Chapter One

General Introduction & Review

Introduction

Computer: machine (device) that can solve a problem by executing a set of instruction.

Computer Architecture: is really a system concept integrating H/W, S/W, algorithms and languages to perform large computation.

Computer Design: is concerned with the hardware design of the computer.

Computer Organization: is concerned with the way the hardware computer operates and the way they are connected together to form the computer system.

1. Computer Components:

Computer Hardware

- Digital computers use the binary number system, which has two digits, 0 and 1.
- A binary digit is called a bit.
- Bits are grouped together as bytes and words to form some type of representation within the computer.
- A sequence of instructions for the computer is known as program.

Computer Software

- Operating System
- Application Programs
- Languages (L.L.L , M.L.L , H.L.L)

H.W/ Identify the following?

- Compiler
- Interpreter
- Assembler

2. Computer Generation

1- Valves (Diodes) 1938 – 1953
2- Transistor 1952 – 1963
3- Integrated Circuits 1962 – 1975
4- A Microprocessor 1977 - …
5- Super computer

Parallel Processing: is a term used to denote a large class of technique that are used to provide simultaneous data processing tasks for the purpose of increasing the computation speed of computer system.

Microprocessors can be classified based on:

1) Their purpose:
    i) General purpose Processors
    ii) Special purpose Processors: this category includes:
        (a) Coprocessor: A coprocessor is a specially designed microprocessor, which can handle its particular function many times faster than the ordinary microprocessor.
        (b) I/O processor: It is a specially designed microprocessor having a local memory of its own, which is used to control I/O devices with minimum CPU involvement.
        (c) DSP: Digital Signal Processor, this processor is specially designed to process the analog signals into a digital form.
        (d) Embed microprocessor: many application for this type of processors like washing machine.

2) Their size
   i) 4-bit Microprocessor
   ii) 8-bit Microprocessor
   iii) 16-bit Microprocessor
   iv) 32-bit Microprocessor
   v) 64-bit Microprocessor

3) Their type
   i) **CSIC** - Complex Instruction Set Microprocessor
   ii) **RISC** - Reduced Instruction Set Microprocessors
   iii) **ASIC**- Application Specific Integrated Circuit

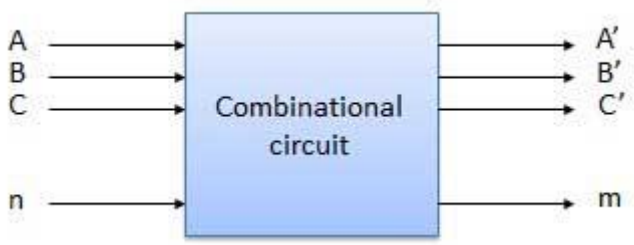
3. Digital Logic

3.1 Combinational circuit

Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and de-multiplexer. Some of the characteristics of combinational circuits are following:

- The output of combinational circuit at any instant of time depends only on the levels present at input terminals.
- The combinational circuits do not use any memory. The previous state of input does not have any effect on the present state of the circuit.
- A combinational circuit can have an n number of inputs and m number of outputs.

Block diagram



The following section elaborates a few important combinational circuits:

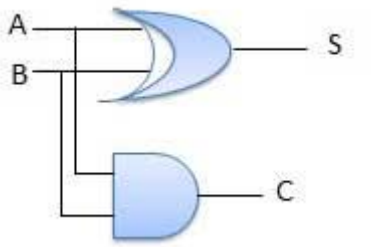
1. Half Adder

Half adder is a combinational logic circuit with two inputs and two outputs. The half adder circuit is designed to add two single bit binary numbers A and B. It is the basic building block for addition of two **single** bit numbers. This circuit has two outputs **carry** and **sum**.

Block diagram of Half Adder

Inputs		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

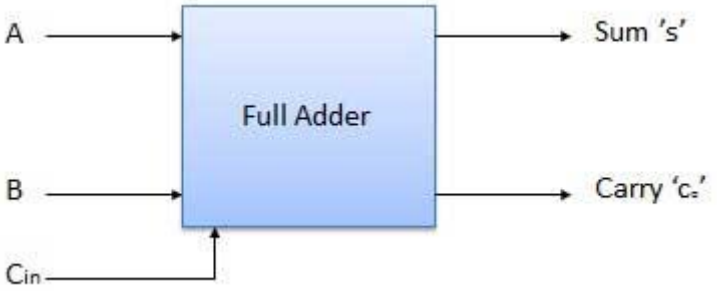
Truth Table



Circuit Diagram

2. Full Adder

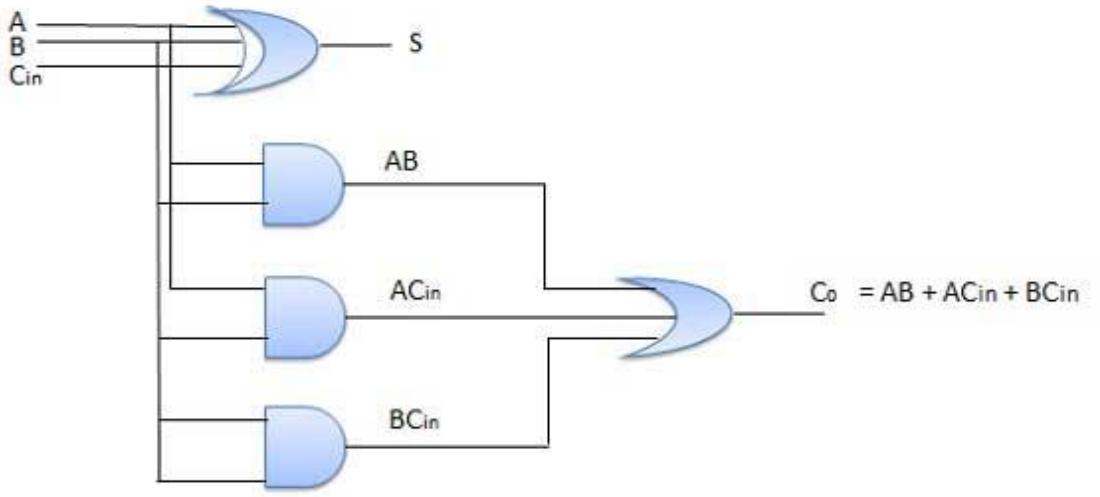
Full adder is developed to overcome the drawback of Half Adder circuit. It can add two one-bit numbers A and B, and carry c. The full adder is a three input and two output combinational circuit.



Block Diagram

Inputs			Output	
A	B	Cin	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Truth Table



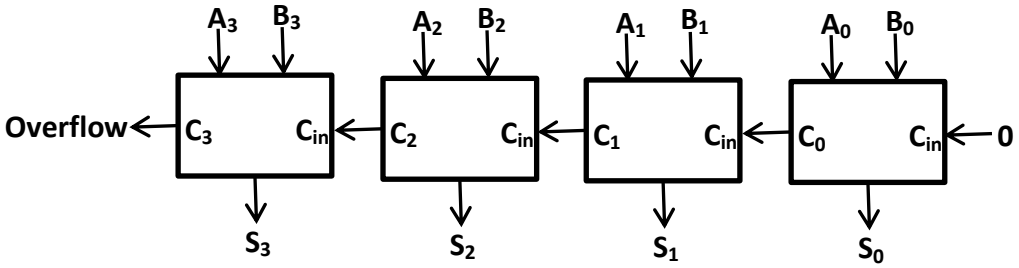
Circuit Diagram

3. N-Bit Parallel Adder

The Full Adder is capable of adding only two single digit binary number along with a carry input. But in practical we need to add binary numbers which are much longer than just one bit. To add two n-bit binary numbers we need to use the n-bit parallel adder. It uses a number of full adders in cascade. The carry output of the previous full adder is connected to carry input of the next full adder.

- 4 Bit Parallel Adder

  In the block diagram, A_0 and B_0 represent the LSB of the four-bit words A and B. Hence Full Adder-0 is the lowest stage. Hence its C_{in} has been permanently made 0. The rest of the connections are exactly same as those of n-bit parallel adder is shown in fig. The four-bit parallel adder is a very common logic circuit.

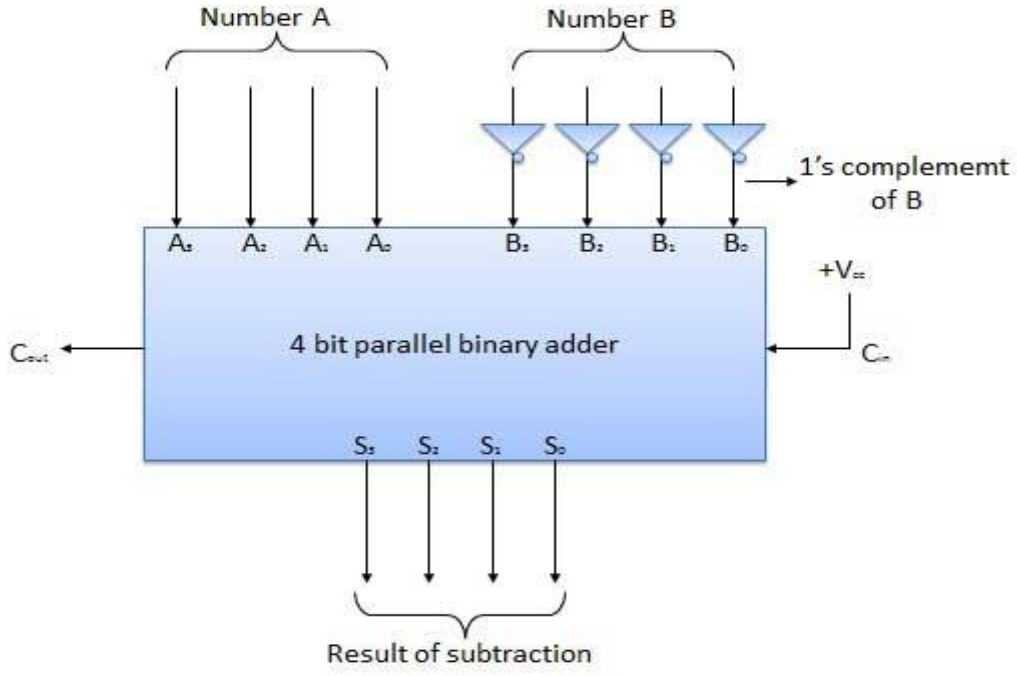


4. N-Bit Parallel Subtractors

The subtraction can be carried out by taking the 1's or 2's complement of the number to be subtracted. For example we can perform the subtraction (A-B) by adding either 1's or 2's complement of B to A. That means we can use a binary adder to perform the binary subtraction.

- 4 Bit Parallel Subtractor

  The number to be subtracted (B) is first passed through inverters to obtain its 1's complement. The 4-bit adder then adds A and 2's complement of B to produce the subtraction. S_3 S_2 S_1 S_0represents the result of binary subtraction (A-B) and carry output C_{out} represents the polarity of the result. If A > B then C_{out} = 0 and the result of binary form (A-B). Else C_{out} = 1 and the result is in the 2's complement form.

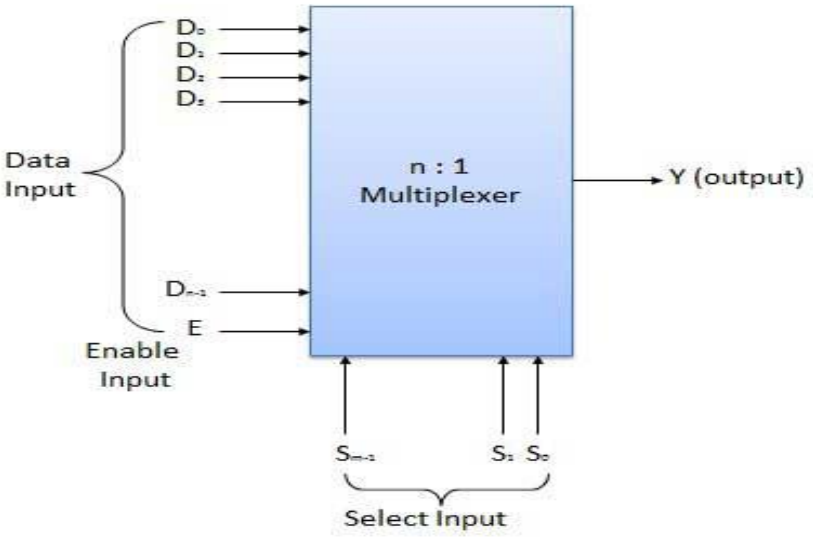


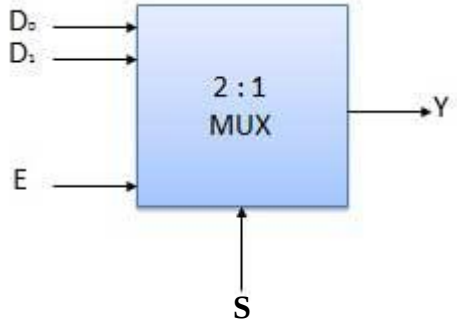
5. Multiplexers

Multiplexer is a ***special type of combinational circuit***. There are ***n-data*** inputs, ***one output*** and ***m*** select inputs with $2^m = n$. It is a digital circuit which selects one of the n data inputs and routes it to the output. The selection of one of the n inputs is done by the selected inputs.

Depending on the digital code applied at the selected inputs, one out of n data sources is selected and transmitted to the single output Y. E is called the enable input which is useful for the cascading. It is generally an active low terminal that means it will perform the required operation when it is low. Multiplexers come in multiple variations:

2: 1 multiplexer; 4: 1 multiplexer; 16: 1 multiplexer; or 32 : 1 multiplexer



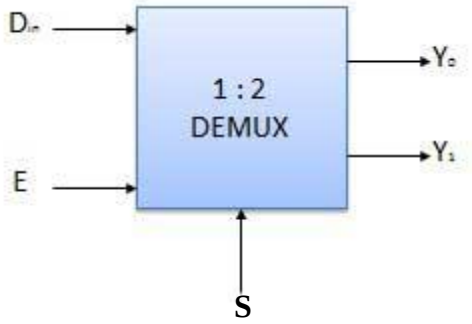


Enable	Select	Output
E	S	Y
0	x	0
1	0	D₀
1	1	D₁

x = Don't care

6. De-multiplexers

A de-multiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. It has only one input, n outputs, m select input. At a time only one output line is selected by the select lines and the input is transmitted to the selected output line. A de-multiplexer is equivalent to a single pole multiple way switch as shown in fig. De-multiplexers come in multiple variations. 1 : 2 demultiplexer.; 1 : 4 demultiplexer. ; 1 : 16 demultiplexer; 1 : 32 demultiplexer

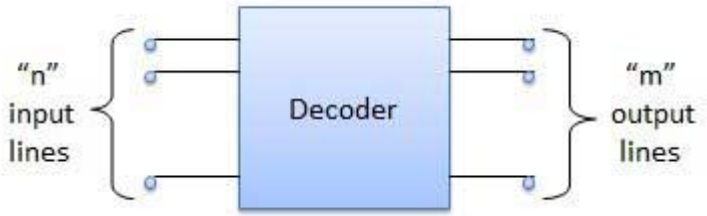


Enable	Select	Output	
E	S	Y0	Y1
0	x	0	0
1	0	0	Din
1	1	Din	0

x = Don't care

7. Decoder

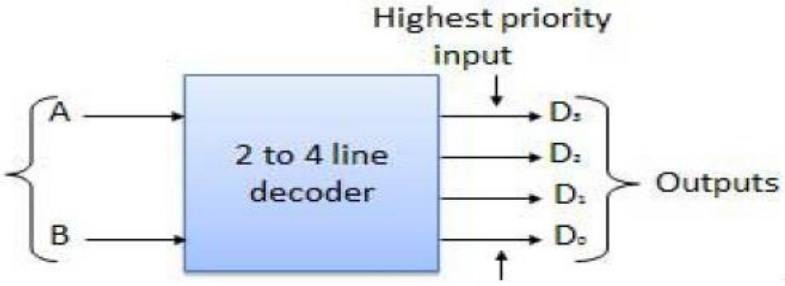
A decoder is a combinational circuit. It has **n** input and to a maximum **m = 2^n** outputs. Decoder is identical to a de-multiplexer without any data input. It performs operations which are exactly opposite to those of an encoder.



Examples of Decoders are: Code converters and BCD to seven segment decoders.

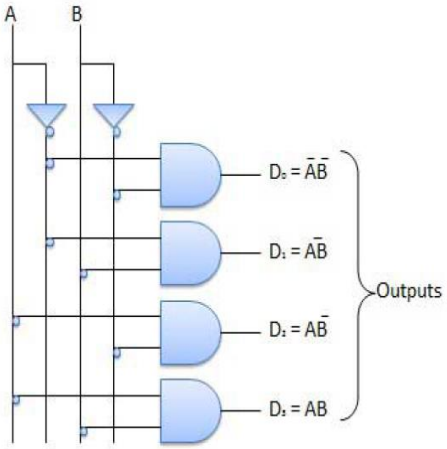
- 2 to 4 Line Decoder

  The block diagram of 2 to 4 line decoder is shown in the fig. A and B are the two inputs where D through D are the four outputs. Truth table explains the operations of a decoder. It shows that each output is 1 for only a specific combination of inputs.



Inputs		Output			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1

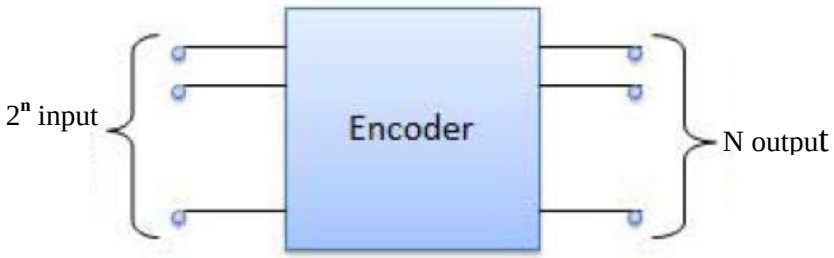
Truth Table



Logic Circuit

8. Encoder

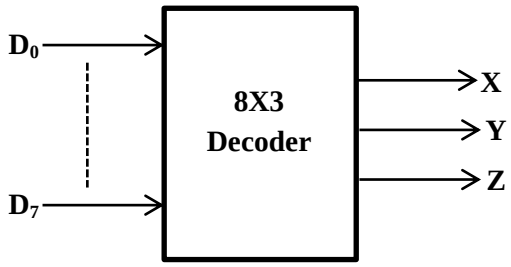
Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder. An encoder has (2^n) number of input lines and **n** number of output lines. An encoder produces an **n** bit binary code corresponding to the digital input number. The encoder accepts an 2^n input digital word and converts it into an **n** bit another digital word.



Examples of Encoders are: Decimal to BCD encoder; Octal to binary encoder; Hexadecimal to binary encoder; etc.

As an example:

Let's consider **Octal to Binary** decoder. As shown in the following figure, an octal-to-binary decoder takes 8 input lines and generates 3 output lines.



Truth Table –

D7	D6	D5	D4	D3	D2	D1	D0	X	Y	Z
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

As seen from the truth table, the output is 000 when D0 is active; 001 when D1 is active; 010 when D2 is active and so on.

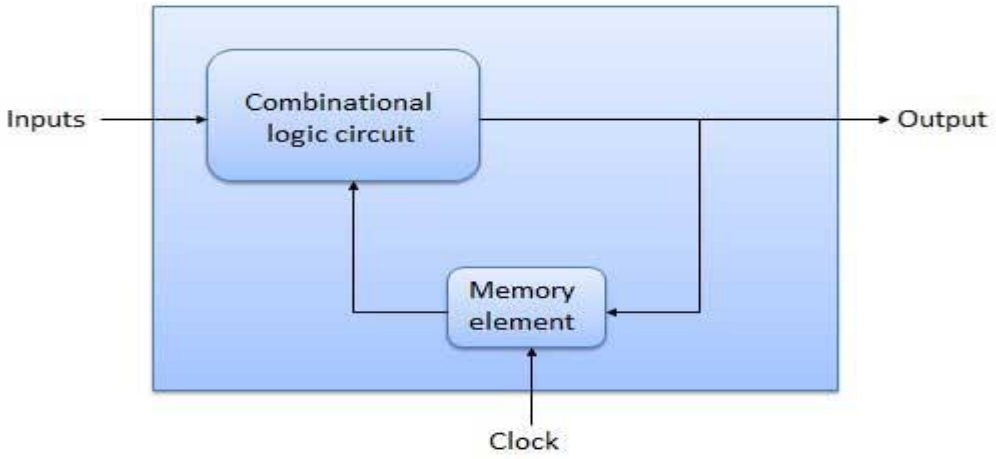
H.W/ Give the internal logic gate for the 1/4 DeMax?

H.W/ Design a 8/3 Encoder with enable line?

H.W/ Design a full adder using 1) Half Adder circuit 2) Using 3/8 decoder and 2 OR gate 3) Using 8/1 Max's

4. Sequential Circuits

The ***combinational*** circuit does not use any memory. Hence the previous state of input does not have any effect on the present state of the circuit. But ***sequential*** circuit has memory so output can vary based on input. This type of circuits uses ***previous input, output, clock and a memory element***.



Block Diagram

Combinational circuits implement the essential functions of a digital computer.

- The current output of a sequential circuit depends not only on the current input, but also on the past history of inputs.
- the current output of a sequential circuit depends on the current input and the current state of that circuit.

- the sequential circuit makes use of combinational circuits.

Flip Flop

A flip flop is an electronic circuit with ***two stable states*** that can be used to ***store binary data***. The stored data can be changed by applying varying inputs.

The simplest form of sequential circuit is the flip-flop. There are a variety of flip-flops, all of which share two properties:

- The flip-flop is a bistable device. It exists in one of two states and, in the absence of input, remains in that state. Thus, the flip-flop can function as a 1-bit memory.
- The flip-flop has two outputs, which are always the complements of each other. These are generally labeled Q and $\bar{Q}$

Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. Flip-flops and latches are used as **data storage** elements. It is the basic storage element in sequential logic. But first, let's clarify the difference between a latch and a flip-flop.

- **THE S–R LATCH**

Figure (1) shows a common configuration known as the S–R flip-flop or S–R latch. The circuit has two inputs, S (Set) and R (Reset), and two outputs, Q and consists of two NOR gates connected in a feedback arrangement.

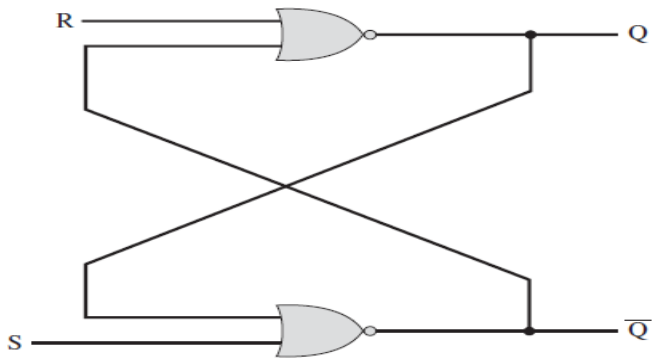


Figure (1): S-R Latch.

The S–R latch can be defined with a table similar to a truth table, called a ***characteristic table***, which shows the next state (Q_{n+1}) or states of a sequential circuit as a function of current states (Q_n) and inputs.

S-R Truth Table

S	R	Q_{n+1}
0	0	No Change (Q_n)
0	1	0
1	0	1
1	1	Not allowed

characteristic table

S	R	Q_n	Q_{n+1}
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	-
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	-

- **CLOCKED S–R FLIP-FLOP**

The output of the S–R latch changes, after a brief time delay, in response to a change in the input. This is referred to as asynchronous operation. More typically, events in the digital computer are synchronized to a clock pulse, so that changes occur only when a

clock pulse occurs. Figure (2) shows this arrangement. This device is referred to as a *clocked S–R flip-flop*.

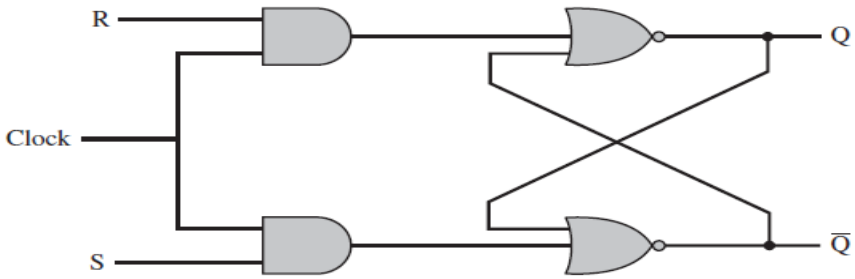
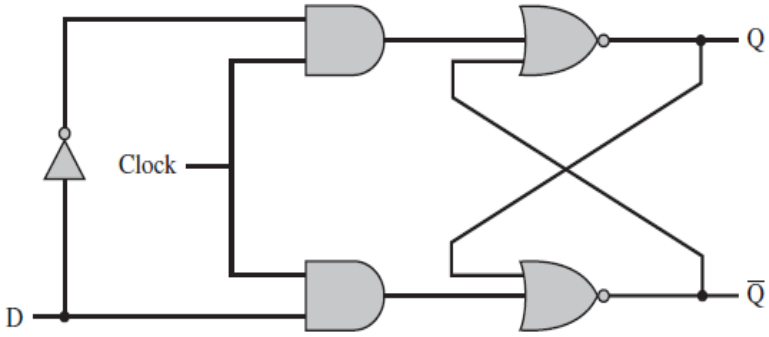


Figure (2): Clocked S-R Flip Flop.

- **D FLIP-FLOP**

One problem with S–R flip-flop is that the condition (R = 1, S = 1) must be avoided. One way to do this is to allow just a single input. The D flip-flop accomplishes this. Figure (3) shows a gate implementation and the characteristic table of the D flip-flop. By using an inverter, the non-clock inputs to the two AND gates are guaranteed to be the opposite of each other.



D	Q_{n+1}
0	0
1	1

Figure (3): D-Flip Flop.

The D flip-flop is sometimes referred to as the data flip-flop because it is, in effect, storage for one bit of data. The output of the D flip-flop is always equal to the most recent value applied to the input.

- **J–K FLIP-FLOP**

Another useful flip-flop is the J–K flip-flop. Like the S–R flip-flop, it has two inputs. However, in this case all possible combinations of input values are valid. Figure (4) shows a gate implementation of the J–K flip-flop.

J	K	Q_{n+1}
0	0	No Change (Q_n)
0	1	0
1	0	1
1	1	Toggle ($\overline{Q_n}$)

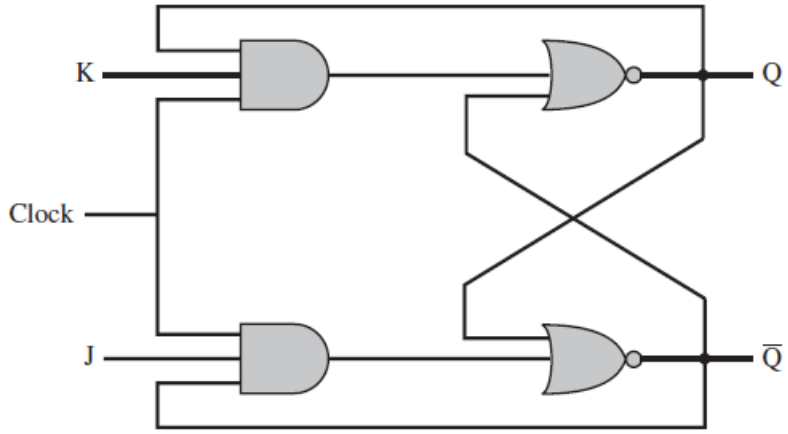
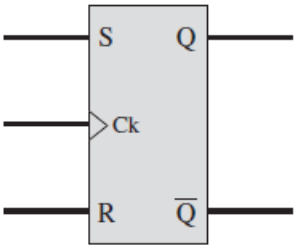
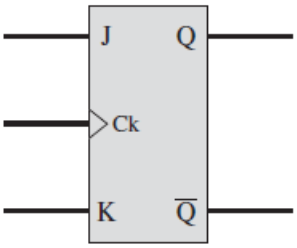
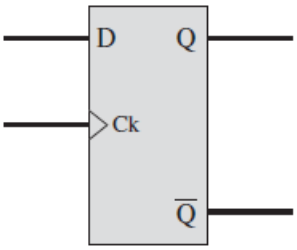


Figure (4): J-K Flip Flop.

The following table shows the basic block diagram of S-R, J-K, and D Flip Flops with corresponding their truth table.

Name	Graphical Symbol	Truth Table
S–R		S R Q_{n+1} 0 0 Q_n 0 1 0 1 0 1 1 1 –
J–K		J K Q_{n+1} 0 0 Q_n 0 1 0 1 0 1 1 1 $\overline{Q_n}$
D		D Q_{n+1} 0 0 1 1