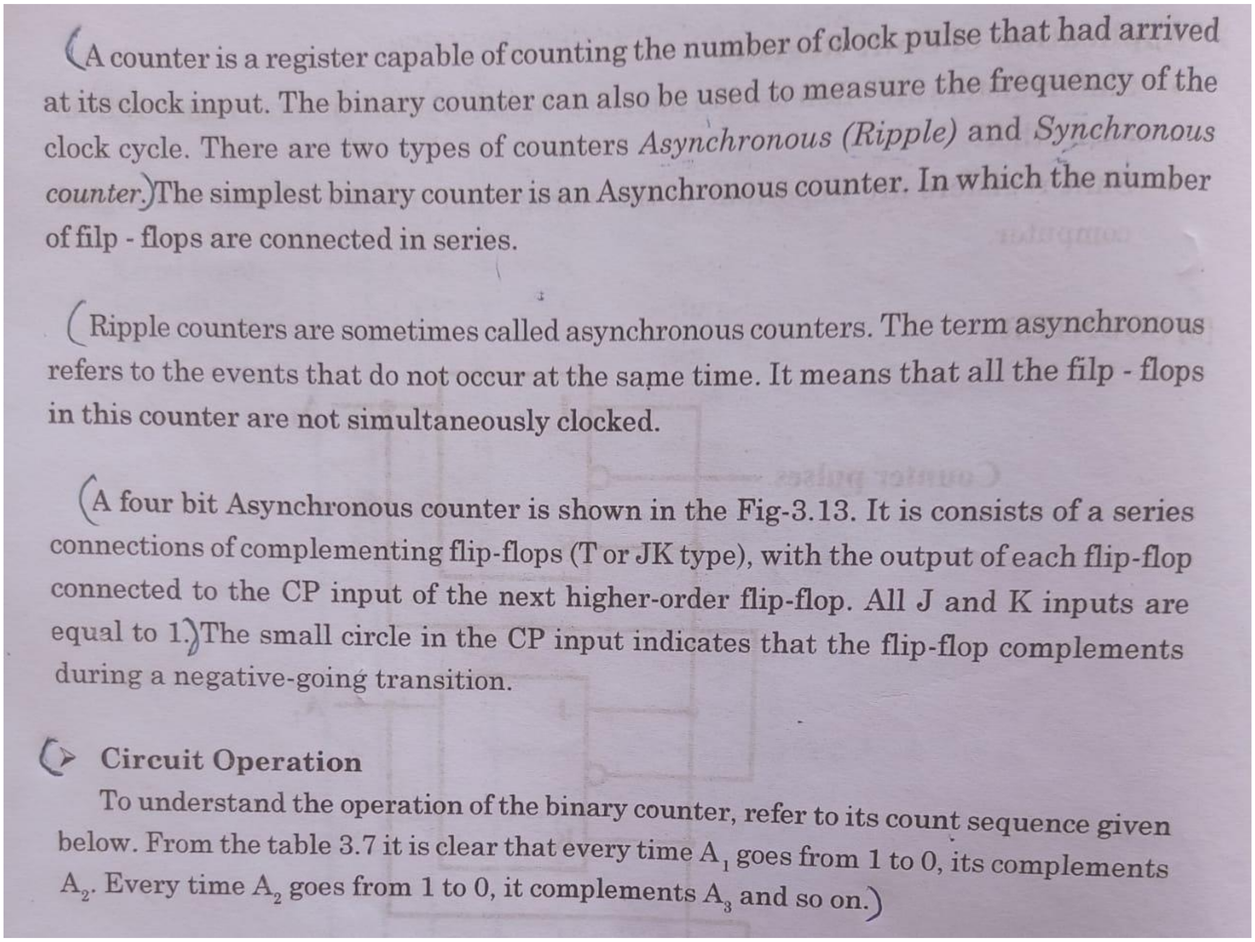


Fig-3.13 : 4-bit Binary Ripple Counter



(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.				

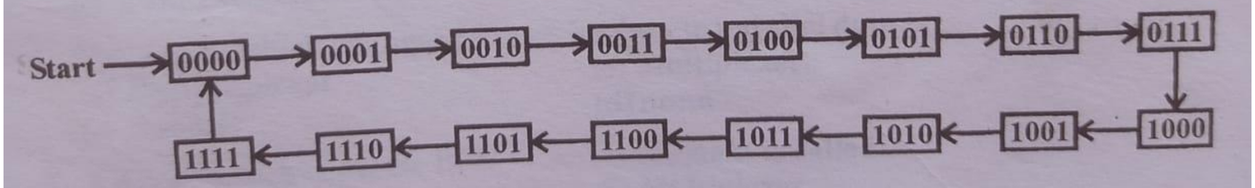


Fig. 3.14 : State Diagram of Ripple counter