University of Technology

الجامعة التكنولوجية

Computer Science Department

قسم علوم الحاسوب

Computer Organization & Logic Design

تركيب الحاسوب والتصميم المنطقي

Dr. Saed Ridha

م.د. سعيد رضا

cs.uotechnology.edu.iq

Part One: Computer Organization

1.Computer Organization

Computer: - electronic device that accepts input, stores large quantities of data, execute complex instructions which direct it to perform mathematical and logical operations and outputs the answers in a human readable form. (See fig. 1)

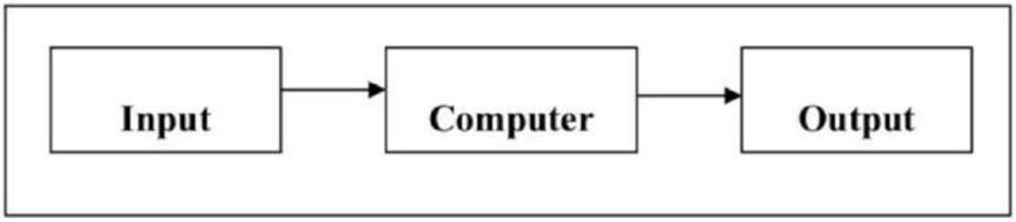


Fig (1) simple model of a computer

Advantages of computer system:

1- Store and retrieve large quantities of data.

2-The speed is faster than in any other form of data processing.

3-A single computer can perform a wide variety of activities as directed by a set of instructions (program).

4-Once data and instructions are fed into the computer, processing is continuous with a minimum of human intervention.

5-Data and programs may be stored inside the computer indefinite and be retrieved quickly.

6- Accuracy is greater than any other system.

2. Computer History

The Computer generation:

1- The first generation from 1946 to 1958 used electronic valves and frequent breakdowns and a rise in temperature due to the large size and weight. Use complex programming language.

2-The second generation from 1958 to 1964 used transistors instead valves, small size, low cost, and high speed. Use high programming language.

3- The third generation from 1965 to 1970 used complete circuit electronic, high speed, accuracy operations, and uses more users. Use high programming language.

4- The fourth generation from 1971 to 1980 used complete circuit electronic involve large number of transistors, small size, high speed in save data and information.

5- The fifth generation from 1980 to 1997 use complete circuit electronic very large and very high speed. As personal computer (PC), supper computer, and use artificial intelligent.

3. Computer Classification

Types of computers on the basis of Technology

•Analog Computers: Analog Computer is a computing device that works on continuous range of values. The results given by the analog computers will only be approximate since they deal with quantities that vary continuously. It generally deals with physical variables such as voltage, pressure, temperature, speed, etc.

•Digital Computers A digital computer operates on digital data such as numbers. It uses binary number system in which there are only two digits 0 and 1 .

The digital computer is designed using digital circuits in which there are two levels for an input or output signal . These two levels are known as logic 0 and logic 1 .Digital Computers can give more accurate and faster results.

•Hybrid Computers A hybrid computer combines the desirable features of analog and digital computers. It is mostly used for automatic operations of complicated physical processes and machines.

4. Computer Structure

Computer system is made of two main parts: -

1-Hardware: refers to the physical components of the computer such as:

- Keyboard, memory, printer…

2-Software: refers to programs, languages, and instructions that make the hardware work for us.

Main components of hardware:

The basic components of a computer system are: (see fig. 2)
1-Input unit
2-Central processing unit: - which consists of control unit. , Arithmetic and logic unit, and Register.
3- Output unit.
4- Memory unit (internal memory).
5- External storage.

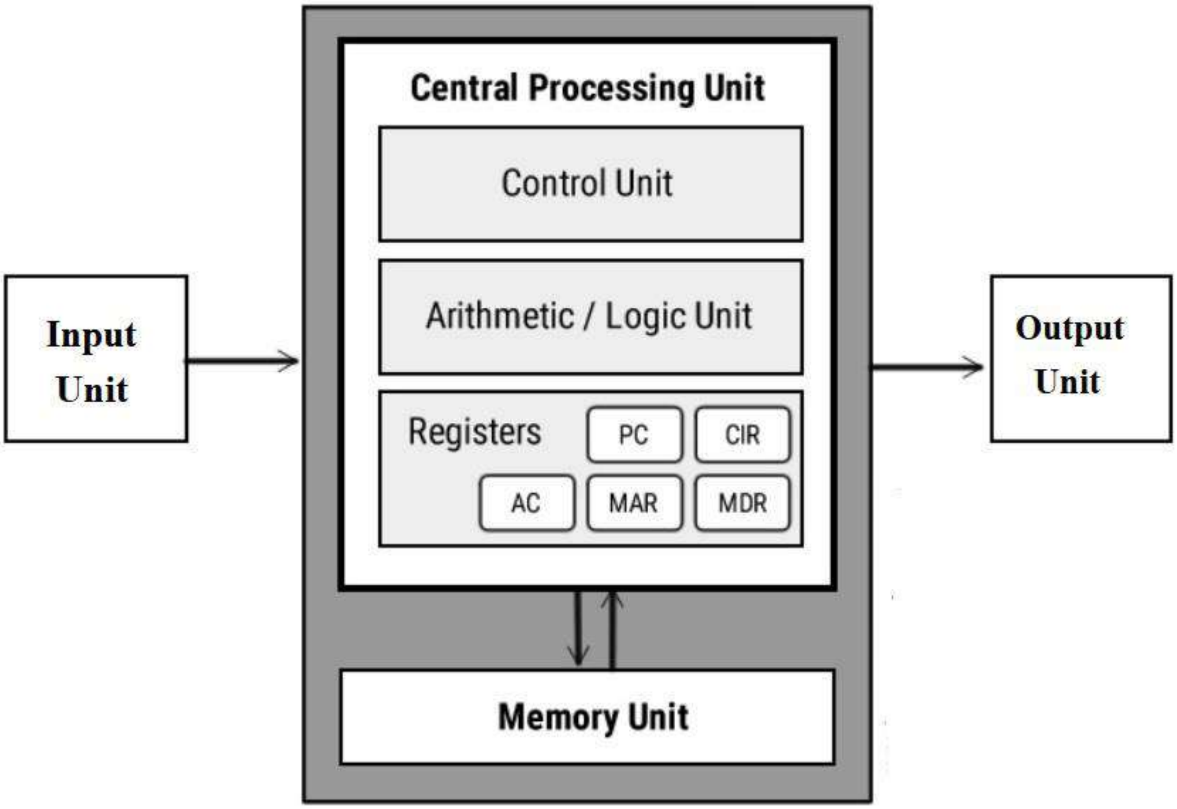


Fig (2) the logical structure of a computer

1-Input unit: the input unit of a computer system accepts data, convert it into electrical impulses that are sent in to internal memory or to the central processing unit (CPU) where can be processed. Such as keyboard, mouse and other devices.

2-Central processing unit (CPU): The brain of any computer system is the CPU, which is sometime called "Processor" or "Microprocessor" in personal computer. The CPU supervises and controls all of the peripheral equipment; perform arithmetic and makes logical decisions. The CPU is responsible for includes the data movement computations and logical operation necessary to convert data into meaningful information.

It is divided into three sections:

2.1 Arithmetic and Logic unit (ALU).

2.2 Control unit.

2.3 Register.

2.1 Arithmetic and Logic unit (ALU): -

Perform the processing of data including arithmetic operations such as addition, subtraction, multiplication, division and logic operations including comparison (ex. A<B) and sorting.

2.2 Control Unit.:

- Direct and coordinates all units of the computer to execute program steps.
- Direct and coordinates all operations of the computer systems.

These operations include: -

1- Control to the input and output devices.
2- Entry and retrieval of information from memory.
3- Routing of information between the memory and the arithmetic and logic unit(ALU).

Control unit automatically coordinates the operation of the entire computer system, although the control unit does not performed any actual processing on the data, It acts as a central nervous system uses to send control signal to other units.

2.3 Register:

Register are devices capable of storing information, receiving data from other areas within the computer and transferring information as directed by the control unit, it is used for temporary storage of data or instruction and the most important register are: -

a- Program counter (PC): It contains the address of the next instruction to be executed.

b- Instruction Register (IR): It contains the instruction being executed.

c- Address Register (AR): holds the address of memory location.

d- Data Register (DR): Holds data that is being transferred to or from memory.

e- Accumulator (AC): Where intermediate arithmetic and logic results are stored.

3- Output unit: Output units are instruments of interpretation and communication between human and computer that let you see. the result of the commands you enter, the most common output device are a display screen (monitor), printer or other device that let you see.

The CPU executes each Instruction in a series of steps: -

1- Fetch the next instruction from memory to IR.

2- Changes the program counter to point to the following instruction.

3- Determine the type of the instruction to be fetched.

4- IF the instruction uses data in memory determines where they are.

5- Fetch the data into the internal CPU register.

6- Execute the instruction.

7- Store the result in the proper place.

8- Go to step 1 to being executing the following instruction.

4- Memory units:

The memory is the part of the computer that holds information (data and Instruction) for processing, fig3 represents the classification of computer memory.

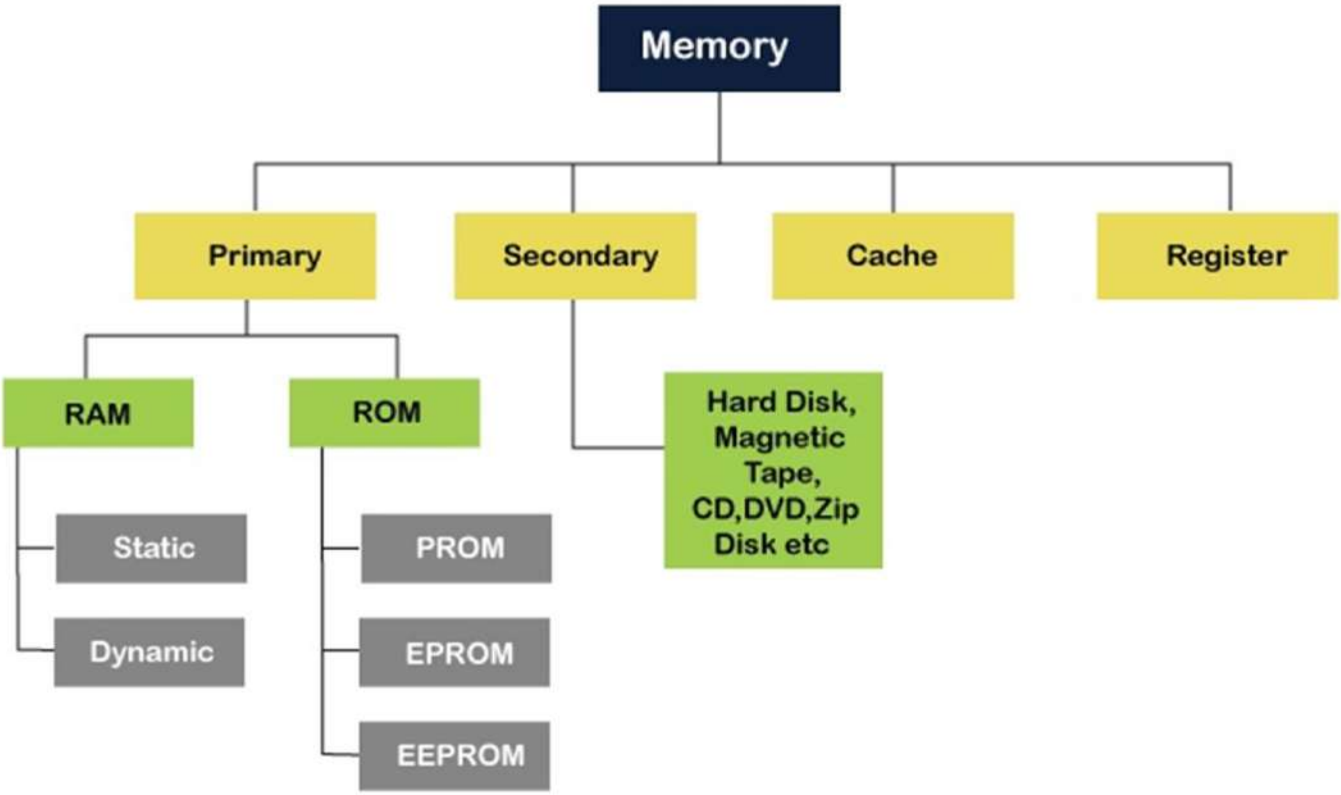


Fig (3): Classification of Computer Memory.

4.1 Primary or Main Memory

Primary memory is also known as the computer system's main memory that communicates directly within the CPU, Main memory is used to kept programs or data when the processor is active to use them. The primary memory is further divided into two parts:

1. RAM (Random Access Memory)
2. ROM (Read Only Memory)

Random access memory (RAM)

Random Access Memory (RAM) is one of the faster types of main memory accessed directly by the CPU. It is the hardware in a computer device to temporarily store data, programs or program results. It is used to read/write data in memory until the machine is working. It is volatile, which means if a power failure occurs or the computer is turned off, the information stored in RAM will be lost. All data stored in computer memory can be read or accessed randomly at any time. The RAM can be either dynamic or static.

a- **Static RAM (SRAM):** is a type of RAM used to store static data in the memory. it mean the stored data will remain permanent stored as long as the power is supplied without the need for periodically rewriting the data in to memory.

b- **Dynamic RAM (DRAM):** is a type of RAM that is used for the dynamic storage of data in RAM. The stored data will not remain permanently stored even with power is applied unless the data are periodically rewritten in to memory; the later operation is called a refresh operation.

Read only memory (ROM)

Is read only memory which can be read from but not written on so that it is called anon-volatile memory, when the user turn the computer off the content of ROM are not changed, table1 show the difference between RAM and ROM , the type of ROM is:

a- Programmable Read Only Memory (PROM):

It is prepared by the maker and can be electrical programmed by the user; it cannot be erased and programmed again this means its content can never be changed.

b- Erasable Programmable Read Only Memory (EPROM):

The maker prepares it and can be electrical programmed by the user, it can be erasing (deleted) by exposure to ultraviolet light and programmed many times.

c- EEPROM (Electrically Erasable Programmable Read Only Memory): The EEROM is an electrically erasable and programmable read only memory used to erase stored data. It is also a non-volatile memory whose data cannot be erased or lost; even the power is turned off. In EEPROM, the stored data can be erased and reprogrammed up to 10 thousand times.

Table1: Difference between RAM and ROM:

Difference	Random Access Memory (RAM)	Read Only Memory (ROM)
Read/Write	Read and write memory	Only read memory
Use	Used to store data that has to be currently processed by CPU temporarily.	It is typically used to store firmware or microcode, which is used to initialize and control hardware components of the computer.
Speed	It is a high-speed memory.	It is much slower than the RAM.
Cost	RAM is expensive.	ROM is cheap .
Function	Used for temporary storage of data and instruction.	Used for permanent storage of data and instruction.
Data-Retention	Volatile memory: program and data erased when power off is.	Permanent memory (non-volatile): program and data remains intact even is power off.
Type	SRAM,DRAM	PROM, EPROM, EEPROM

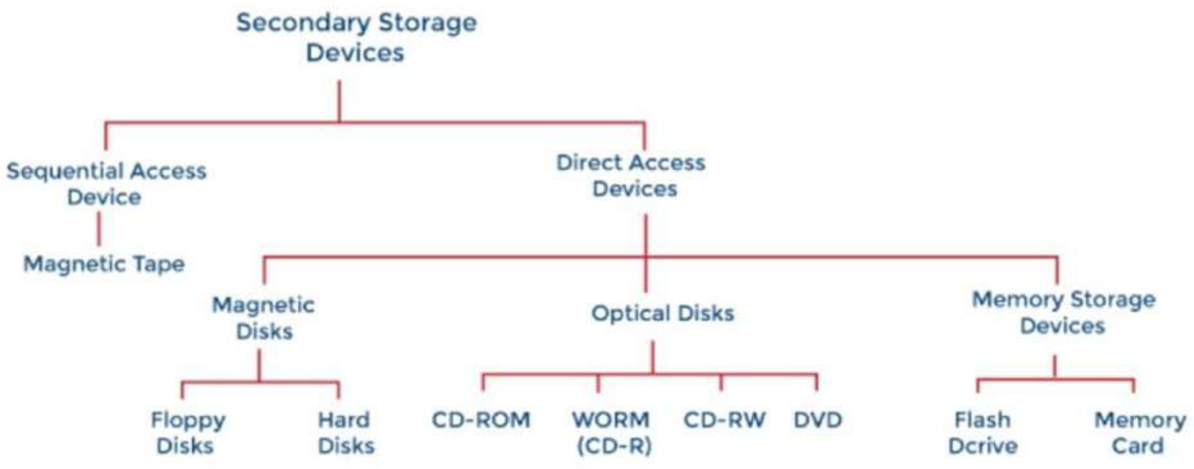
4.2 Secondary storage (External storage)

Secondary storage refers to the storage methods and technologies used for the long-term storage of non-critical data that doesn't need to be accessed as frequently as primary storage. The goal of secondary storage is to retain data until you overwrite or delete it.

The key features of secondary memory storage devices are:

1. Very high storage capacity.
2. Permanent storage (non-volatile), unless erased by user.
3. Relatively slower access.
4. Stores data and instructions that are not currently being used by CPU but may be required later for processing.
5. Cheapest among all memory.

Fig 4 shows the classification of commonly used secondary storage devices.



Fig(4):Classification of Secondary Storage Devices.

The classification of commonly used secondary storage devices:

1- Sequential storage Device It is a class of data storage devices that read stored data in a sequence as the magnetic tape in old computer system.

2- A direct-access storage device (DASD) is another name for secondary storage devices that store data in discrete locations with a unique address, such as hard disk drives, optical drives and most magnetic storage devices.

a. Magnetic Disk: A magnetic disk is a storage device that uses a magnetization process to write, rewrite and access data.

- **Magnetic Hard Disk: a** magnetic disk is a storage device that uses a magnetization process to write, rewrite and access data. Bits are stored in the magnetized in sports along concentric circles called **tracks.** The minimum quantity of information which can be transferred is a **sector.**

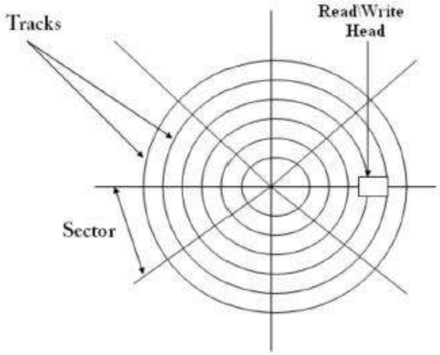


Fig (3) Hard Disk

- **Magnetic Floppy disk:** floppy disk is compose of a thin and flexible disk of a magnetic storage medium in a square or nearly square plastic enclosure lined with a fabric that removes dust particles from the spinning disk. The early floppy disks were 5.25 inch in diameter and were packaged in semi flexible jacket. . Floppy disks store digital data which can be read and written when the disk is inserted into a floppy disk drive (FDD) connected to or inside a computer or other device.

b. Optical Disk: an optical disk is any computer disk that uses optical storage techniques and technology to read and write data. It is a computer storage disk that stores data digitally and uses laser beams to read and write data. Such as (CD ,DVD and other)

c. Memory Storage Devices: A memory device includes USB drives, flash memory devices, SD and memory cards,

- **Flash Memories:** Flash memories are high-density read\write memories (high-density translates into large bit storage capacity) that are nonvolatile, which means that data can be stored indefinitely without power they are sometimes used in place of floppy or small.
- **A memory card** is an electronic data storage device used for storing digital information, typically using flash memory. These are commonly used in portable electronic devices, such as digital cameras, mobile phones, laptop computers, tablets, PDAs, portable media players, video game consoles, synthesizers, electronic keyboards and other devices

4.3 Cache Memory

It is a very high speed memory placed between RAM and CPU. It is storage buffer that stores the data that is used more and make them available to CPU when needed at fast rate. Cache memory stores copies of the data that used frequently by CPU from main memory (Ram) locations, so that they are immediately available to the CPU when needed**.**

5. Operating System`

1. operating system

An operating system is a program that acts as an intermediary between a user of a computer and the computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs. The primary goal of an operating system is thus to make the computer system convenient to use. A secondary goal is to use the computer hardware in an efficient manner. An operating system is an important part of almost every computer system. A computer system can be divided roughly into four components: the hardware, the operating system, the applications programs, and the users.

2. Network:

A network is a set of devices (often referred to as nodes) connected by media links. A node can be a computer, printer, or any other device capable of sending and /or receiving data generated by other nodes on the network. The links connecting the devices are often called communication channels.

Type of the network:

a- LAN (Local Area Network).

b- MAN (Metropolitan Area Network).

c- WAN (Wide Area Network).

3. **Internet:**

   Internet is the world-wide super network of computer networks that links computers around the world.

   URL: Each day when we use the Internet to check our mail online, visit a web page or browse an FTP folder, we use our browser. And while there can be a great number of Internet browsers out there, each of them offering different functions and boasting a different design, one thing that unites all of them is the fact that they are built with a single purpose - to handle URLs. **The URL** Each file available on the World Wide Web can be identified and accessed through its corresponding URL. Standing for Uniform Resource Locator, a URL represents the global web address of documents, including web pages or image files, and programs such as CGI applications or Java applets. Its main mission is to identify the location of a document or a program available on the web and specify the mechanism for accessing it through a web browser.

Number Systems Operation:

1- Decimal Numbers.

2- Binary Numbers.

3- Octal Numbers.

4- Hexadecimal Numbers.

1- Decimal Numbers: In the decimal number system each of the ten digits (10digits), 0 through 9 (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9).

Decimal weight … …10^4 10^3 10^2 10^1 10^0. 10^{-1} 10^{-2} 10^{-3} ….

Example (1): **$(345)_{10}$**

$300+40+5=10^2*3+10^1*4+10^0*5=345= (345)_{10}$

↓ ↓ ↓

3 4 5

Example (2): **$23.5 = (23.5)_{10}$**

$2*10^1 + 3*10^0 +5*10^{-1}$ $= 20+3+0.5=23.5$

Where $10^0 =1$

2- Binary Numbers: The binary number system its two digits a base- two system. The two binary digits (bits) are 1 and 0 (1,0).

Binary weight 2^3 2^2 2^1 2^0

Weight value 8 4 2 1

A- Binary – to – Decimal Conversion:

***Binary number 1101101 where $2^0 =1$**

1 1 0 1 1 0 1

2^6 2^5 2^4 2^3 2^2 2^1 $2^0 = 2^6 *1+ 2^5*1+2^4*0+2^3*1+2^2*1+2^1*0+2^0*1$

$= 64+32+0+8+4+0+1=96+13=109$ □$(109)_{10}$

***The fractional binary number 0.1011 0. 1 0**

1 1

2^{-1} 2^{-2} 2^{-3} 2^{-4} = $1*2^{-1}+0*2^{-2}+1*2^{-3}+1*2^{-4}$ =

0.5+0+0.125+0.0625=0.6875 ⟹ $(0.6875)_{10}$

B- Decimal – to – Binary Conversion:

1. **Convert a decimal whole number to binary using the repeated division – by – 2 method.**
2. **Convert a decimal fraction to binary using the repeated Multiplication – by – 2 method.**

Example (1):

Number $(58)_{10}$ ==== ⟹ $(111010)_2$

2	58		mod LSB
2	29	=	0
2	14	=	1
2	7	=	0
2	3	=	1
2	1	=	1
	0	=	1

↑ ======= ⟹ $(111010)_2$

MSB

Example (2):

Number $(0.3125)_{10}$ ====== ⟹ $(0101)_2$

MSB	carry	
		0.3125*2
	0	0.6250*2
	1	0.2500*2
	0	0.5000*2
	1	0.0000
LSB ↓		

$(0101)_2$

3- *Octal Numbers:* The octal number system is composed of eight digits, which are 0, 1, 2, 3, 4, 5, 6, and 7.

To count above 7, begin another column and start over: 10, 11, 12, 13, 14, 15, 16, and 17.

20, 21, 22, 23, 24, 25, 26, and 27.

30, 31, … … … … …37.

A- Octal – to – Decimal conversion: **Weight**

… … 8^3 8^2 8^1 8^0

Octal number 2374 $\Longrightarrow$ **$(1276)_{10}$**

xample:

$$(2374)_8 = 2*8^3+3*8^2+7*8^1+4*8^0$$

$$= 2*512+3*64+7*8+4*1$$

$$= 1024+192+56+4$$

$$= (1276)_{10}$$

B- Decimal – to – Octal Conversion:

Example:

Decimal number $(359)_{10}$ $\Longrightarrow$ **$(547)_8$**

8	**359**		**mod LSB** ↑	
8	**44**	= $\boxed{7}$		
8	**5**	= $\boxed{4}$		$\Longrightarrow$ **$(547)_8$**
	0	= $\boxed{5}$		
			MSB	

C- Octal – to – Binary Conversion:

Octal digit can be represented by a 3-bit binary number.

Octal digit binary

0	1	2	3	4	5	6	7
000	001	010	011	100	101	110	111

***Example*s:**

$(25)_8$	$(140)_8$
$(2 \quad 5)_8$	$(1 \quad 4 \quad 0)_8$
$(010101)_2$	$(001100000)_2$

D- Binary – to – Octal Conversion:

Conversion binary number to octal number is start with right – most group of three bits and moving from right to left.

***Example*s:**

$(110101)_2$	$(101111001)_2$
110 101	101 111 001
6 5	5 7 1
$(6 \quad 5)_8$	$(5 \quad 7 \quad 1)_8$
$(65)_8$	$(571)_8$

4- ***Hexadecimal Numbers:*** The hexadecimal number system has a base of sixteen; it is composed of 16 digits and alphabeticcharacters.

Decimal	**Binary**	**Hexadecimal**
0	**0000**	**0**
1	**0001**	**1**
2	**0010**	**2**
3	**0011**	**3**
4	**0100**	**4**
5	**0101**	**5**
6	**0110**	**6**
7	**0111**	**7**
8	**1000**	**8**
9	**1001**	**9**
10	**1010**	**A**
11	**1011**	**B**
12	**1100**	**C**
13	**1101**	**D**
14	**1110**	**E**
15	**1111**	**F**

A- Binary – to – Hexadecimal conversion:

4-bit groups, starting at the right-most bit.

***Example*: (1100101001010111)$_2$ ======= (CA57)$_{16}$1100**

	1010	0101	0111

B- Hexadecimal – to – Binary Conversion:

***Example*: (10A4)$_{16}$ ========= (1000010100100)$_2$**

1	0	A	4
0001	0000	1010	0100

C- *Hexadecimal – to –Decimal Conversion:* **By to method**

*** First method:**

***Example*: (A85)$_{16}$ ==== (2693)$_{10}$**

1- **Convert to binary number.**

2- **Convert from binary number to decimal number.**

A	8	5
1010	1000	0101 =

$$2^{11}*1+2^{10}*0+2^{9}*1+2^{8}*0+2^{7}*1+2^{6}*0+2^{5}*0+2^{4}*0+2^{3}*0+2^{2}*1+2^{1}*0+2^{0}*1=$$

$$2^{11}+2^{9}+2^{7}+2^{2}+2^{0}=2048+512+128+4+1=2693= (2693)_{10}$$

*** Second method:**

***Example*: (E5)$_{16}$ ======== (229)$_{10}$**

$$(E5)_{16}=E*16^{1}+5*16^{0}=14*16+5*1=224+5=229= (229)_{10}$$

D- ***Decimal – to – Hexadecimal Conversion:***

Example**:** **Convert the decimal number 650 to hexadecimal by repeated division by 16.**

$(650)_{10}$ ===== $(28A)_{16}$

Mod LSD

16 | 650

16 | 40 ====== A

16 | 2 ====== 8 **MSD 2 8 A LSD = $(28A)_8$**

| 0 ====== 2

MSD

Arithmetic Number System

1-Binary Arithmetic:

1- Binary Addition.
2- Binary Subtraction.
3- Binary Multiplication.
4- Binary Division.

1- ***Binary Addition:*** The four basic rules for adding binary digits (bits) are as follows.

0+0=0 Sum of 0 with a carry 0

0+1=1 Sum of 1 with a carry 0

1+0=1 Sum of 1 with a carry 0

1+1=1 0 Sum of 0 with a carry 1

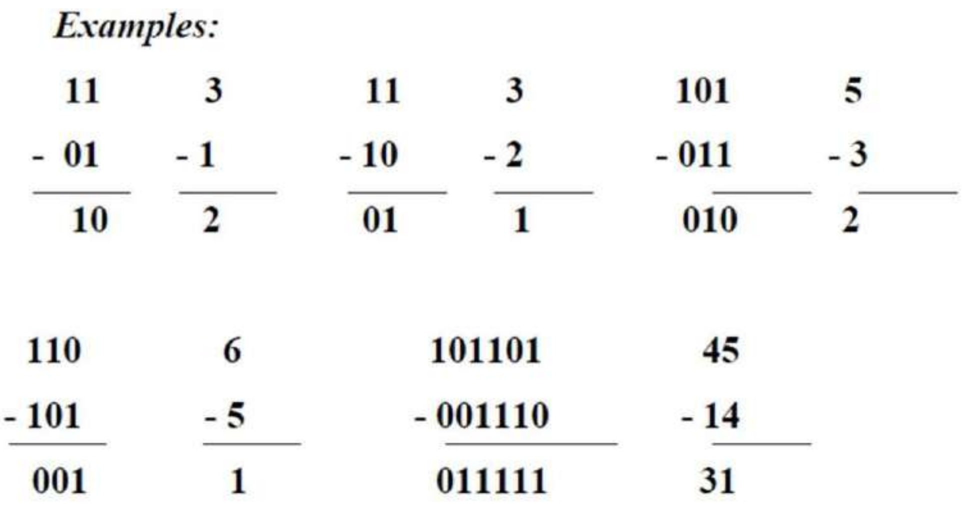
Examples

$$\begin{array}{r} 110 \\ +\ 100 \\ \hline 1010 \end{array} \qquad \begin{array}{r} 6 \\ +4 \\ \hline 10 \end{array} \qquad \begin{array}{r} 111 \\ +011 \\ \hline 1010 \end{array} \qquad \begin{array}{r} 7 \\ +3 \\ \hline 10 \end{array}$$

$$\begin{array}{r} 1111 \\ +\ 1100 \\ \hline 11011 \end{array} \qquad \begin{array}{r} 15 \\ +12 \\ \hline 27 \end{array}$$

2- ***Binary Subtraction:*** The four basic rules for subtracting are as follows.

0-0=0
1-1=0
1-0=1
0-1=1 0-1 with a borrow of 1



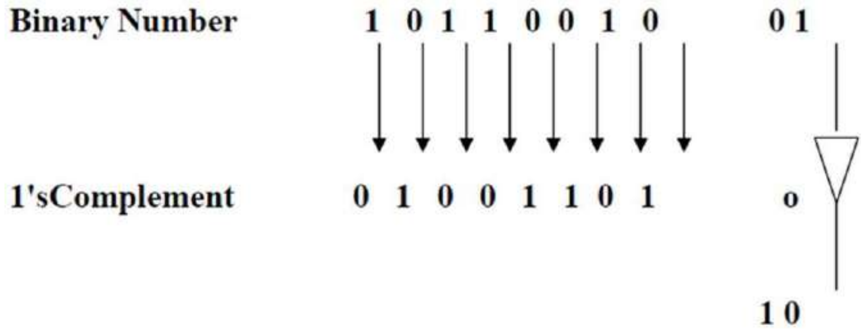
Examples:

```
  11      3        11      3        101       5
- 01    - 1      - 10    - 2      - 011     - 3
-----   -----    -----   -----    -----    -----
  10      2        01      1        010       2

 110        6          101101        45
- 101      - 5        - 001110      - 14
-----     -----       --------     -----
 001        1          011111        31
```

3- 1's And 2's Complement of Binary Number:

The 1's complement and the 2's complement of binary number are important because they permit the representation of negative numbers.



Binary Number 1 0 1 1 0 0 1 0

1'sComplement 0 1 0 0 1 1 0 1

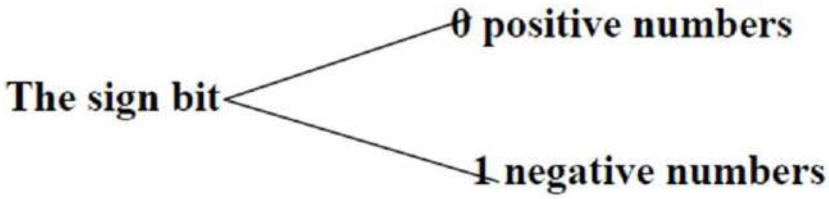
2's Complement of a binary number is found by adding 1 to the LSB of the 1's Complement.
2's Complement= (1's Complement) +1

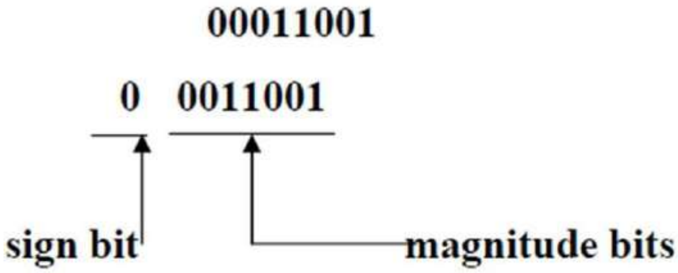
```
Binary number    10110010
1'scomplement    01001101
Add1                   + 1
--------------------------

2's complement   01001110
```

In decimal number complement such as:
0====9
7====2
6====3
9====0
4====5
1====8

Signed Numbers: Signed binary number consists of both sign and magnitude information.

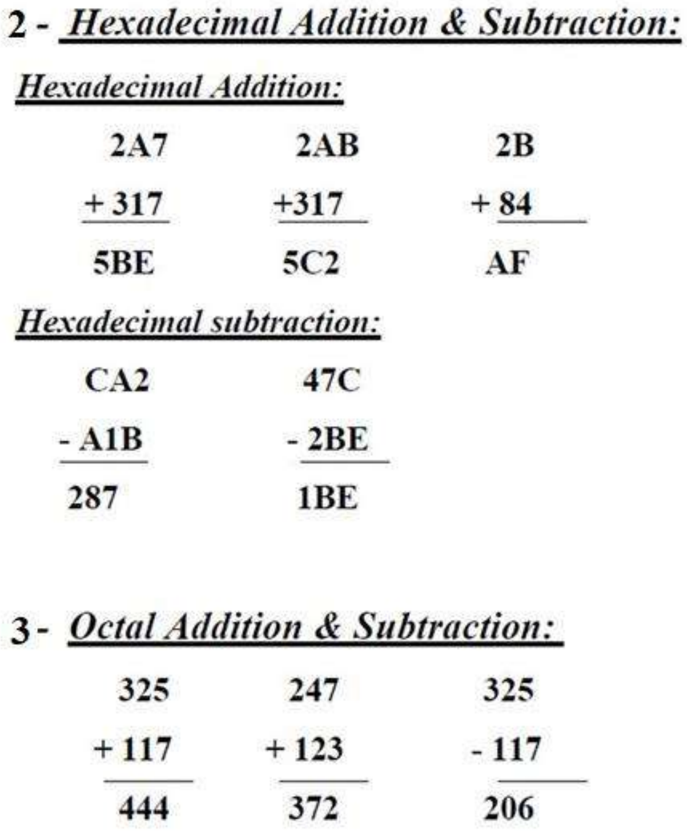




Example: **Express the decimal number - 39 as an 8-bit number in the sign-magnitude, 1's complement, and 2's complement forms.**

Solution:
1- Write the 8-bit number for +39 00100111
2- 1's complement 11011000
3- Add 1 1

Sign bit negative 1 1011001 = - 39



2 - *Hexadecimal Addition & Subtraction:*

Hexadecimal Addition:

```
   2A7        2AB        2B
 + 317       +317      + 84
   5BE        5C2        AF
```

Hexadecimal subtraction:

```
   CA2        47C
 - A1B      - 2BE
   287        1BE
```

3 - *Octal Addition & Subtraction:*

```
   325        247        325
 + 117      + 123      - 117
   444        372        206
```

Binary Coded Decimal (BCD):

Binary coded decimal means that each decimal digit, 0 through 9, is represented by a binary code of four bits.

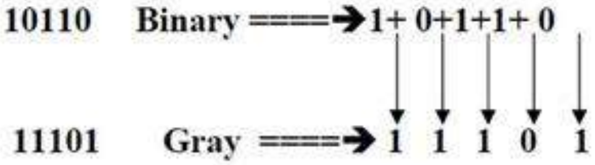
The 8 4 2 1 (BCD) Code:

The 8 4 2 1 code is a type of (BCD) code. The 8 4 2 1 indicates the binary weights of the four bits (2^3, 2^2, 2^1, 2^0).

4 -*The Gray Code:*

Example:

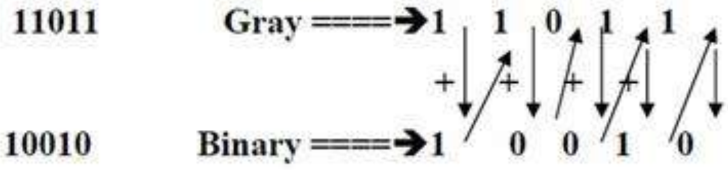
Convert binary to Gray



10110 Binary ====➔1+ 0+1+1+ 0

11101 Gray ====➔ 1 1 1 0 1

Convert Binary to Gray

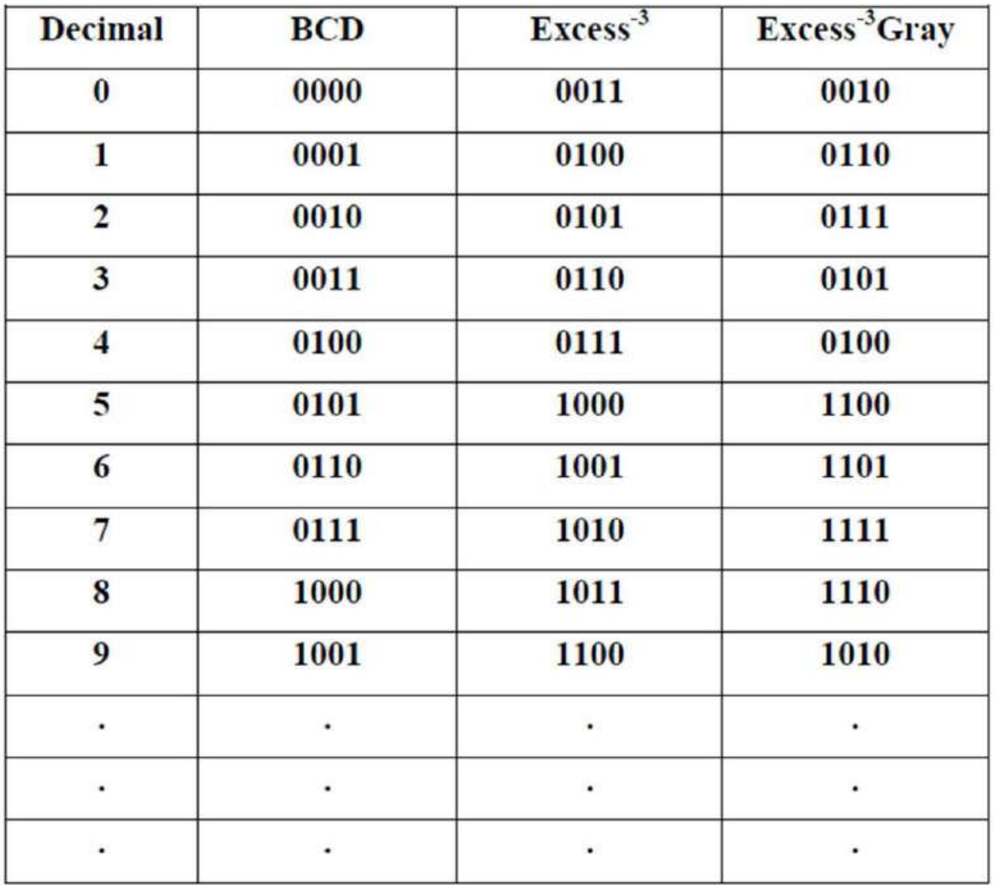


11011 Gray ====➔1 1 0 1 1

10010 Binary ====➔1 0 0 1 0

Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	.	.
.	.	.
.	.	.

5- Excess-3 Code: Addition three to any number in decimal number or binary number such as in table.

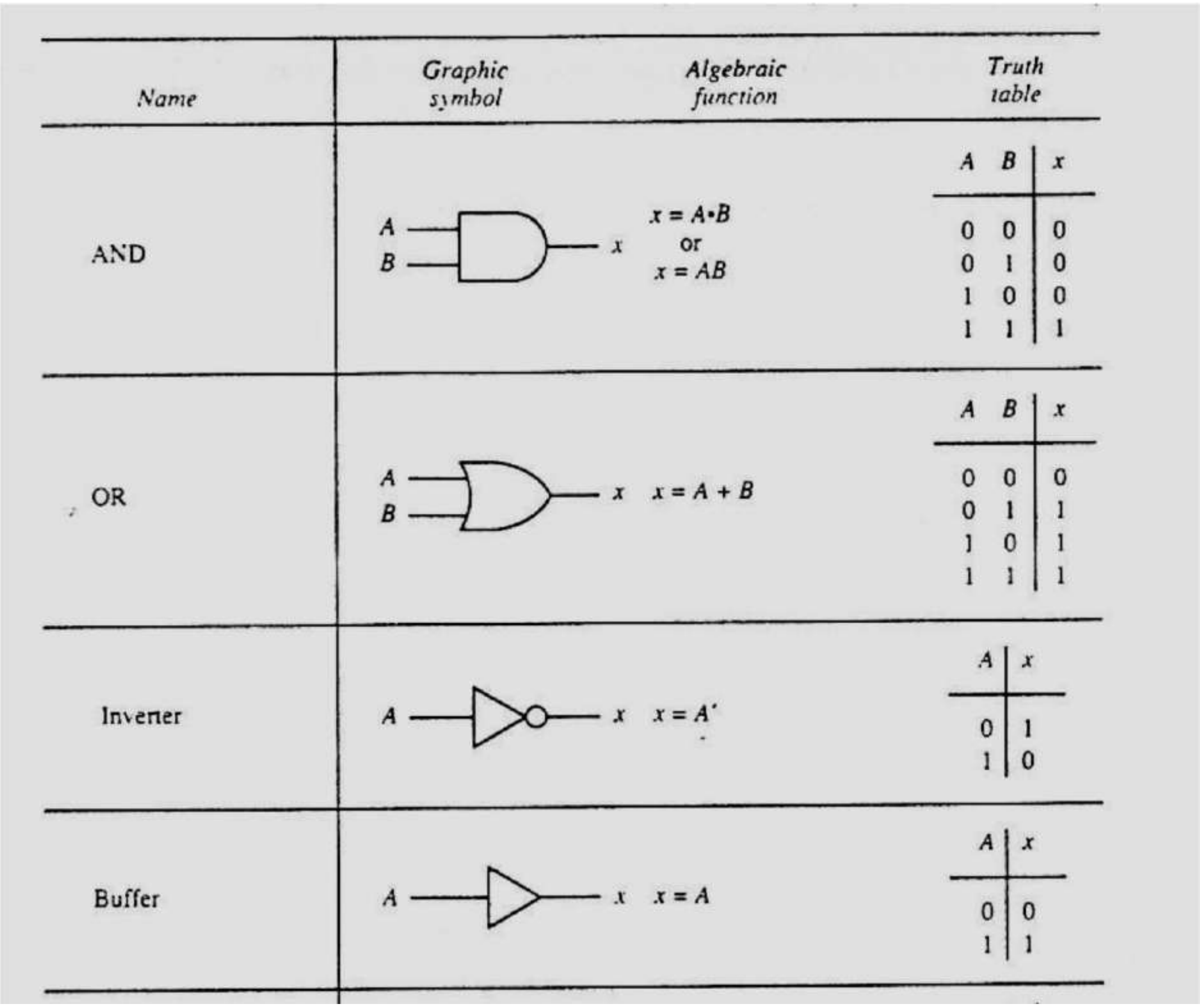


Decimal	BCD	Excess^{-3}	Excess^{-3}Gray
0	0000	0011	0010
1	0001	0100	0110
2	0010	0101	0111
3	0011	0110	0101
4	0100	0111	0100
5	0101	1000	1100
6	0110	1001	1101
7	0111	1010	1111
8	1000	1011	1110
9	1001	1100	1010
.	.	.	.
.	.	.	.
.	.	.	.

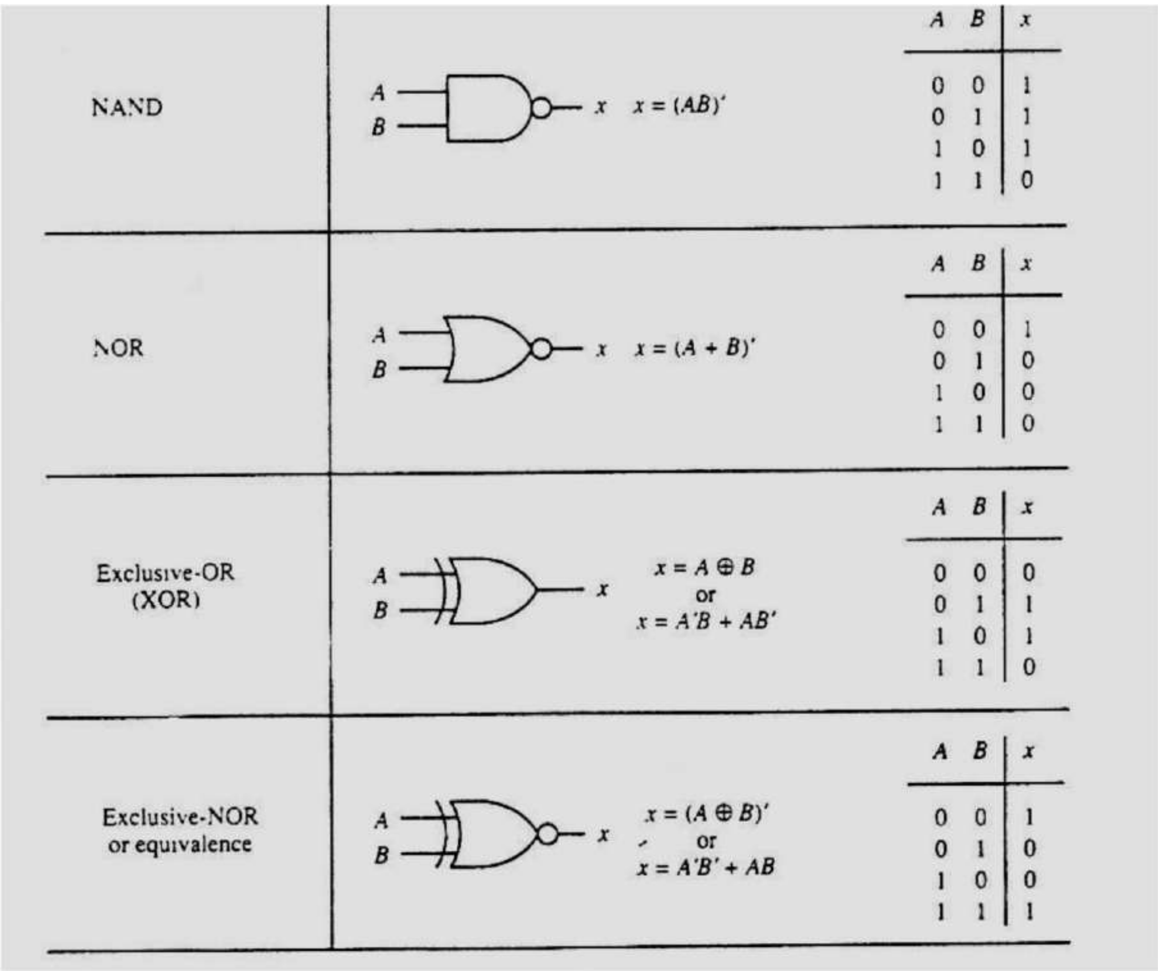
Part Two: Logic Design

Logic Gats:

1- Set of Gats

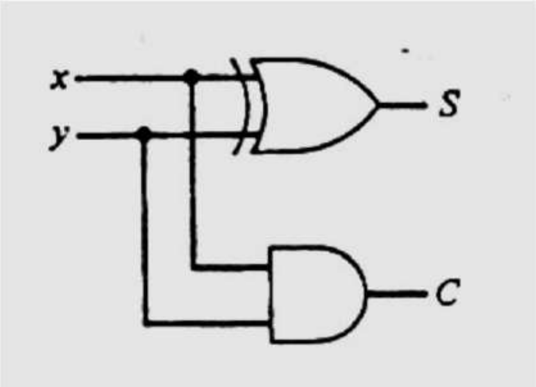


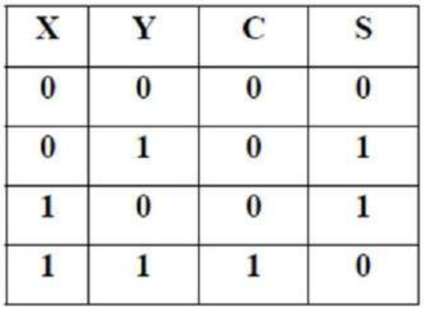
Name	Graphic symbol	Algebraic function	Truth table		
AND		$x = A \cdot B$ or $x = AB$	A	B	x
			0	0	0
			0	1	0
			1	0	0
			1	1	1
OR		$x = A + B$	A	B	x
			0	0	0
			0	1	1
			1	0	1
			1	1	1
Inverter		$x = A'$	A		x
			0		1
			1		0
Buffer		$x = A$	A		x
			0		0
			1		1



		A	B	x
NAND	$x = (AB)'$	0	0	1
		0	1	1
		1	0	1
		1	1	0
		A	B	x
NOR	$x = (A + B)'$	0	0	1
		0	1	0
		1	0	0
		1	1	0
		A	B	x
Exclusive-OR (XOR)	$x = A \oplus B$ or $x = A'B + AB'$	0	0	0
		0	1	1
		1	0	1
		1	1	0
		A	B	x
Exclusive-NOR or equivalence	$x = (A \oplus B)'$ or $x = A'B' + AB$	0	0	1
		0	1	0
		1	0	0
		1	1	1

2- *Half – Adder:* The basic digital arithmetic circuit is the addition of two binary digits. Input variables of a half-adder call augends & addend bits. The output variables the sum & carry.





X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Figure (10-a) Logic diagram for half adder

Figure (10-b) Truth table for half adder

Half- Adder questions:

$S=\overline{X}Y+X\overline{Y}$

S=X (+)Y

C=X*Y

3-*Full-Adder:* A full - adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs &two outputs.

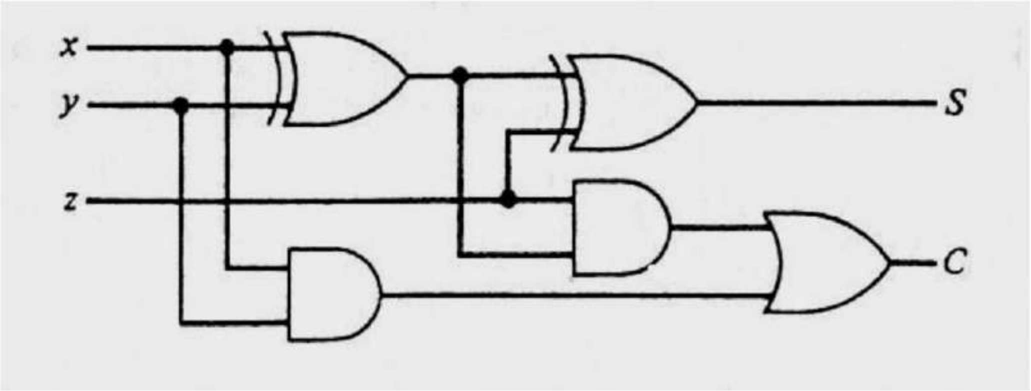


Figure (11) Logic diagram for full adder (Logic Diagram)

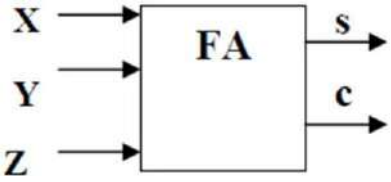


Figure (12) Block diagram for full adder

Inputs			Out puts	
X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure (13) Truth table for full adder

Full - Adder questions:

S=x (+) y (+) z

C=XY+ (XZ (+) YZ)

C=X*Y+ (X (+) Y) Z

Boolean Algebra &Logic Simplification:

1-Rules of Boolean algebra:

1- A+0=A

2- A+1=1

3- A*0=0

4- A*1=A

5- A+A=A

6- $A+\overline{A}=1$

7- A*A=A

8- $A*\overline{A}=0$

9- $\overline{\overline{A}}=A$ **======Demoragan's theorems**

10- A+BA=A

11- $A+\overline{A}B=A+B$

12- (A+B)(A+C)=A+BC

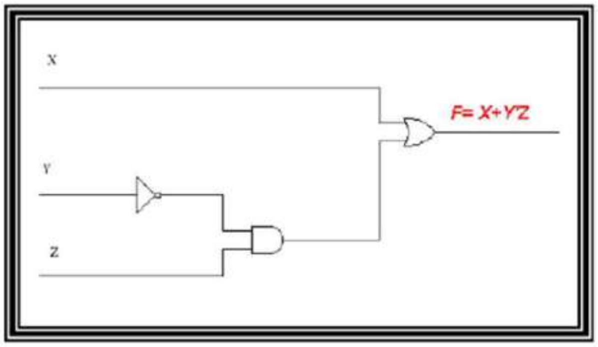
2- Examples:

Example 1:

F = X + ýz

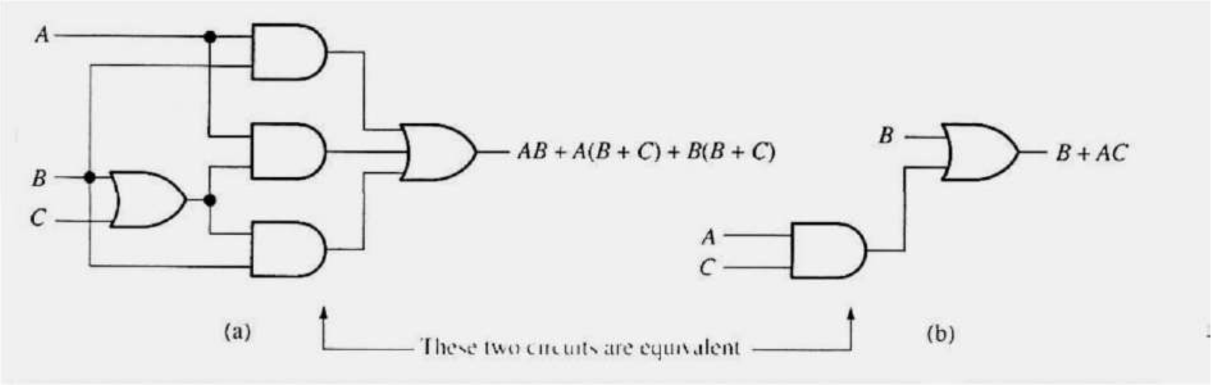
Determine the truth table and logic diagram

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



Example 2:

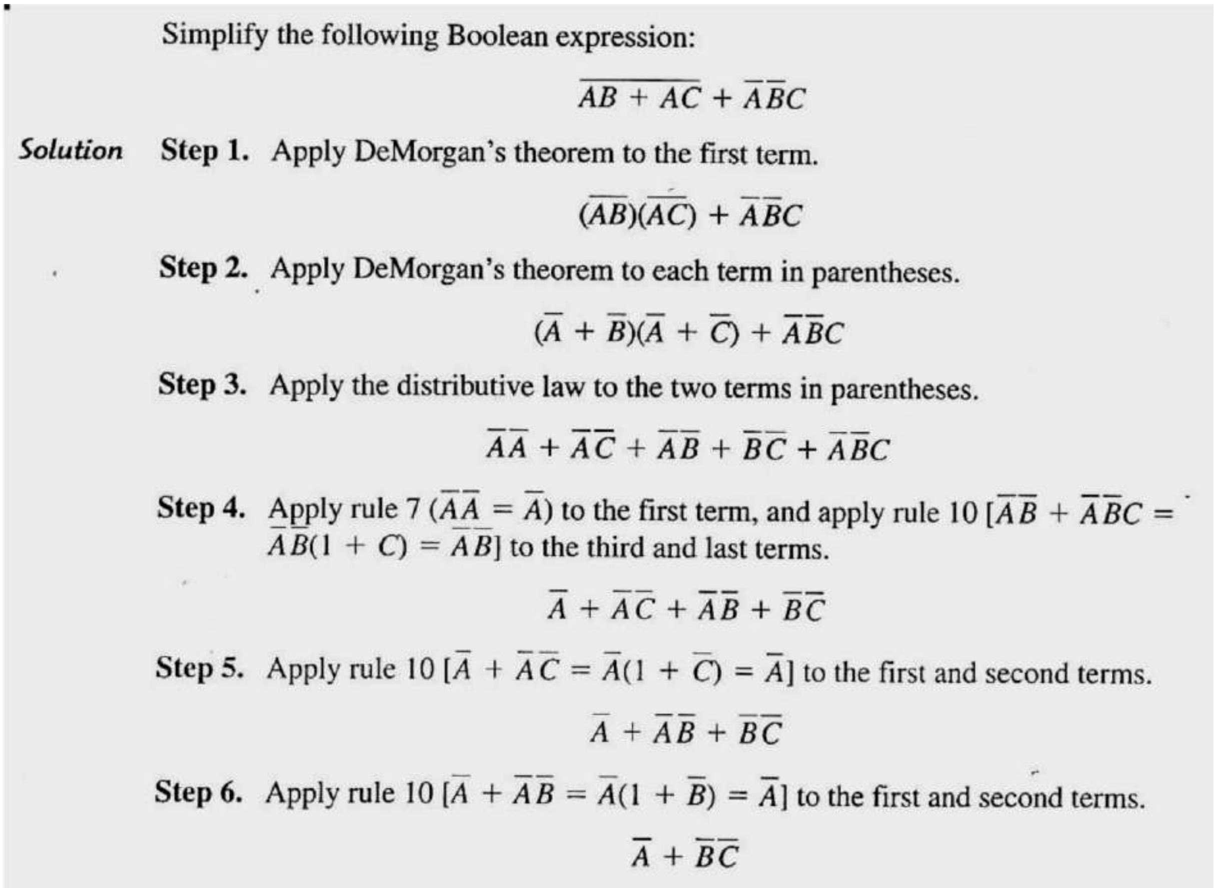
AB+ A (B+C)+ B(B+C)
1- AB+AB+AC+BB+BC
2- AB+AB+AC+B+BC
3- AB+AC+B+BC
4- AB+AC+B
5- B+AC



Example 3:

F=ABC+ABĆ+ĂC
F= AB(C+Ć) +ĂC
F= AB+ĂC

Example 4:



Simplify the following Boolean expression:

$$\overline{AB + AC} + \overline{A}\,\overline{B}C$$

Solution **Step 1.** Apply DeMorgan's theorem to the first term.

$$(\overline{AB})(\overline{AC}) + \overline{A}\,\overline{B}C$$

Step 2. Apply DeMorgan's theorem to each term in parentheses.

$$(\overline{A} + \overline{B})(\overline{A} + \overline{C}) + \overline{A}\,\overline{B}C$$

Step 3. Apply the distributive law to the two terms in parentheses.

$$\overline{A}\,\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$$

Step 4. Apply rule 7 ($\overline{A}\,\overline{A} = \overline{A}$) to the first term, and apply rule 10 [$\overline{A}\,\overline{B} + \overline{A}\,\overline{B}C = \overline{A}\,\overline{B}(1 + C) = \overline{A}\,\overline{B}$] to the third and last terms.

$$\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

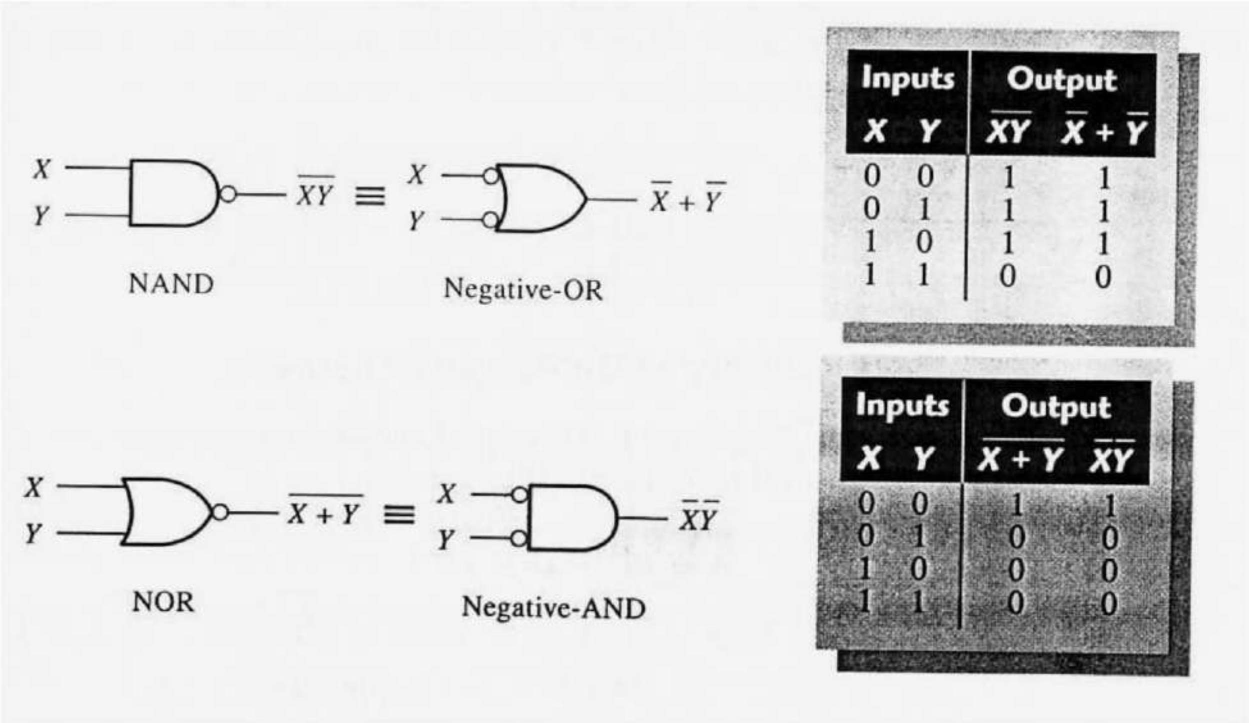
Step 5. Apply rule 10 [$\overline{A} + \overline{A}\,\overline{C} = \overline{A}(1 + \overline{C}) = \overline{A}$] to the first and second terms.

$$\overline{A} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

Step 6. Apply rule 10 [$\overline{A} + \overline{A}\,\overline{B} = \overline{A}(1 + \overline{B}) = \overline{A}$] to the first and second terms.

$$\overline{A} + \overline{B}\,\overline{C}$$

3- Demorgan's theorems:



Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X}+\overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Inputs		Output	
X	Y	$\overline{X+Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Example 1:

a- $\overline{\overline{(A+B)}+\overline{C}} = \overline{\overline{(A+B)}}\ \overline{\overline{C}} = (A+B)C$

b- $\overline{(\overline{A}+B)+CD} = \overline{(\overline{A}+B)}\ \underline{CD} = (\overline{\underline{A}}\ \overline{B})\ (\overline{C}+\overline{D}) = \overline{A}\ \overline{B}\ (C+\overline{D})$

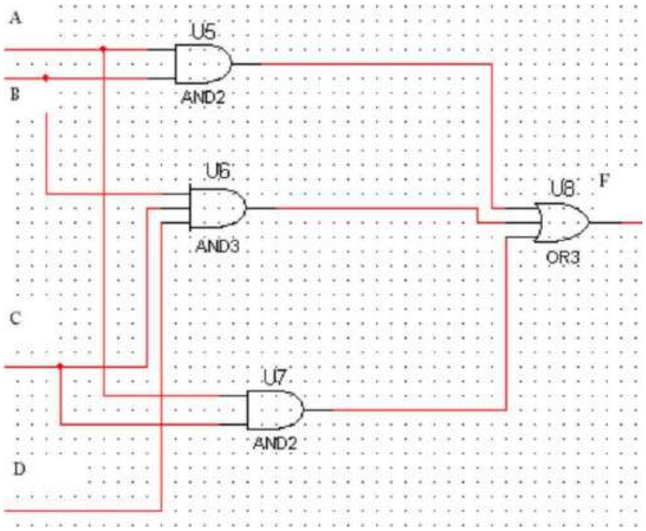
c- $\overline{(A+B)\ \overline{C}\ \ \overline{D}+E+\ \overline{F}} = (\overline{(A+B)\ \overline{C}\ \overline{D}})\ (\overline{E+\overline{F}})$

$= (A*\overline{B}+\overline{C}+D)*(E\ \overline{F})$

$= (\ A*\overline{B}+\overline{C}+D)\ E\ \overline{F}$

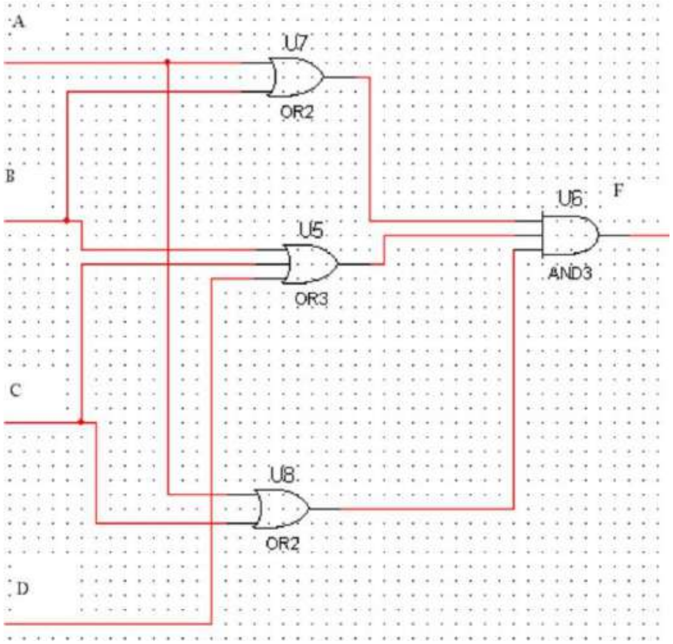
5- Sum – Of – Products (SOP):

X=AB+BCD+AC



6- Product – Of – Sum(POS):

(A+B)(B+C+D)(A+C)



Example: SOP

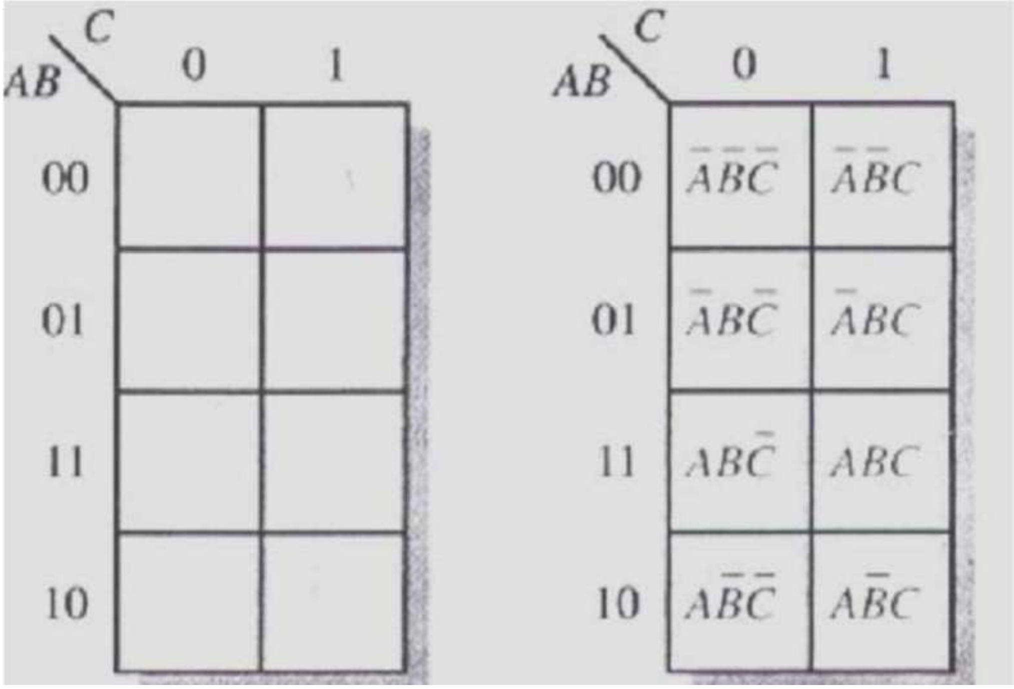
A	B	X	F
0	0	0	
0	1	1	$\overline{A}B$
1	0	1	$A\overline{B}$
1	1	0	

Example: POS

A	B	X	F
0	0	0	$\overline{A}+\overline{B}$
0	1	1	
1	0	1	
1	1	0	$A+B$

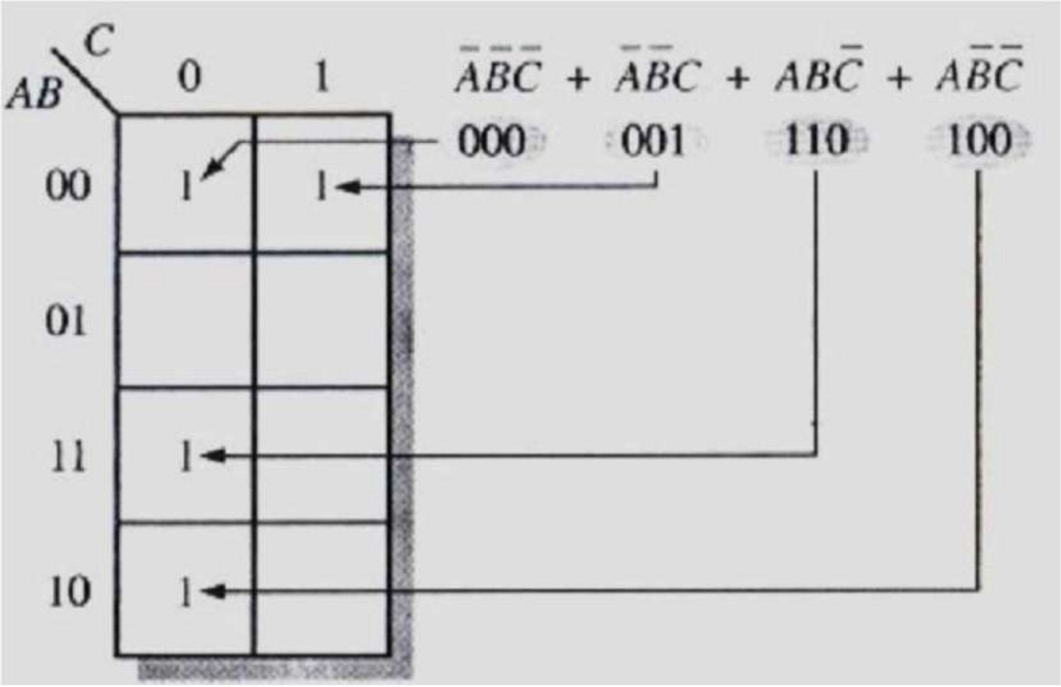
Karnaugh map:

1- Three – variable karnaugh map.



AB \ C	0	1
00		
01		
11		
10		

AB \ C	0	1
00	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$
01	$\bar{A}B\bar{C}$	$\bar{A}BC$
11	$AB\bar{C}$	ABC
10	$A\bar{B}\bar{C}$	$A\bar{B}C$

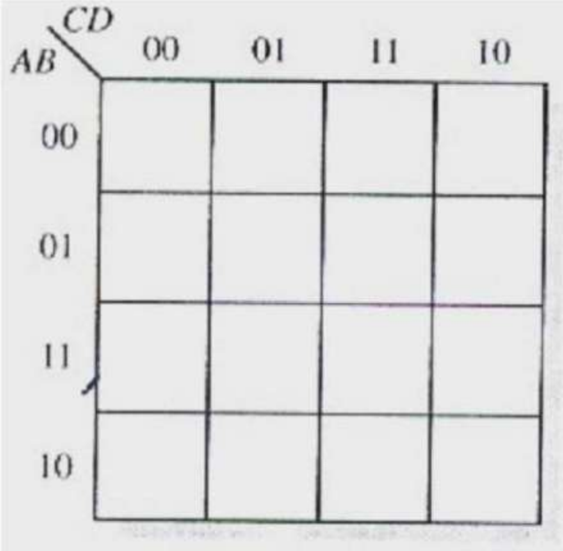


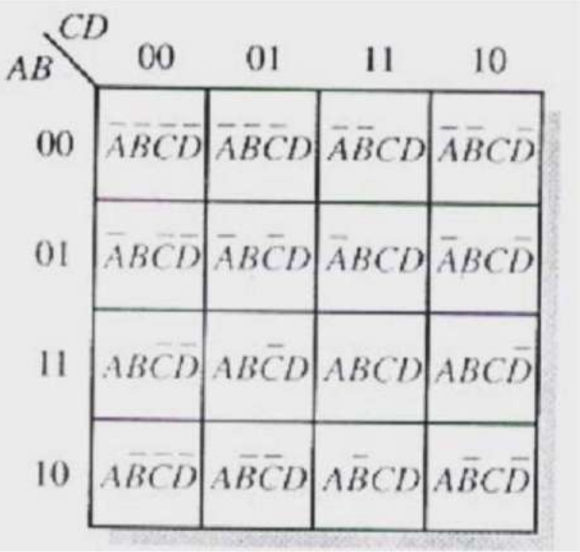
$$\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + AB\bar{C} + A\bar{B}\bar{C}$$

000 001 110 100

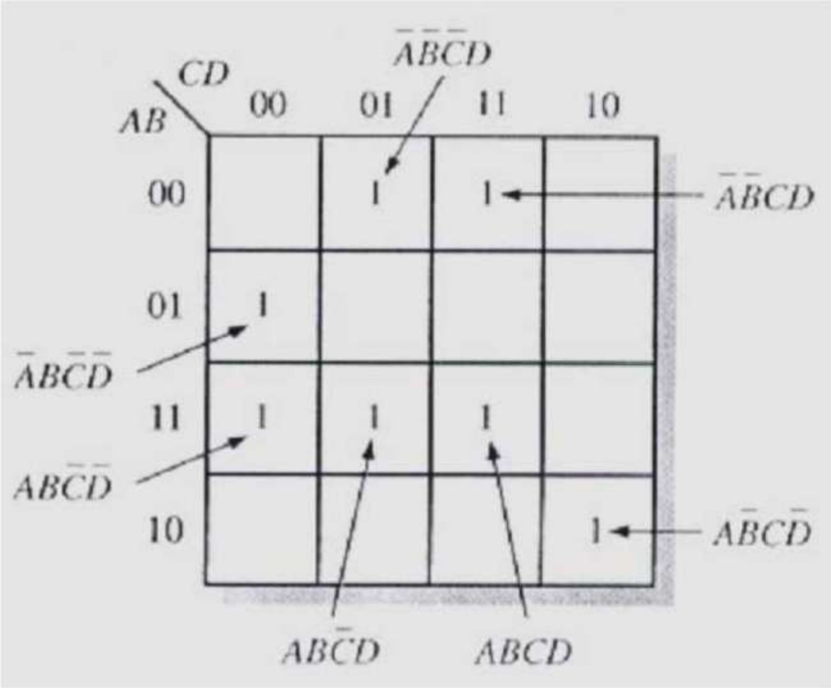
AB \ C	0	1
00	1	1
01		
11	1	
10	1	

2- Four – variable karnaugh map



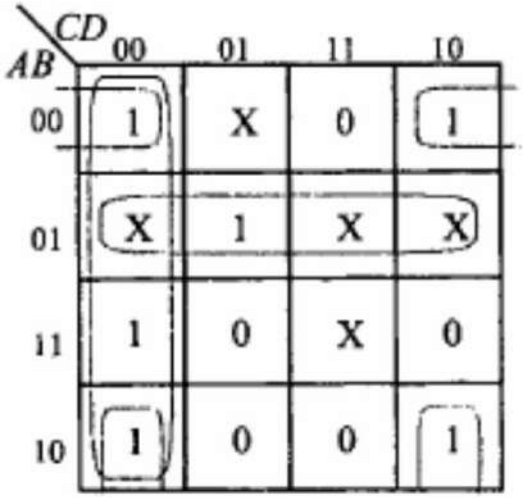


AB \ CD	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}CD$	$\bar{A}\bar{B}C\bar{D}$
01	$\bar{A}B\bar{C}\bar{D}$	$\bar{A}B\bar{C}D$	$\bar{A}BCD$	$\bar{A}BC\bar{D}$
11	$AB\bar{C}\bar{D}$	$AB\bar{C}D$	$ABCD$	$ABC\bar{D}$
10	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}D$	$A\bar{B}CD$	$A\bar{B}C\bar{D}$

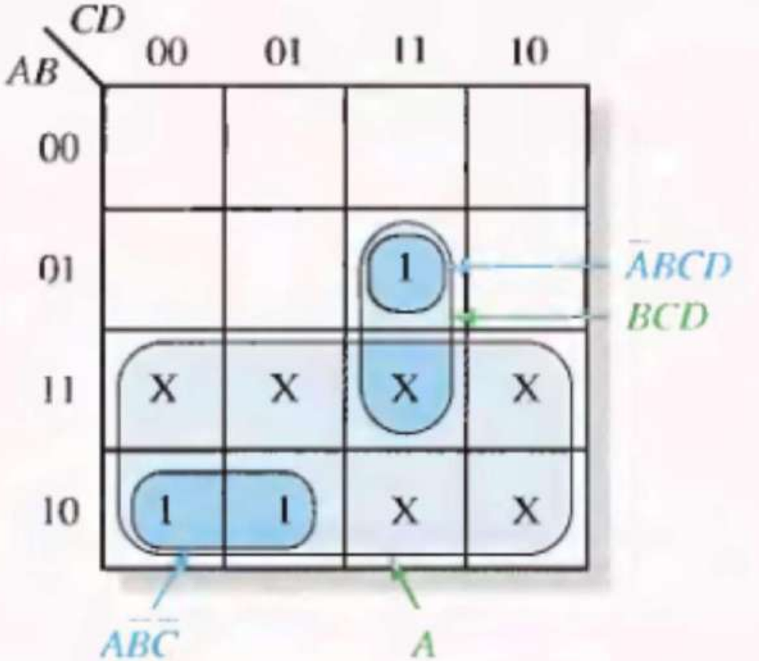


3- Don’t care karnaugh map:-

The squares of a K-map marked with 1’s for the function. The other squares are assumed to be 0’s. This is not always true, because there may be situations



AB \ CD	00	01	11	10
00	1	X	0	1
01	X	1	X	X
11	1	0	X	0
10	1	0	0	1

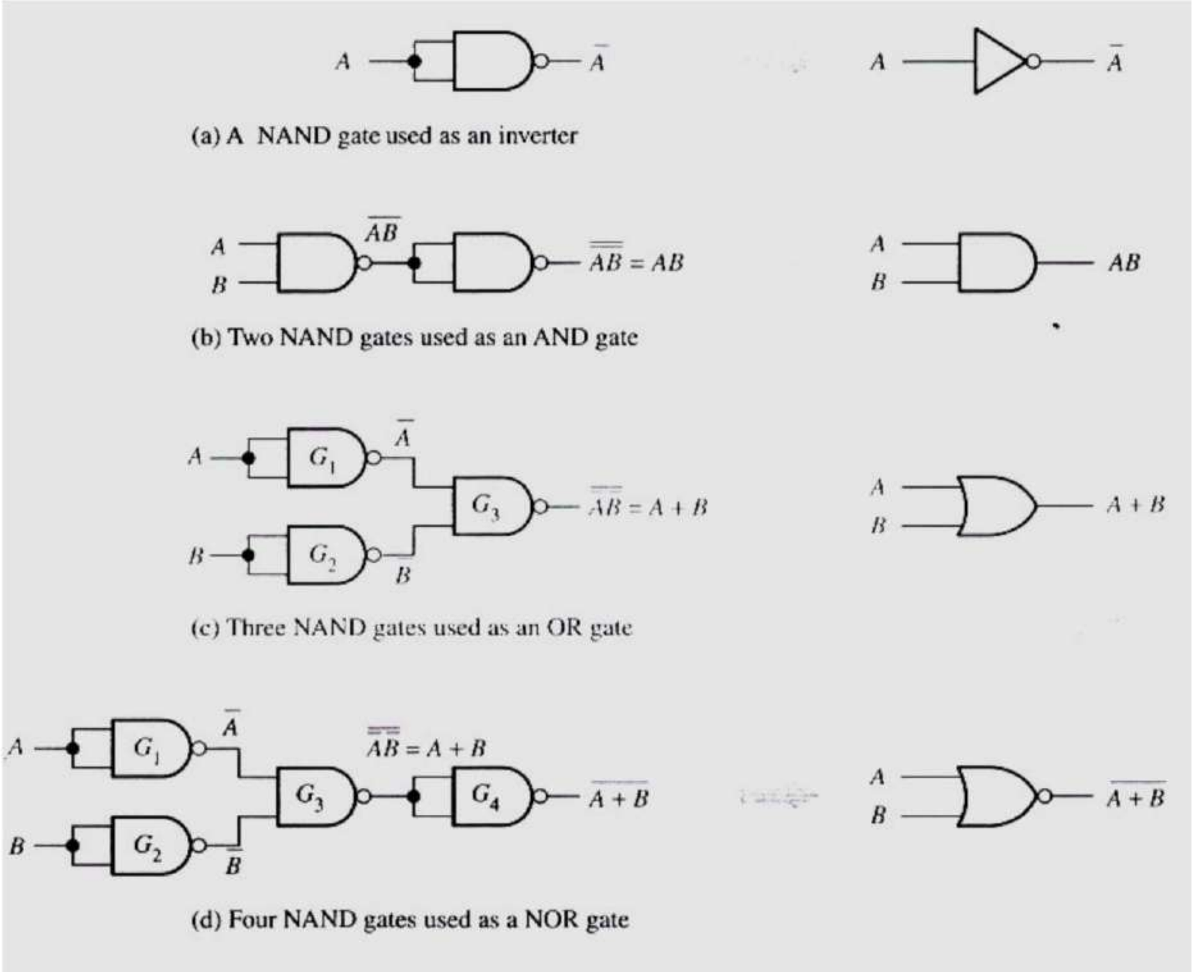


AB \ CD	00	01	11	10
00				
01			1	
11	X	X	X	X
10	1	1	X	X

(b) Without “don’t cares” $Y = A\bar{B}\bar{C} + \bar{A}BCD$
With “don’t cares” $Y = A + BCD$

Combinational Logic:

1-The NAND Gate as a Universal Logic Element:



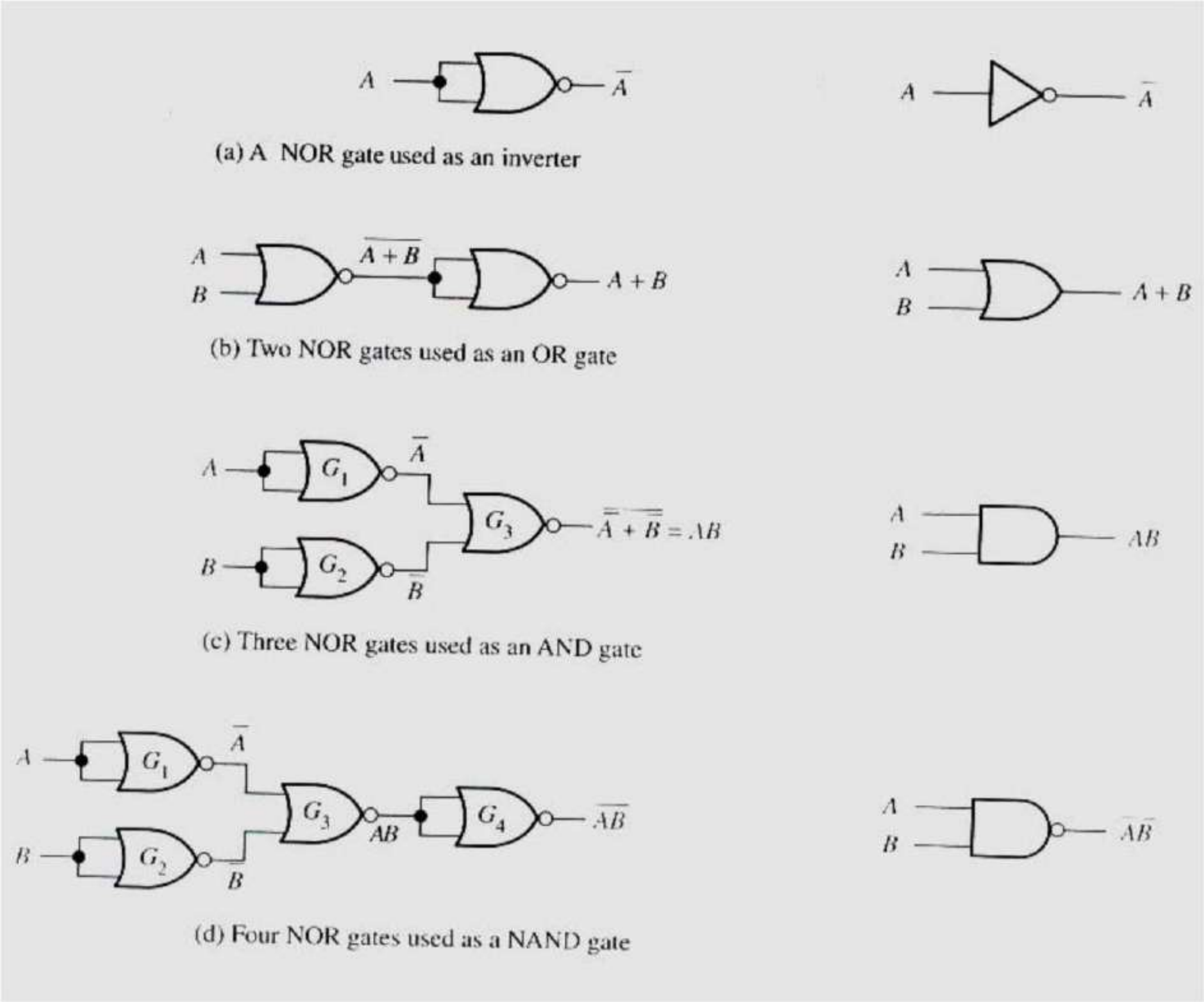
(a) A NAND gate used as an inverter

(b) Two NAND gates used as an AND gate

(c) Three NAND gates used as an OR gate

(d) Four NAND gates used as a NOR gate

2-The NOR Gate as a Universal Logic Element:



(a) A NOR gate used as an inverter

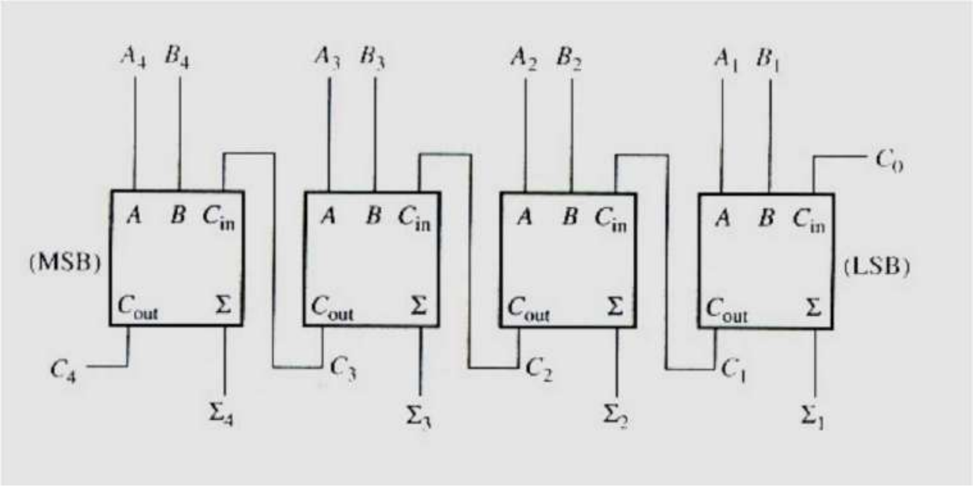
(b) Two NOR gates used as an OR gate

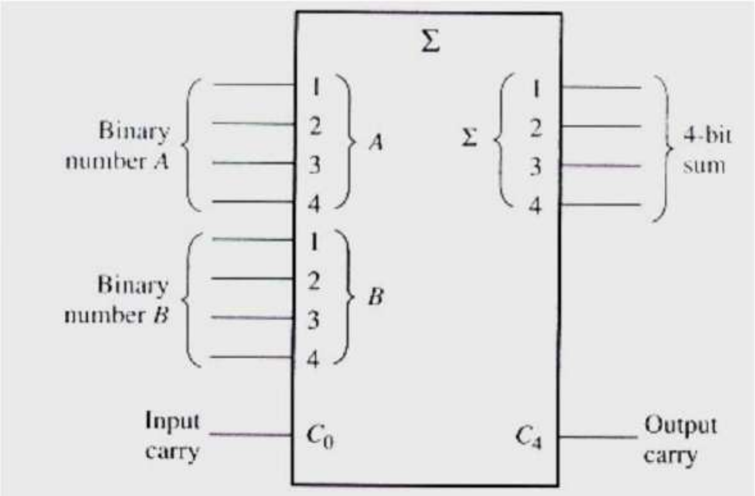
(c) Three NOR gates used as an AND gate

(d) Four NOR gates used as a NAND gate

3- 4- Bit Parallel Adder:

A group of four bits is a nibble. A basic 4-bit parallel adder is implementation with four full adder stages.





Example:

Draw the 4-bit parallel adder, find the sum and output carry for the addition of the following two 4-bit numbers if the input carry (C_{n-1}) is 0:

A4A3A2A1=1010 and B4B3B2B1=1011

Solution:

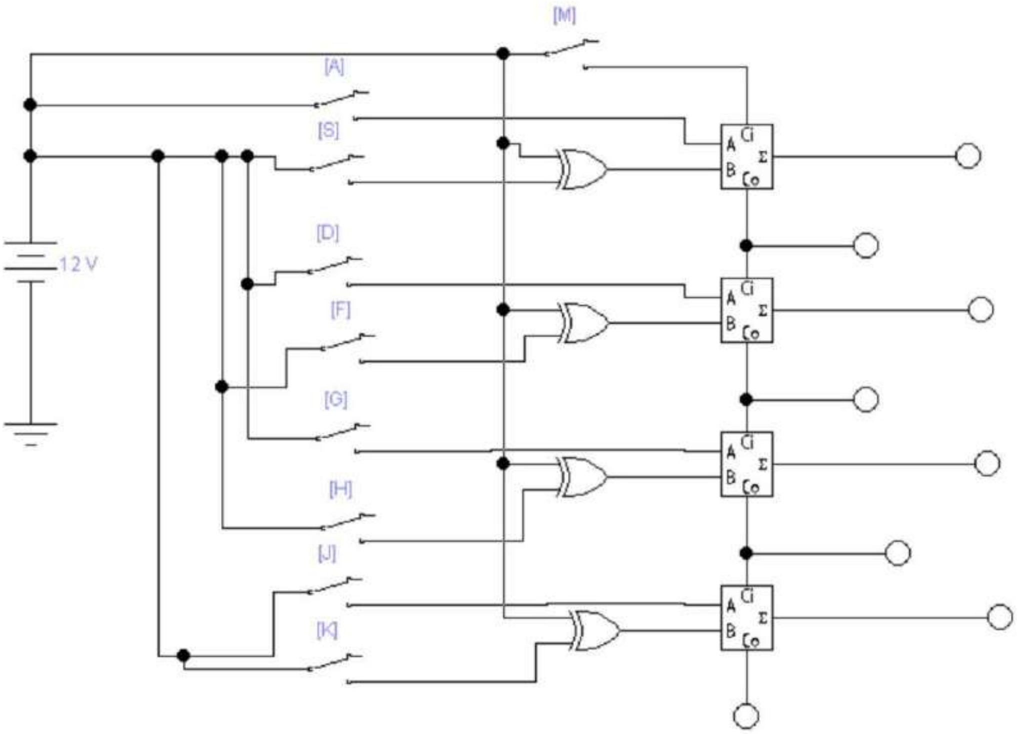
For n=1

A1=0, B1=1, C_{n-1}=0

Σ =1, and C1=0

For n=2
A2=1, B2=1, C_{n-1}=0
Σ=0, and C2=1
For n=3
A3=0, B3=0, C_{n-1}=1
Σ=1, and C3=0
For n=4
A4=1, B4=1, C_{n-1}=0
Σ=0, and C4=1

4- 4-Bit subtracted Adder:



Decoders & encoders:

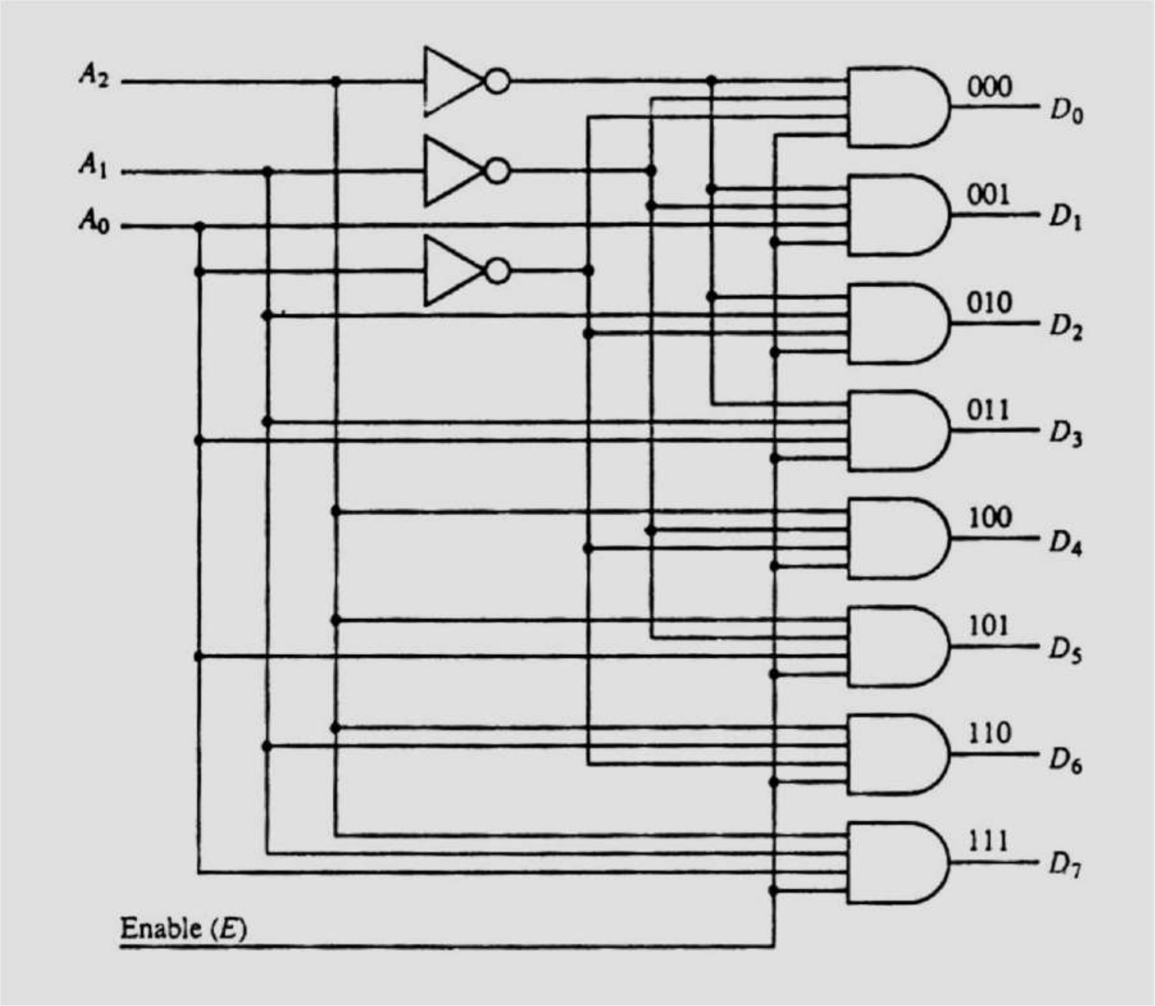
1- Decoder:

A decoders is combinational circuit that converts binary information form the n coded inputs to a maximum of 2n unique outputs.

That decoders are called n-to-m line decoders where m <=2n.

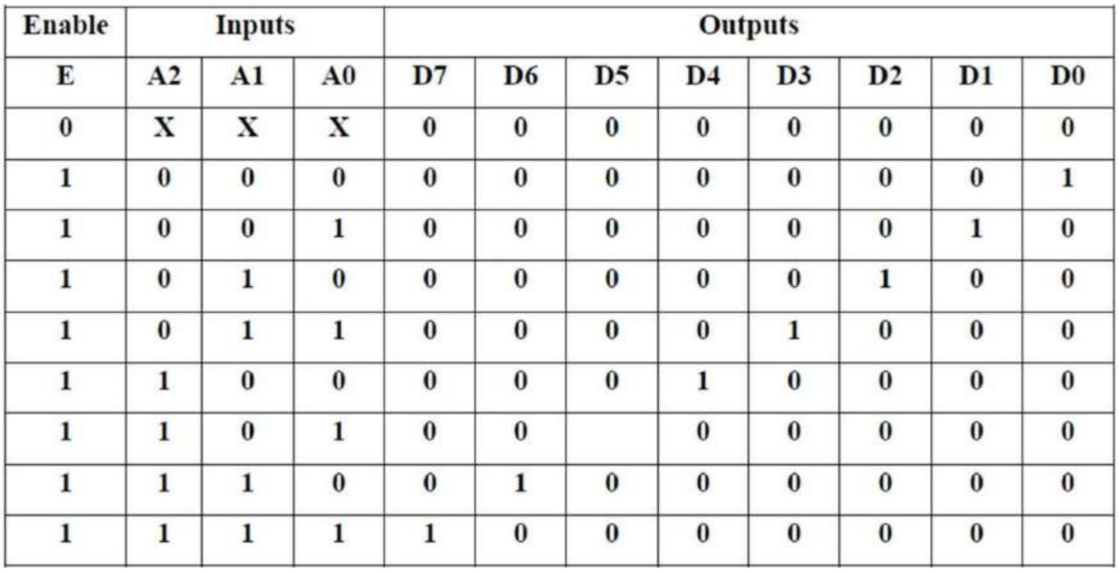
The logic diagram of a 3-to-8 line decoder is three data inputs, A0, A1, and A2 are decoded into eight out puts, each out puts representing one of the combinations of the three binary input variables.

This decoder is a binary – to – octal conversion.

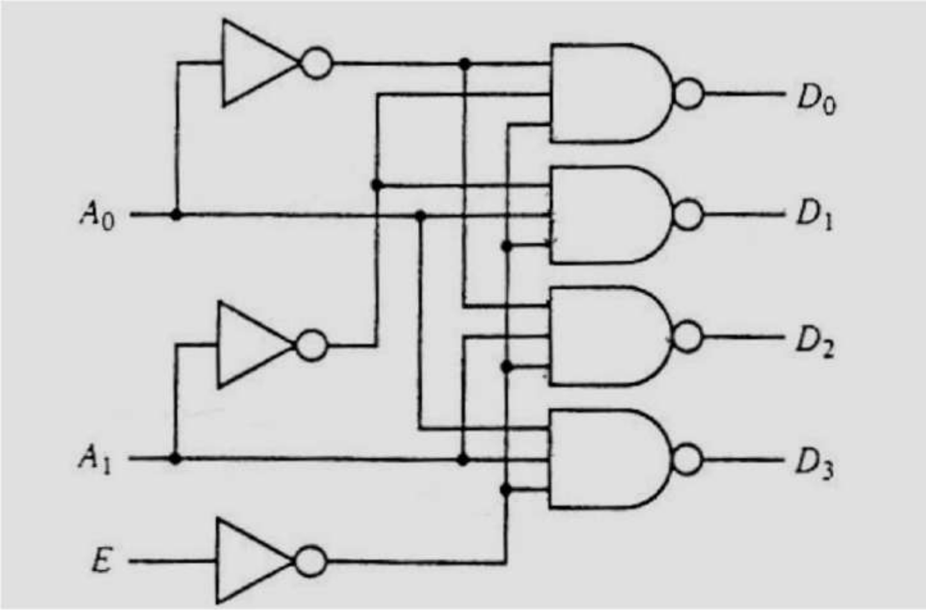


Logic Diagram of 3-8 Decoder

Truth table 3-8 Decoder

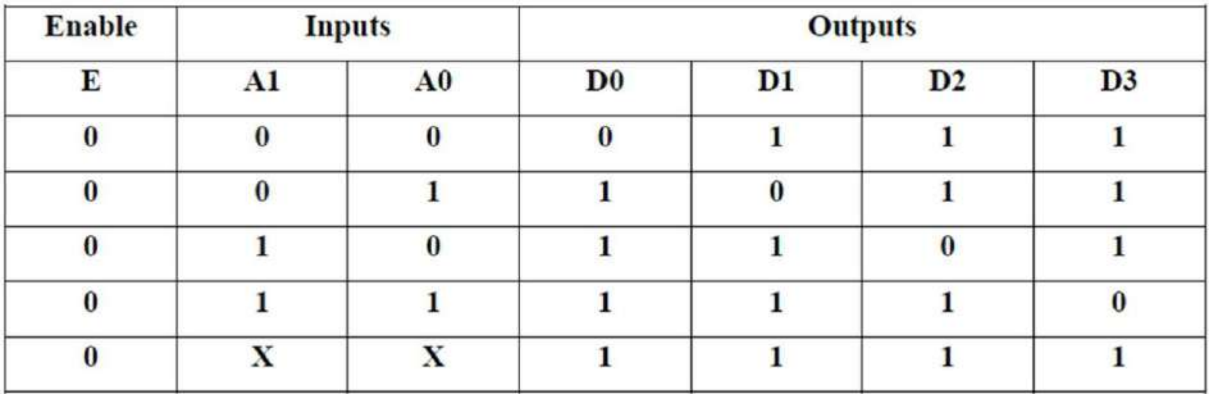


Enable	Inputs			Outputs							
E	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0		0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0



Logic Diagram 2-4 Decoder

Truth table 2-4 Decoder



Enable	Inputs		Outputs			
E	A1	A0	D0	D1	D2	D3
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
0	X	X	1	1	1	1

2- Encoder:

An encoder is a digit circuit that performs the inverse operation of a decoder. An encoder has 2_n (or less) input lines and n output lines. An encoder is the octal – to – binary encoder.

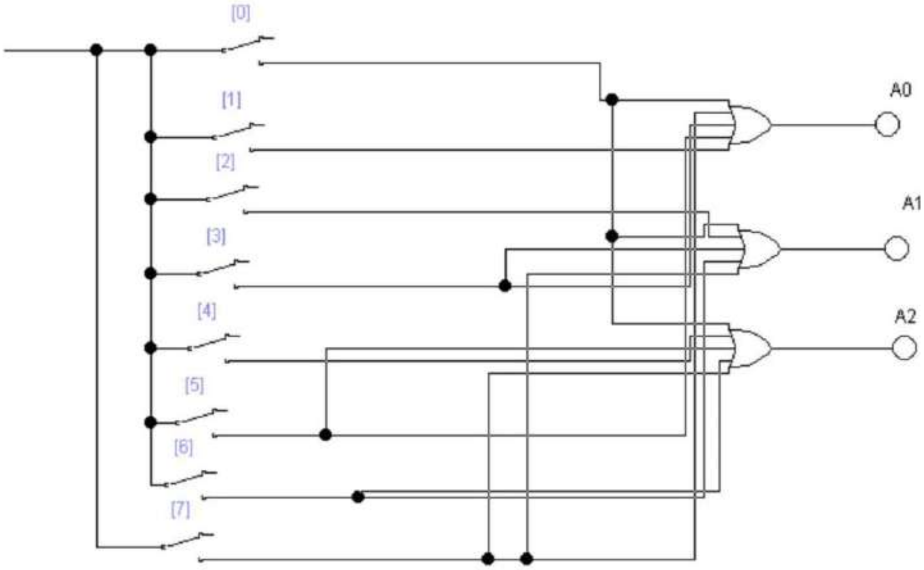
It has eight inputs, one for each of the octal digits, and three outputs that generate the corresponding binary number.

A0 = D1+D3+D5+D7
A1 = D2+D3+D6+D7
A2 = D4+D5+D6+D7
(Implementation in three OR gates)

Truth Table for 8-3 Encoder Octal to Binary

Inputs								Outputs		
D7	D6	D5	D4	D3	D2	D1	D0	A2	A1	A0
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

Logic Diagram for 8-3 Encoder



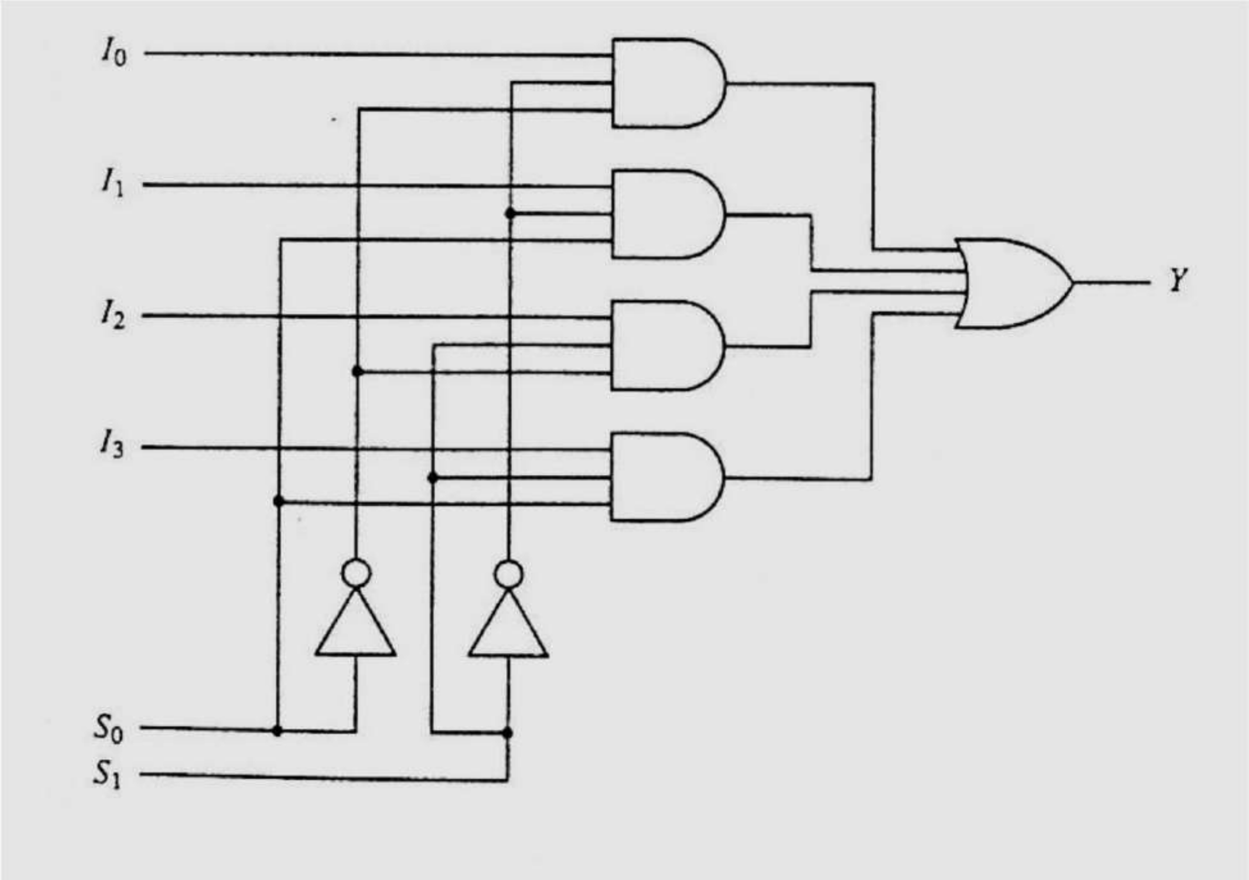
Encoder Octal - to - Binary

3- Multiplexers:

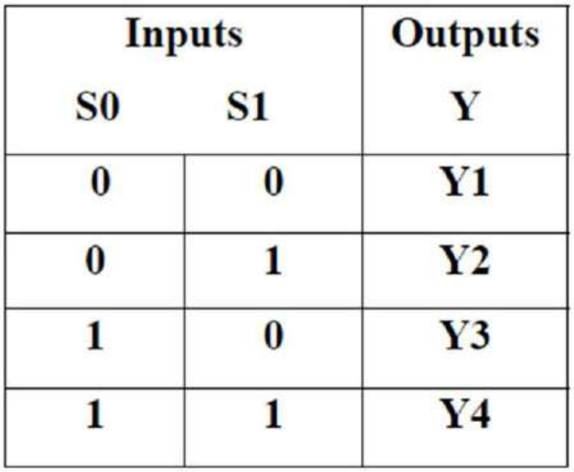
A multiplexer is a combinational circuit that receiver binary information form one of 2_n input data lines and directs it to a single output line.

The selection of a particular input data line for the output is determined by a set of selection inputs. A 2_n- to- 1 , A 4-to-1. Multiplexer is called Data Selector.

Logic Diagram for 4-1 Multiplexer



Truth Table for 4-1 Multiplexer

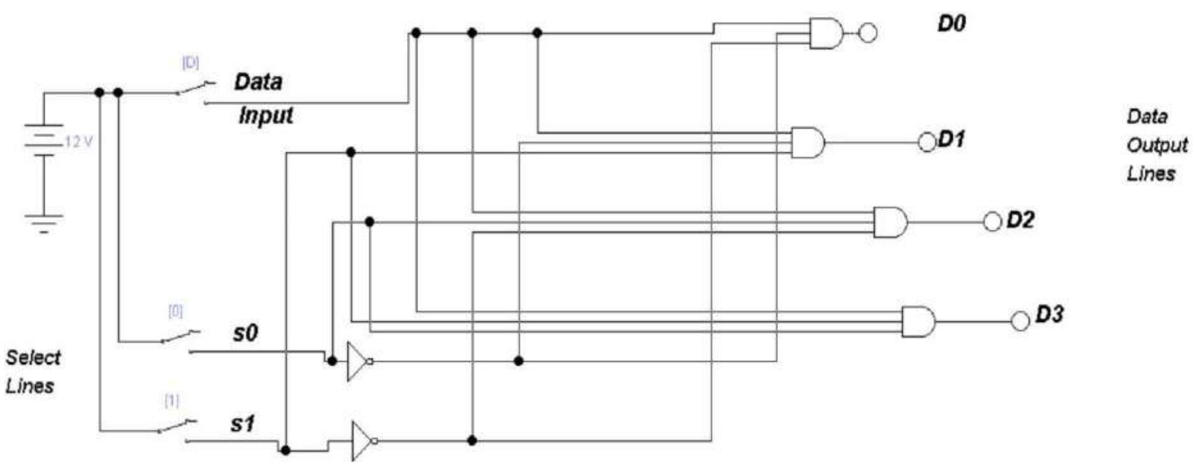


Inputs		Outputs
S0	S1	Y
0	0	Y1
0	1	Y2
1	0	Y3
1	1	Y4

4-Demultiplexers:

A demultiplexer (DEMUX) basically reverses the multiplexing function. It takes digital information from one line and distributes it to a given number of output lines. For this reason, the demultiplexer is also known as a data distributor. As you will learn, decoders can also be used as demultiplexers.
A 1 to 4 lines demultiplexer (DEMUX) circuit. The data input line goes to all of the AND gates. The two data select lines enable only one gate at a time, and the data appearing on the data input line will pass through the selected gate to the associated data output line.

Logic Diagram for 1-4 Demultiplexer



Truth Table for 1-4 Demultiplexer

Inputs			Outputs			
Data	S0	S1	D4	D3	D2	D1
0	0	0	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Flip-Flop:

The storage elements employed in clocked sequential circuits are called flip-flops. A flip -flops is a binary cell capable of storing one bit of information. It has two outputs, one for the normal value and one for the complement value of the bit stored in it.

Type of flip-flops:

1- SR flip-flops.

2- D flip-flops.

3- JK flip-flops.

Latches :

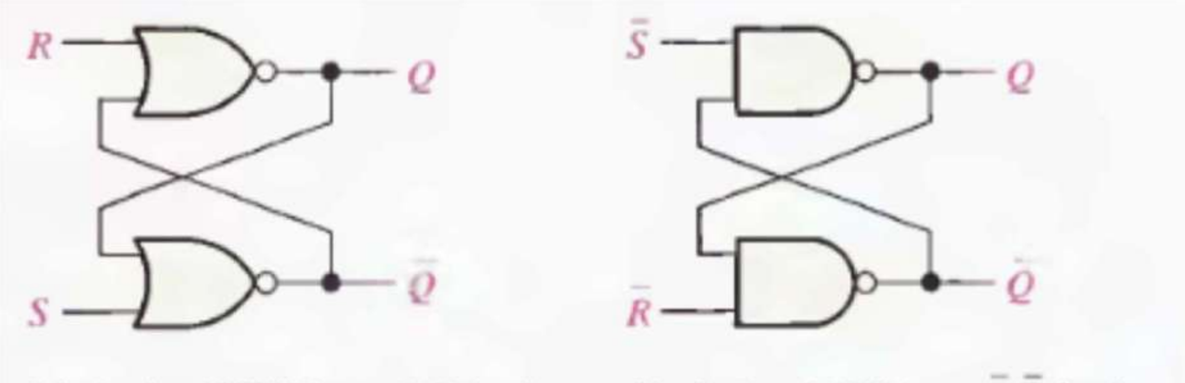
The latch is a type of temporary storage device that has two enable states (bistable) and is normally placed in a category separate from that of flip-flops. Latches are similar to flip-flops because they are bistable devices that can reside in either of two states using a feedback arrangement, in which the outputs are connected back to the opposite inputs. The main difference between latches and flip-flop is the method used for changing their state.

The S-R (SET-RESET) Latch:

A latch is a type of bistable logic device or multivibrator. An active – HIGH input S-R (SET-RESET) latch is formed with two cross-couple NOR gates, as shown in figure (19-a); an active-LOW input latch is formed with two-couple NAND gates, as shown in figure (19-b). Notice that the output of each gate is connected to an input of the opposite gate;

this produces the regenerative feedback that is characteristic of all latches and flip-flops.

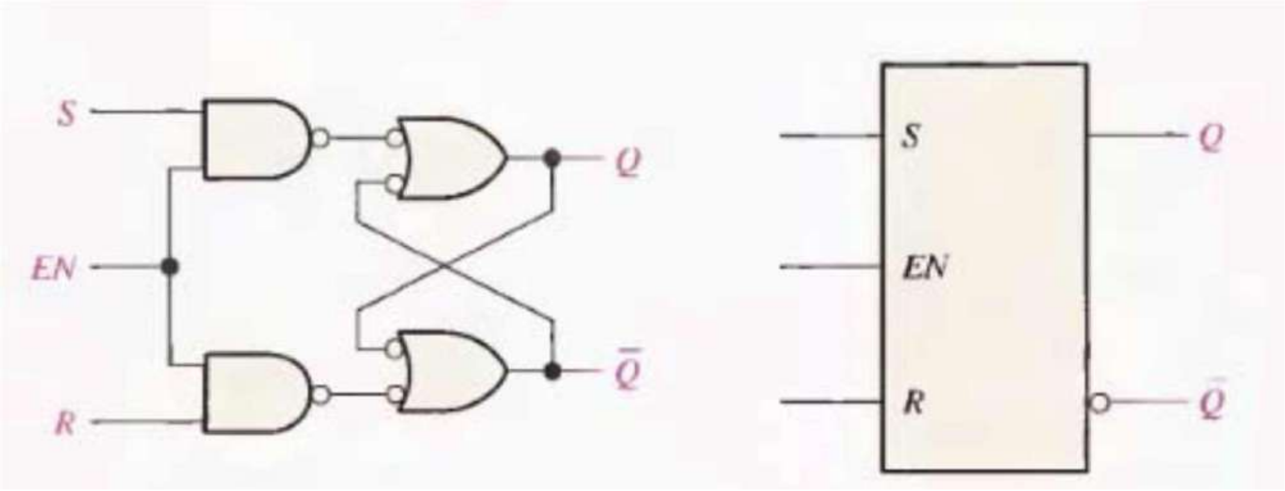
Logic Diagram for S-R Latch



(a) Active-HIGH input S-R latch (b) Active-LOW input $\bar{S}$-$\bar{R}$ latch

The Gated S-R Latch :

A gated latch requires an enable input, EN (G is also used to designate an enable input). The logic diagram and logic symbol for a gated S-R latch are shown in figure (19- c, d). the S and R input control the state to which the latch will go when a HIGH level is applied to the EN input. The latch will not change until EN is HIGH; but as long as it remains HIGH, the output is controlled by the state of the S and R inputs. In this circuit, the invalid state occurs when both S and R are simultaneously HIGH.



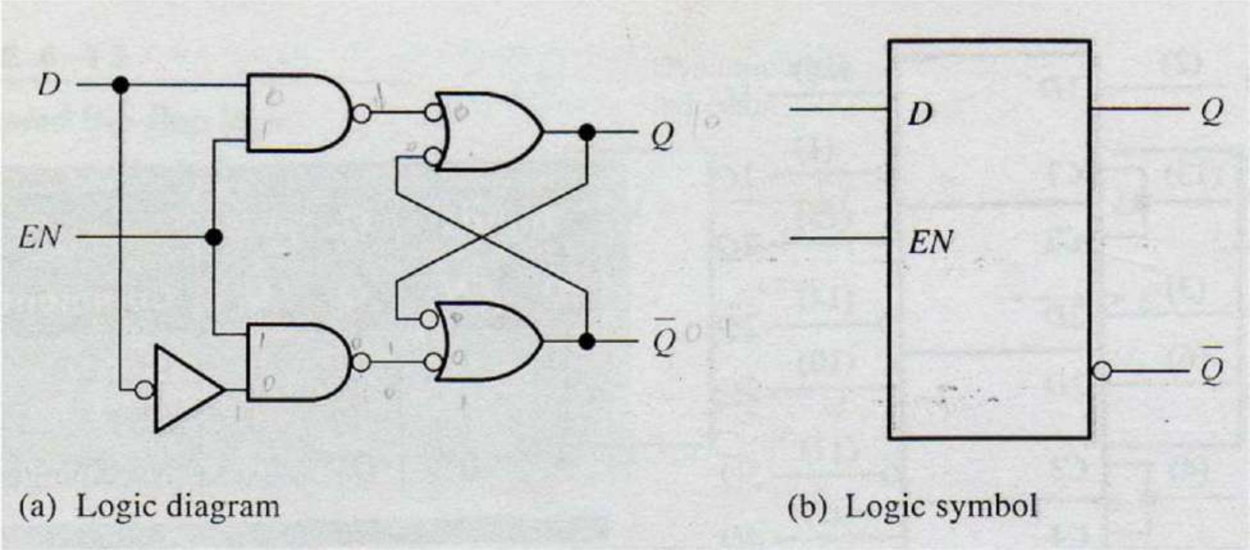
Logic diagram **Logic symbol**

Truth Table for S-R latch

Inputs		Outputs		Comments
S	**R**	**Q**	$\bar{Q}$	
1	**1**	**1**	**1**	**Invalid condition**
0	**1**	**1**	**0**	**Latch set**
1	**0**	**0**	**1**	**Latch reset**
0	**0**	**N.C**	**N.C**	**No change**

2- The Gated D Latch:

Another type of gated latch is called the D latch. It differs from the S-R latch because it has only one input in addition to EN. This input is called the D (data) input. Figure (20–a) contains a logic diagram and logic symbol of a D latch. When the D input is HIGH an the EN input is HIGH, the latch will set. When the D input is LOW and EN is HIGH, the latch will reset. Stated another way, the output Q follows the input D when EN is HIGH.



(a) Logic diagram

(b) Logic symbol

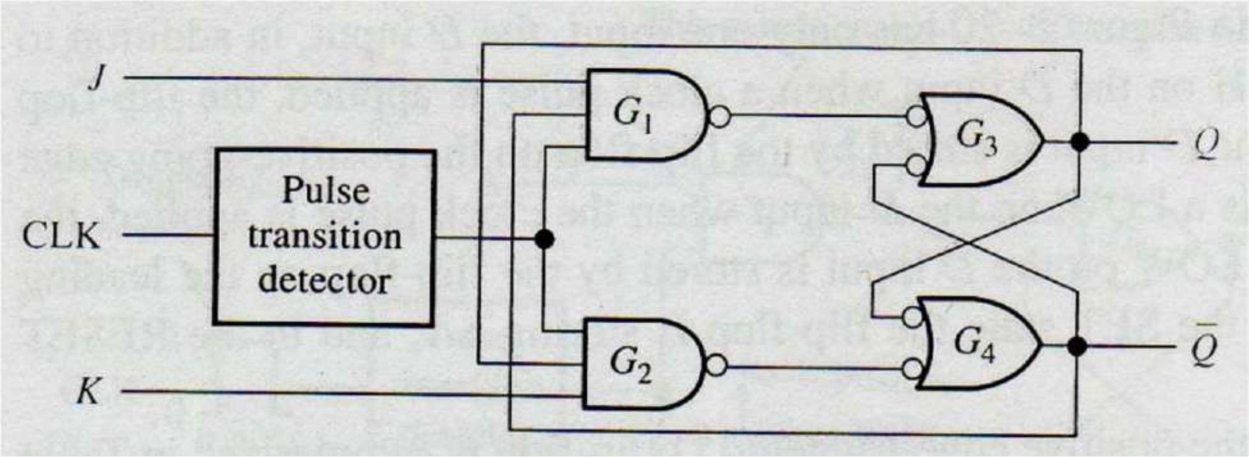
Truth Table for D latch

Inputs		Outputs		
D	**CLK**	**Q**	$\bar{Q}$	**Comments**
1	↑	**1**	**0**	**Set(stor1)**
0	↑	**0**	**1**	**Reset(stor0)**

J -K FLIP-FLOPS:

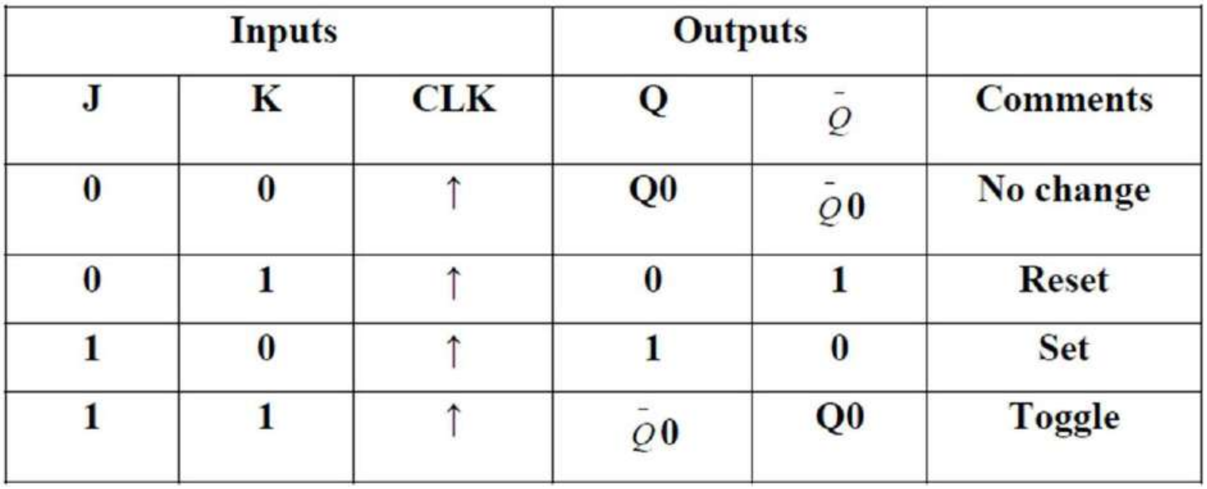
The J-K flip-flop is versatile is a widely used type of flip-flop. The functioning of the J-K flip-flop is identical to that of the S-R flip-flop in the SET. RESET and no-change conditions of operation. The deference is that the J-K flip-flop has no invalid state as does the S-R flip-flop.

Figure (21-a) shows the basic internal logic for a positive edge-triggered J-K flip-flop. It differs from the S-R edge-triggered flip-flop in that Q output is connected back to the input of gate G2, and the Q output is connected back to the input of gate G1. The two control inputs are labeled J and K in honor of jack kilby, who invented the integrated circuit. A J-K flip-flop can also be of the negative edge-triggered type, in which case the clock input is inverted.



The logic Diagram for J-K FF

The Truth Table for J-K FF



Inputs			Outputs		
J	K	CLK	Q	$\bar{Q}$	Comments
0	0	↑	Q0	$\bar{Q}0$	No change
0	1	↑	0	1	Reset
1	0	↑	1	0	Set
1	1	↑	$\bar{Q}0$	Q0	Toggle

Shift Register: A register is a digital circuit with two basic functions:
1- data storage, 2- data movement.

The storage capability of a register makes it an important type of memory device. The concept of storing a 1 or 0 in a D flip flop. A 1 is applied to the data input, and clock puls is applied that stores the 1 by setting the flip-flop when the 1 on the input is removed, the flip-flop remains in the set state, there by storing the 1. A similar procedure applies to the storage of a 0 by resetting the flip-flop.

Type of shift register:

1- Serial in\ Serial out shift right.

2- Serial in\ Serial out shift left.

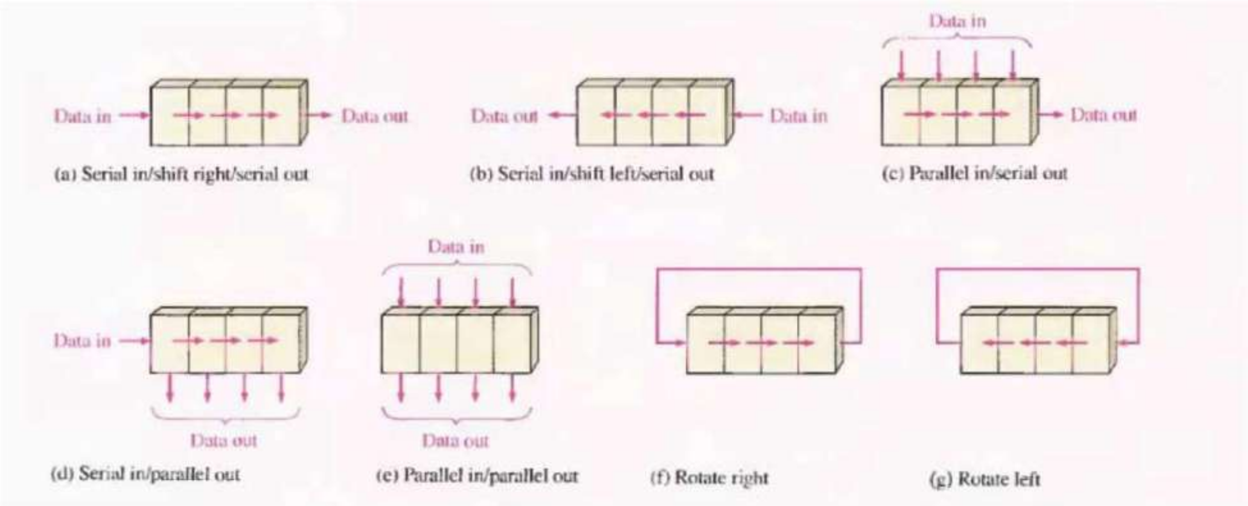
3- Parallel in\Serial out.

4- Serial in\Parallel out.

5-Parallel in\ Parallel out.

6 Rotate right.

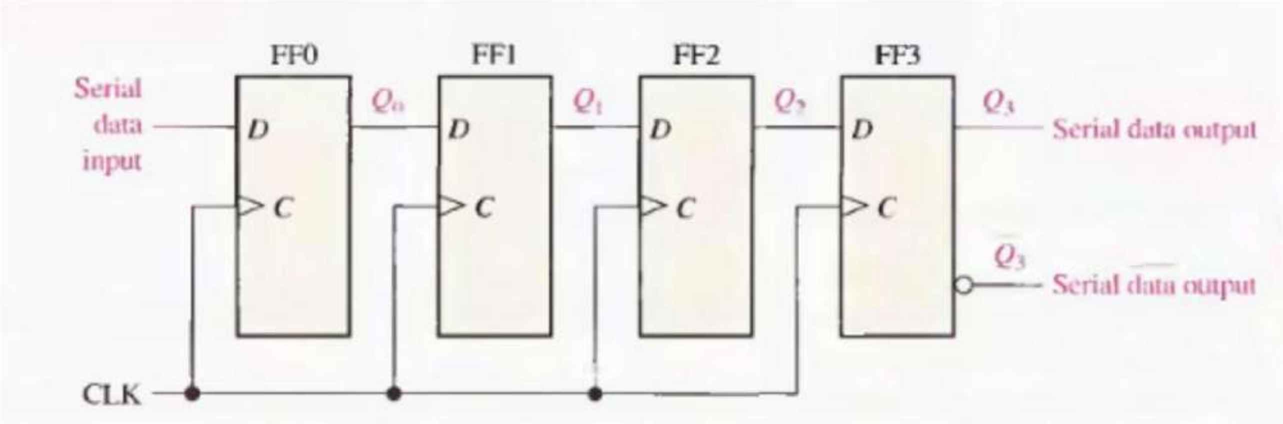
7- Rotate left.



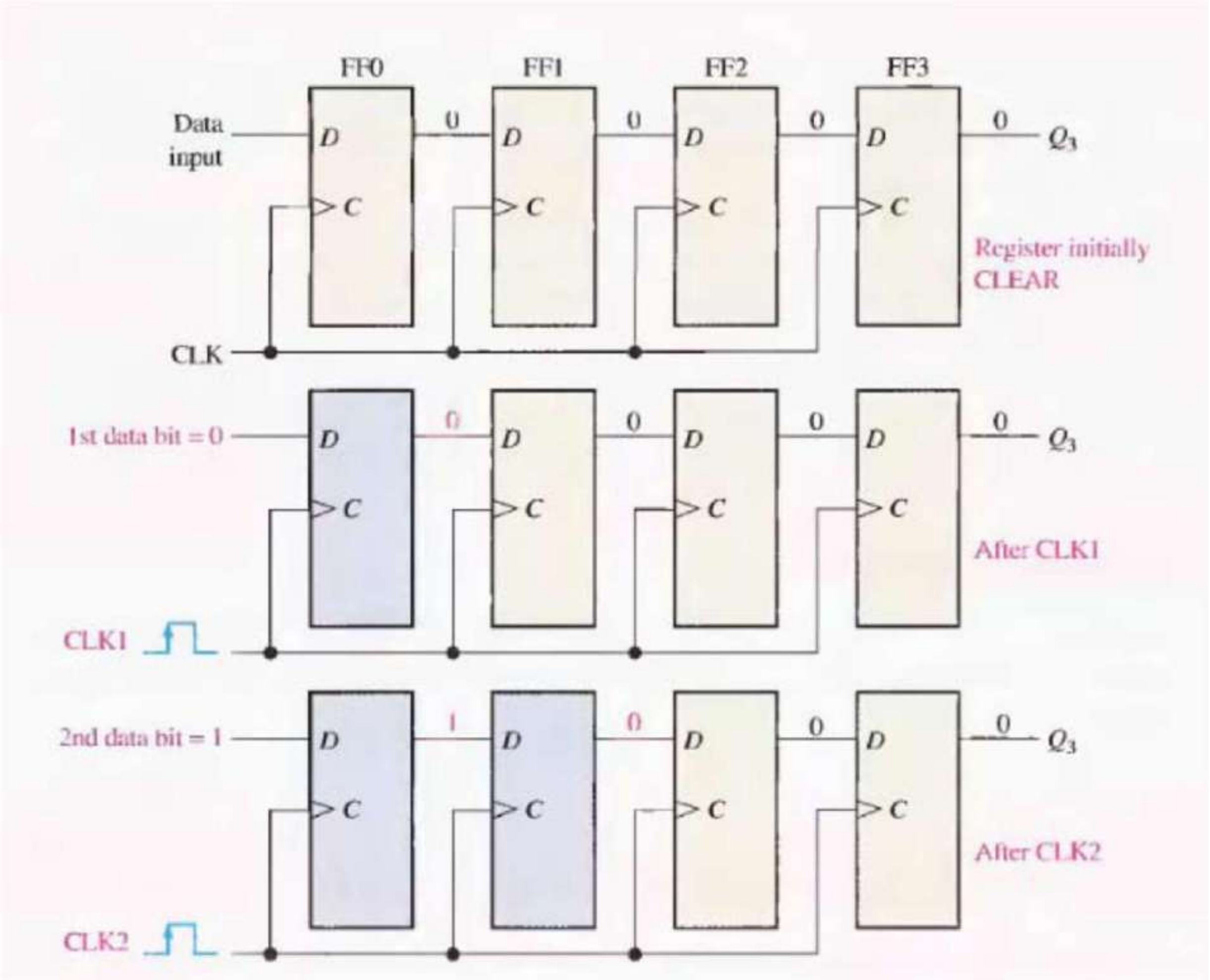
Data in
Data out
(a) Serial in/shift right/serial out
Data out
Data in
(b) Serial in/shift left/serial out
Data in
Data out
(c) Parallel in/serial out
Data in
Data out
(d) Serial in/parallel out
Data in
Data out
(e) Parallel in/parallel out
(f) Rotate right
(g) Rotate left

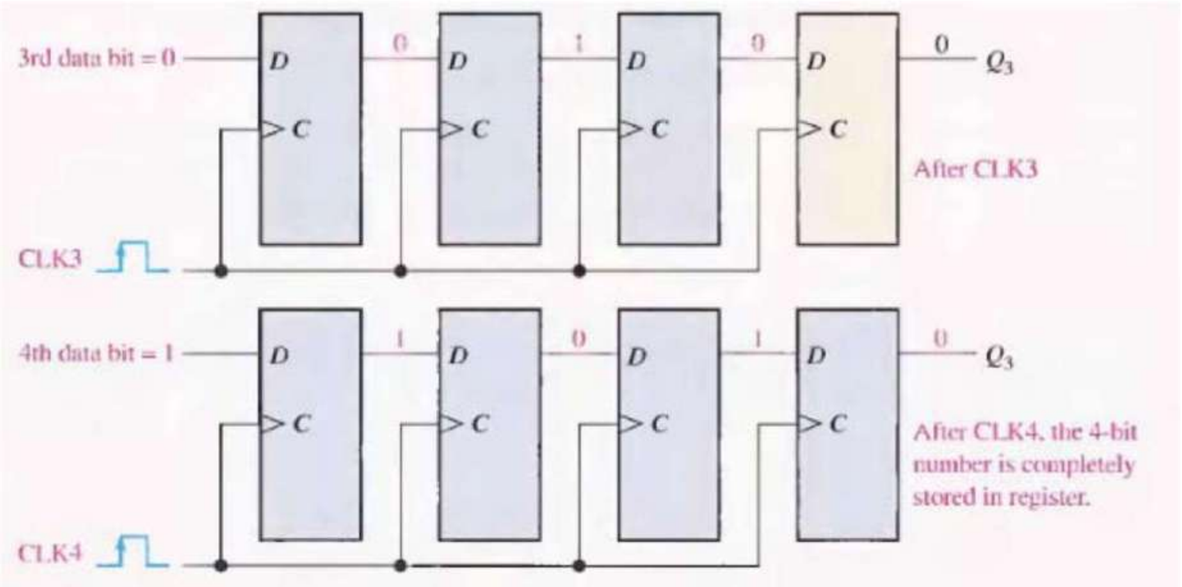
Types of Shift Registers

1- Serial in \ Serial out shift Register:

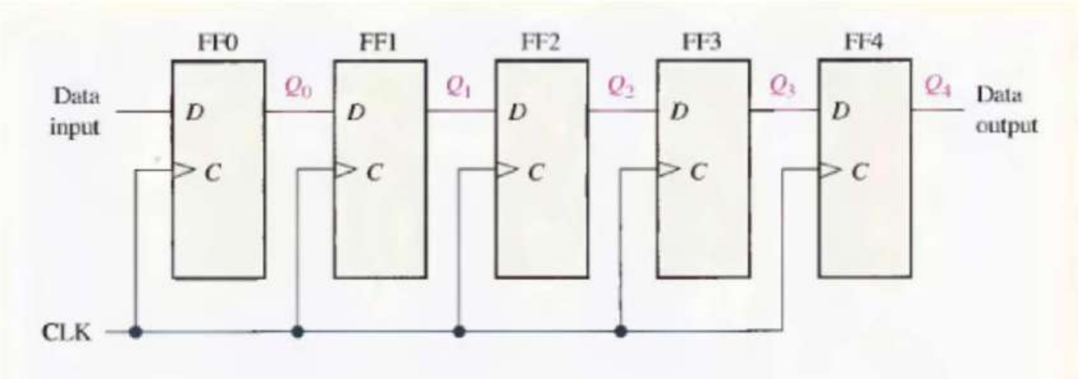


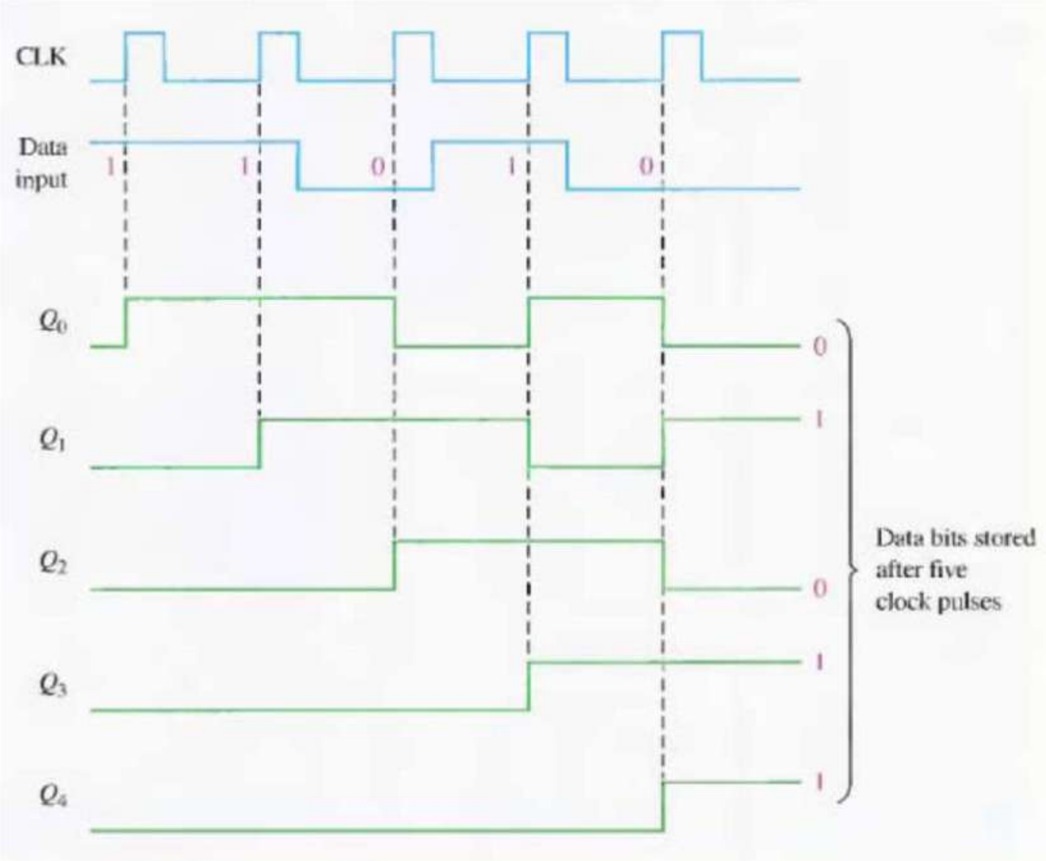
Example: 1
Shift Register 4-bit


FF0
FF1
FF2
FF3
Data input
D
C
0
Q3
Register initially CLEAR
CLK
1st data bit = 0
After CLK1
CLK1
2nd data bit = 1
1
After CLK2
CLK2

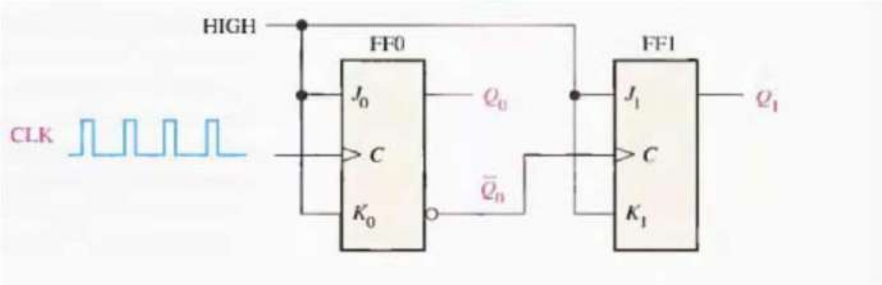

3rd data bit = 0
D
C
0
1
Q3
After CLK3
CLK3
4th data bit = 1
After CLK4, the 4-bit number is completely stored in register.
CLK4

Example: 2
Draw 5-bit shift register and write wave form

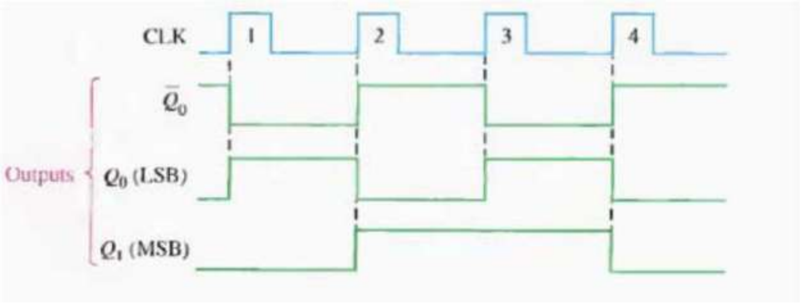




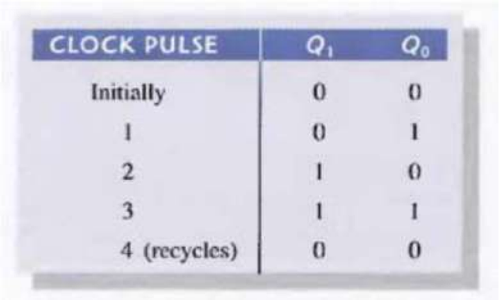
Binary Counter: The binary counter is consist two types.
1- Asynchronous counter operation.
2- Synchronous counter operation.
1- Asynchronous counter operation:
In figure (25-a, b, c) shows a 2-bit counter connected for asynchronous operation. Notice that the clock (CLK) is applied to the clock input (C) of only the first flip-flop, FF0, which is always the least significant bit (LSB). The second flip-flop, FF1, is triggered by the Q0 output of FF0. FF0 changes state at the positive-going edge of each clock pulse, but FF1 changes only when triggered by a positive-going transition of the Q0 output of FF0. Because of the inherent propagation delay time through a flip-flop, a transition of the input clock pulse (CLK) and transition of the Q0 output of FF0 can never occur at exactly the same time. Therefore, the two flip-flops are never simultaneously triggered, so the counter operation is asynchronous.



2-Bit Asynchronous Binary Counter



Time Diagram 2-Bit Asynchronous Binary Counter

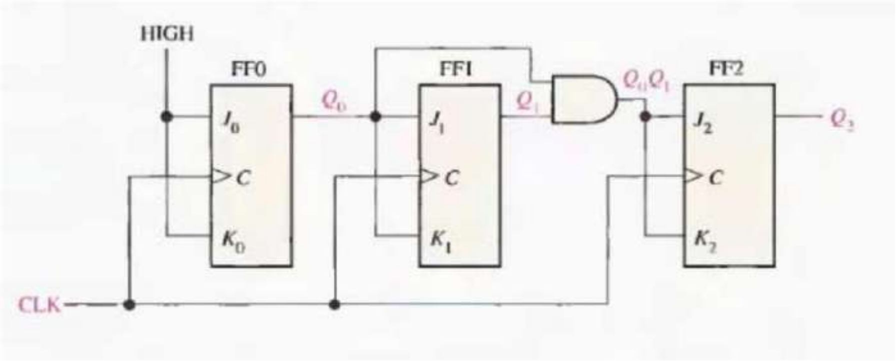


CLOCK PULSE	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

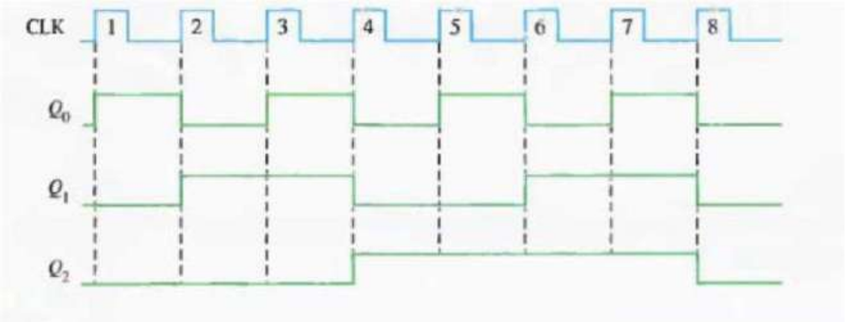
Truth Table 2-Bit Asynchronous Binary Counter

2- Synchronous counter operation:

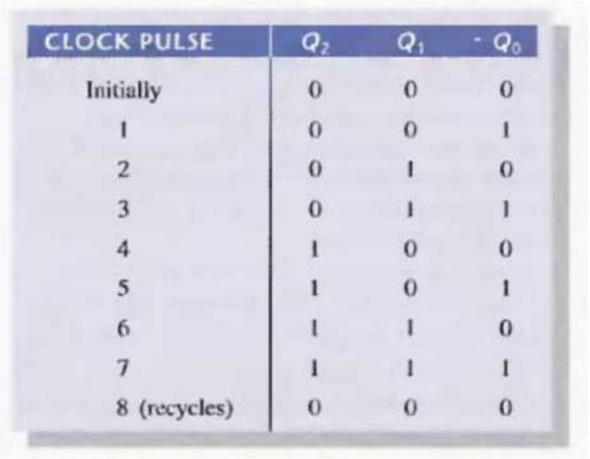
The term synchronous refers to events that have a fixed time relationship with each other. A synchronous counter is one in which all the flip-flops in the counter are clocked at the same time by a common clock pulse.



3-Bit Synchronous Binary Counter



Time diagram 3-Bit Synchronous Binary Counter



CLOCK PULSE	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recycles)	0	0	0

truth table for 3-Bit Synchronous Binary Counter

Answer these questions:

Q1- Convert the following:

1- (CF8E)16 to ( )10
2- (1725)10 to ( )16
3- (148.625)10 to ( )2
4- (7526)8 to ( )10
5-(2591)10) to ( )16
6- (B2F8)16 to ( )10

Q2- Perform the following:

1- (2AB)16 – (317)16
2- (101101)2 – (1110)2
3- (6410)8 – (324)8
4- (2CF)16 – (FDB)16
5- (4732)8 + (4611)8

Q3- Express the decimal number -98, -68 as 8-bit number in the sign-magnitude, 1'S and 2'S Complement.

Q4- Design Full-Adder circuit.

Q5- Design Half-Adder circuit.

Q6- Draw the 4-bit parallel adder, find the sum and output carry for the addition of the following two 4-bit numbers if the input carry (Cn-1) is 0: A4A3A2A1=1011 and B4B3B2B1=0111.

Q7- use K- map to minimize the following SOP expression and convert to POS in K- map.

F(A,B,C,D)= Ʃ2,3,4,5,6,7,9,12,13,14,15

Q8- Design 3-to-8 lines decoders. OR Design Binary-to-Octal line decoder.

Q9- Design 2- to – 4 lines decoders

Q10- Design block diagram of quadruple 2-to-1 line multiplexer.

Q11- Design SR flip-flop and explain function.

Q12- Design D flip-flop and explain function.

Q13- Design JK flip-flop and explain function.