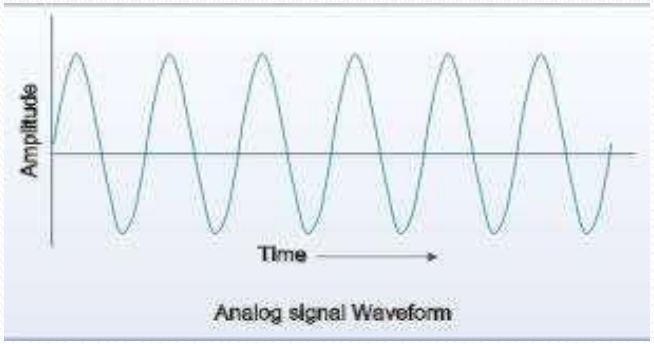
INTRODUCTION

CHAPTER-1

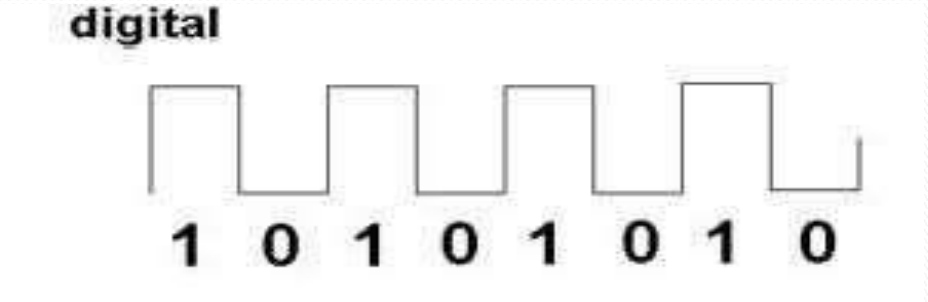
ANALOG SIGNAL

- An **analog signal** is any continuous signal for which the time-varying feature (variable) of the signal is a representation of some other time varying quantity, i.e., *analogous* to another time varying signal.
- For example, in an analog audio signal, the instantaneous voltage of the signal varies continuously with the pressure of the sound waves.



DIGITAL SIGNAL

- A **digital signal** is a signal that is being used to represent data as a sequence of discrete values; at any given time it can only take on one of a finite number of values in most digital circuits.
- The signal can have two possible values(0,1); this is called a binary signal or **logic signal**.[4]

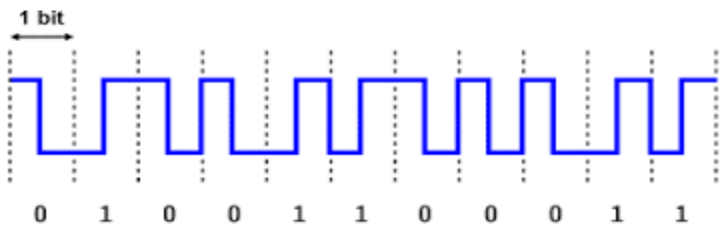


Difference Between Analog And Digital Signal

Factors	Analog	Digital
Waves	Denoted by Sine waves	Denoted by Square waves
Signal	Continuous signal representing physical measurements	Discrete signal representing discrete time signals generated by digital modulation
Data Transmission	Subject to deterioration by noise	Noise-immune without deterioration
Bandwidth	Consumes less bandwidth	Consumes more bandwidth
Memory	Stored in the form of wave signal	Stored in the form of binary bit
Power	Draws large power	Draws negligible power
Impedance	Low impedance	High order of 100 megaohm
Errors	Analog instruments have considerable observational errors	Digital instruments are free from observational errors

Advantages Of Digital Signal

- Easier To Design.
- Digital System Have Got Fast Response Time.
- Information Can Be Stored And Retrived Very Easily.
- More Accurate And Have Great Percision.
- Less Effected By Noise.
- Easier To Use Because Direct Display Of Data Is Convenient To Read.
- Digital Ckt. Can Be Fabricated On IC Chips.



Advantages of Analog Signal

- Best suited for the transmission of audio and video.
- Consume less bandwidth than digital signals to carry the same information.
- Analog signal is less susceptible to noise.

Applications Of Digital Signal

- Data Base Management System(DBMS) Used In Banks, Offices, Institutes, Shops etc. Using Computers.
- Process Monitoring And Control System In Industries Using Computers, PLC's, Robots.
- Digital Signal Processing And Digital Communication.
- Entertainment Appliances Like CD/DVD Player LED TVs, Digital Cameras.
- Medical Instruments Like Digital X-ray Machin Ultra Sound Machines etc.
- Combustion Control In Modern Vechicles.

Applications Of Analog signal

→Thermometer.
→photocopiers
→old land-line telephone
→audio tapes
→VCRs (same as TV)

NUMBER SYSTEM

CHAPTER -2

Number system & its types

- Code using symbols that refers to a set of items.
- **Types of Number system**

  Binary number system

  Decimal number system

  Octal number system

  Hexa -decimal number system

Binary Number System

- A number system which have two values 0 and 1 is called binary number system.
- The base or radix is 2.

Decimal Number system

- The decimal number system contains ten distinct symbols (0 to 9).
- The base or radix is 10.

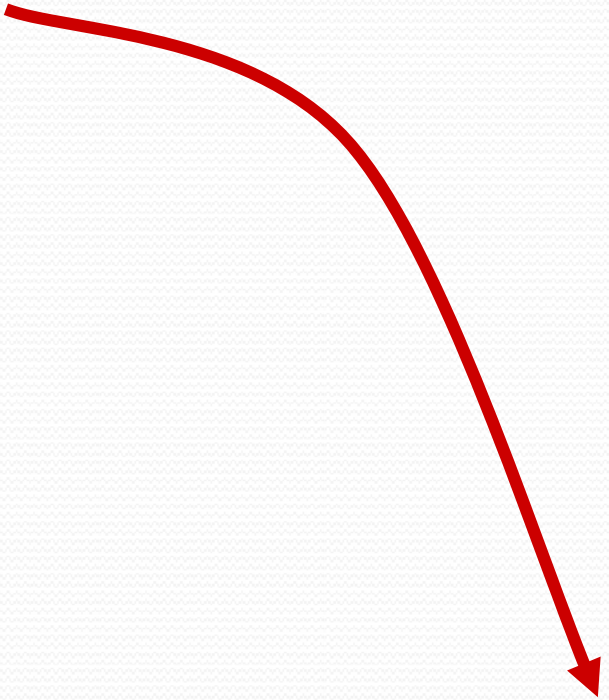
Decimal to Binary conversion

- Technique
  - Divide by two, keep track of the remainder
  - First remainder is bit 0 (LSB, least-significant bit)
  - Second remainder is bit 1
  - Etc.

Example

$125_{10} = ?_2$

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1



$125_{10} = 1111101_2$

Binary to Decimal Conversion

- Technique
  - Multiply each bit by 2^n, where n is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

101011_2 =>

$1 \times 2^0 = 1$

$1 \times 2^1 = 2$

$0 \times 2^2 = 0$

$1 \times 2^3 = 8$

$0 \times 2^4 = 0$

$1 \times 2^5 = 32$

43_{10}

Octal Number system

- In this system, there are eight distinct symbols that represent octal numbers(0 to 7).
- The base or radix is 8.

Hexa-decimal Number system

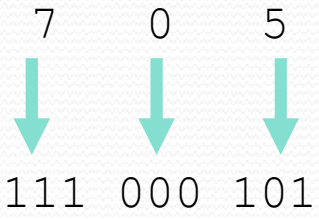
- This system uses 16 distinct symbols (o to 9, A toF).
- The base or radix is 16.

Octal to Binary conversion

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation

Example

$705_8 = ?_2$



$705_8 = 111000101_2$

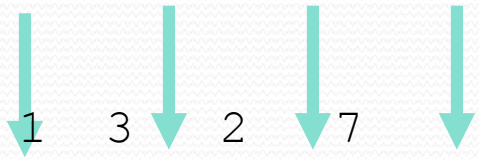
Binary to Octal

- Technique
  - Group bits in threes, starting on right
  - Convert to octal digits

Example

$1011010111_2 = ?_8$

1 011 010 111



1 3 2 7

$1011010111_2 = 1327_8$

Octal to Decimal

- Technique
  - Multiply each bit by 8^n, where n is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

724_8 =>

$4 \times 8^0 = 4$

$2 \times 8^1 = 16$

$7 \times 8^2 = 448$

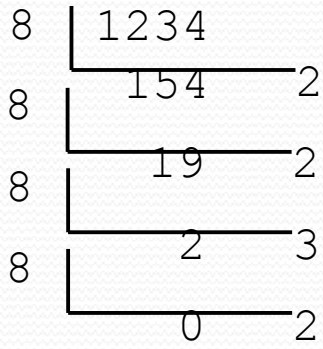
468_{10}

Decimal to Octal

- Technique
  - Divide by 8
  - Keep track of the remainder

Example

$1234_{10} = ?_{8}$



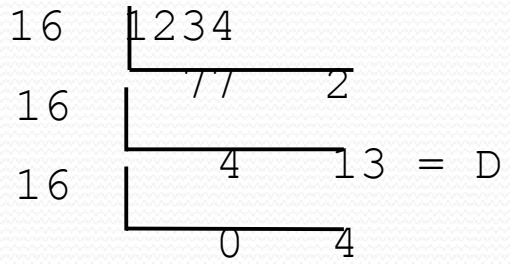
$1234_{10} = 2322_{8}$

Decimal to Hexadecimal

- Technique
  - Divide by 16
  - Keep track of the remainder

Example

$1234_{10} = ?_{16}$



$1234_{10} = 4D2_{16}$

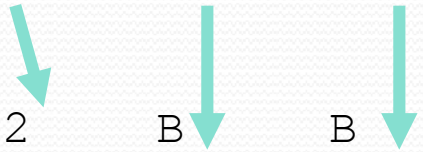
Binary to Hexadecimal

- Technique
  - Group bits in fours, starting on right
  - Convert to hexadecimal digits

Example

$1010111011_2 = ?_{16}$

10 1011 1011



2 B B

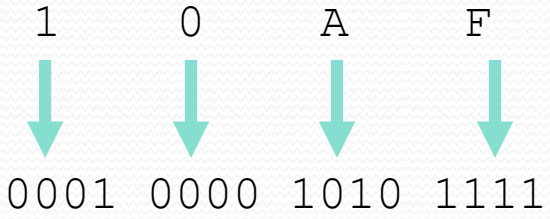
$1010111011_2 = 2BB_{16}$

Hexadecimal to Binary

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation

Example

$10AF_{16} = ?_2$



$10AF_{16} = 0001000010101111_2$

Hexadecimal to Decimal

- Technique
  - Multiply each bit by 16^n, where n is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

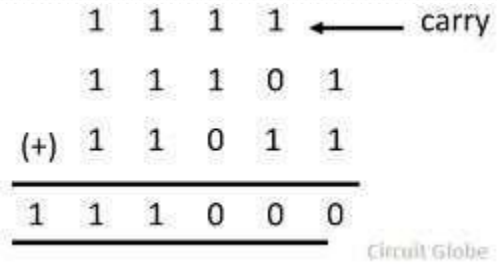
ABC_{16} =>

C x 16^0 = 12 x 1 = 12
B x 16^1 = 11 x 16 = 176
A x 16^2 = 10 x 256 = 2560

2748_{10}

Binary Addition

- The binary number system uses only two digits 0 and 1. So , there are four basic operations.



0+0=0

0+1=1

1+0 = 1

1+1 = 10

Binary Subtraction

The subtraction of binary digit depends on four basic operation.

0-0 = 0

1-0 = 1

1 – 1 = 0

10 – 1 = 1

- 1011011 – 10010 = 1001001:

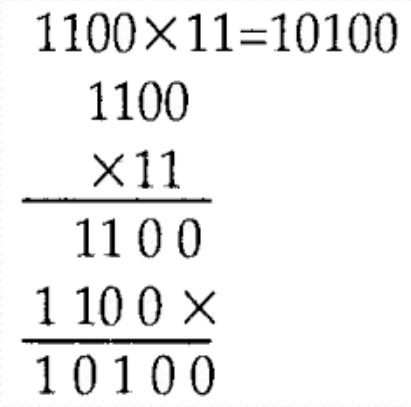
	1	0	1	1	0	1	1
–			1	0	0	1	0
	1	0	0	1	0	0	1

- 100010110 – 1111010 = 10011100:

	0	1	1	1	$^{1}0$			
	$\not{1}$	$^{1}\not{0}$	$^{1}\not{0}$	$^{1}\not{0}$	$\not{1}$	$^{1}0$	1	1	0
–			1	1	1	1	0	1	0
		1	0	0	1	1	1	0	0

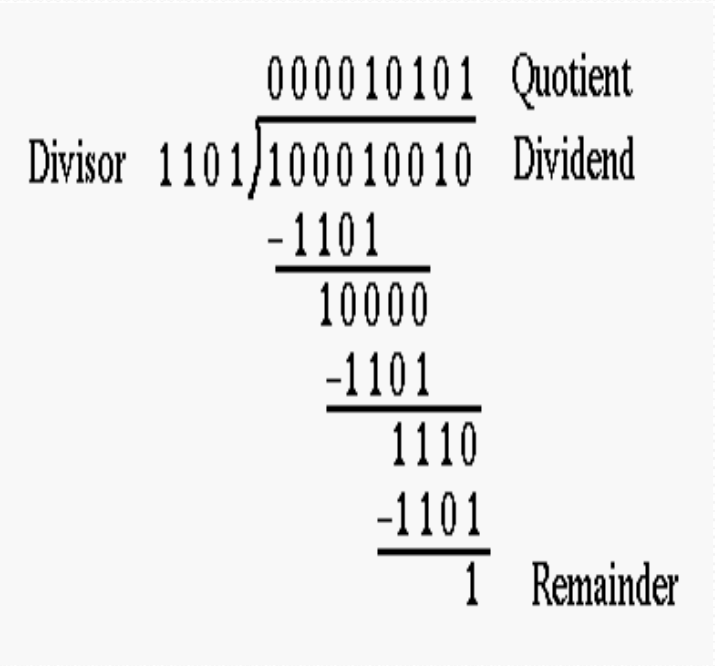
Binary Multiplication

- The rules for binary multiplication are:
- 0 x 0 = 0
- 0 x 1 = 0
- 1 x 0 = 0
- 1 x 1 = 1



Binary Division

- The rules binary division are:
- 0 ÷ 1 =0
- 1 ÷ 1 = 1



Codes and Parity

CHAPTER- 3

Binary Codes

- Binary codes are used in computers and digital communication. These binary codes can be classified as:
- Weighted codes
- Non- weighted codes

Binary Codes

- Electronic digital systems use signals that have two distinct values and circuit elements that have two stable states.
- Digital systems represent and manipulate not only binary numbers, but also many other discrete elements of information.
- Any discrete element of information distinct among a group of quantities can be represented by a binary code.
- Binary codes merely change the symbols, not the meaning of the elements of information that they represent.

Weighted codes

- A weighted code is one in which each digit position has a specific weight or value.
- Examples of weighted codes are 8421(BCD),2421,5211etc.

Binary –Coded Decimal

- Binary-Coded Decimal is a weighted code because each decimal digit can be obtained from its code word by assigning a fixed weight to each code-word bit.
- The weights for the BCD bits are 8, 4, 2, and 1, and for this reason the code is sometimes called the 8421 code.

Non – Weighted Codes

- In non-weighted codes, each digit of the code do not have any position weight.
- Example of non –weighted codes are: ASCII CODE, Excess -3 code, Gray code.

Gray Code

Table 2-10
A comparison of 3-bit binary code and Gray code.

Decimal Number	*Binary Code*	*Gray Code*
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Excess -3 code

- The code word for each decimal digit is the corresponding BCD code word plus 0011_2.
  - 0010 = 2 in BCD
  - + 0011_2
  - = 0101 = 2 in excess-3

Gray Code

- Gray code is a code where only one bit changes at a time while traversing from o to any decimal number in sequence.
- This is a useful property when converting analog values into digital values, since it eliminates the problem of misinterpreting asynchronous changes to bits between valid values.

Parity and its types

- In computers, **parity** is a technique that checks whether data has been lost or written over when it is moved from one place in storage to another or when it is transmitted between computers.

- **TYPES OF PARITY**
- Even parity
- Odd parity

EVEN PARITY

- In this method , one extra bit known as parity bit is added to the binary information.
- The parity is added in such a way that the total number of one's becomes even.

ODD PARITY

- It is similar to even parity method but the total number of one's should be odd.
- Example of Even and Odd parity

Original Data	Even Parity	Odd Parity
0 0 0 0 0 0 0 0	0	1
0 1 0 1 1 0 1 1	1	0
0 1 0 1 0 1 0 1	0	1
1 1 1 1 1 1 1 1	0	1
1 0 0 0 0 0 0 0	1	0
0 1 0 0 1 0 0 1	1	0

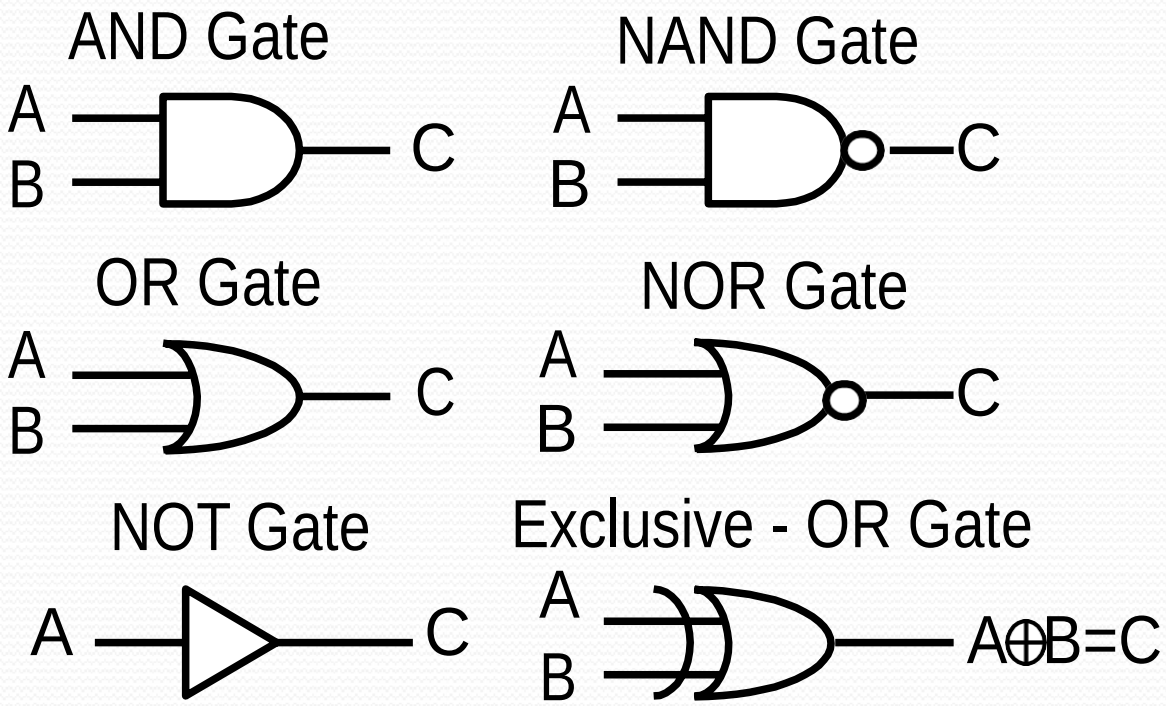
LOGIC GATES AND FAMILIES

CHAPTER- 4

Logic Gates

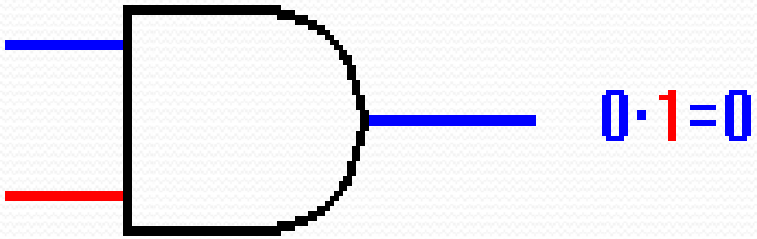
Logic gates are electronic digital circuit perform logic functions. Commonly expected logic functions are already having the corresponding logic circuits in Integrated Circuit (I.C.) form.

Types of Logic Gates



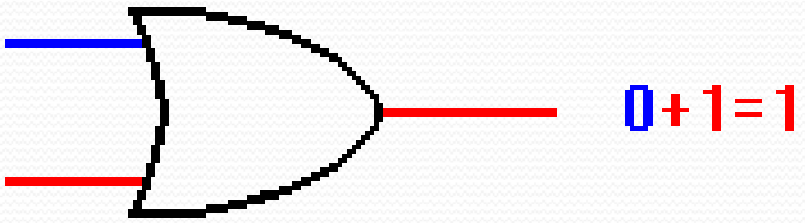
AND Gate

- The AND gate implements the Boolean AND function where the output only is logical 1 when all inputs are logical 1.
- The standard symbol and the truth tabel for a two input AND gate is:



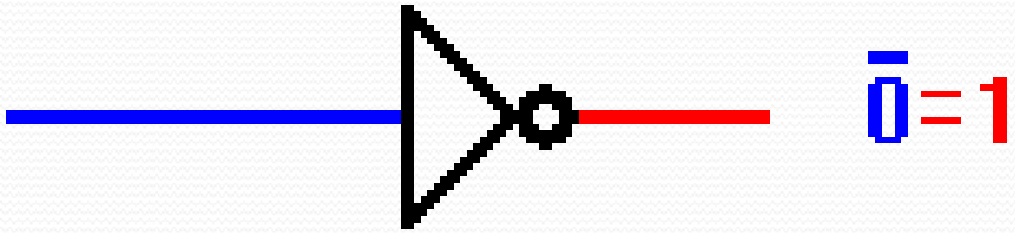
OR Gate

- The OR gate implements the Boolean OR function where the output is logical 1 when just input is logical 1.
- The standard symbol and the truth table for a two input OR gate is:



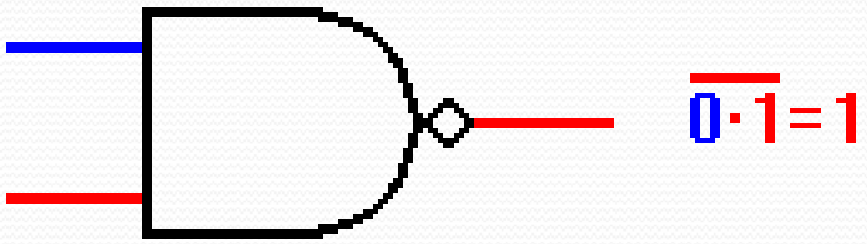
NOT Gate

- The NOT gate implements the Boolean NOT function where the output is the inverse of the input.
- The standard symbol and the truth table for the NOT gate is:



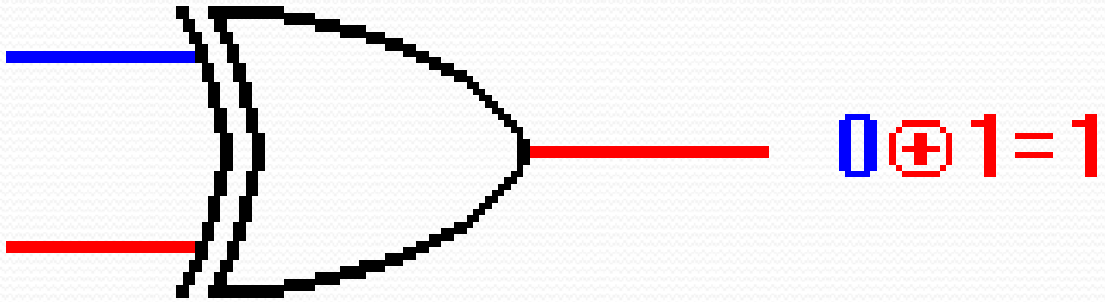
NAND Gate

- The NAND gate is an AND gate followed by a NOT gate. The output is logical 1 when one of the inputs are logical 0
- The standard symbol and the truth table for the NAND gate is:



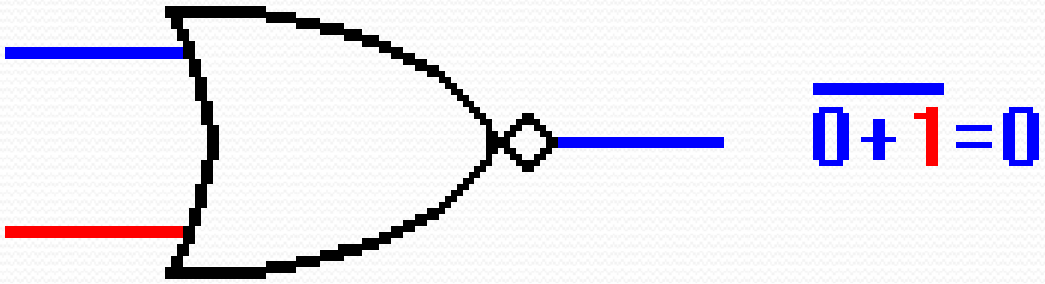
XOR Gate

- The XOR gate produces a logic 1 output only if its two inputs are *different*. If the inputs are the same, the output is a logic 0
- The XOR symbol is a variation on the standard OR symbol. It consists of a plus (+) sign with a circle around it. The logic symbol, as shown here, is a variation on the standard OR symbol.



NOR Gate

- The NOR is a combination of an OR followed by a NOT gate. The output is logical 1 when non of the inputs are logical 0
- The standard symbol and the truth table for the NOR gate is:

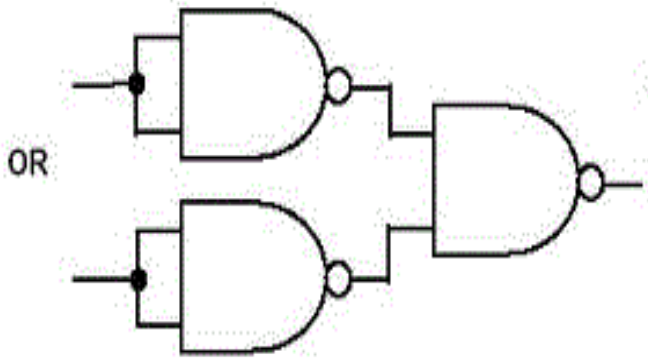


UNIVERSAL GATES

- A universal gate is actually **NAND and NOR gate**. It is simply because they can be used to construct other gates. To build big and complex digital system, we only use NAND and NOR gate.

NAND GATE AS UNIVERSAL GATE

- NAND gate is used to make OR gate , NOT gate. These gate help us in making al the gates for eg: NAND gate as NC



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

NOR GATE AS UNIVERSAL GATE

- NOR gate is used for make OR gate , NOT gate . with the help of these gate we can make any gate for ex : NOR gate as OR gate.

Introduction to TTL logic family

- The full form of TTL is Transistor Transistor logic . The digital ICs in the TTL family used only transistor to perform the basic logic operations . TTL families are used as more complex devices in digital system .

TTL CHARACTERISTICS

- **Transistor-transistor logic (TTL)**
  - based on bipolar transistors
  - one of the most widely used families for small- and medium-scale devices – rarely used for VLSI
  - typically operated from 5V supply
  - typical noise immunity about 1 – 1.6 V
  - many forms, some optimised for speed, power, etc.
  - high speed versions comparable to CMOS (~ 1.5 ns)
  - low-power versions down to about 1 mW/gate

Introduction to CMOS logic family

- CMOS stands for Complementary Metal Oxide Semiconductor. A complementary pair uses both p or n channel MOSFETs.

CMOS CHARACTERISTICS

- **Complementary metal oxide semiconductor (CMOS)**
  - most widely used family for large-scale devices
  - combines high speed with low power consumption
  - usually operates from a single supply of 5 – 15 V
  - excellent noise immunity of about 30% of supply voltage
  - can be connected to a large number of gates (about 50)
  - many forms – some with t_{PD} down to 1 ns
  - power consumption depends on speed (perhaps 1 mW

LOGIC SIMPLIFICATION

CHAPTER- 5

INTRODUCTION

- ALL the digital circuit operation depend upon only two values that is either 1 or 0 . Where the value of 1 and 0 denoted the predefined voltage level . So , Boolean algebra can used for the analysis , simplification design of digital circuit .
- There are three operation in Boolean Algebra :
- Logical addition
- Logical multiplication
- Logical inversion

Postulates OF Boolean Algebra

- A statement that is not proved but assumed to be true is called the postulates .
- The basic postulates of Boolean Algebra are:
- Commutative laws
- Associative laws
- Distributed laws
- Identity rule
- Complement rule

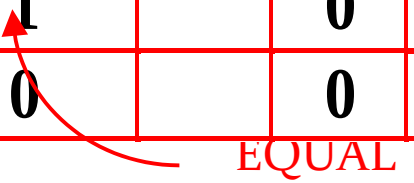
DEMORGAN'S THEOREMS

- A great mathematician named demorgan gives two theorems of Boolean Algebra . These theorem are very useful and powerful identities used in Boolean Algebra.

DeMorgan's Theorem #1

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

A	B	A • B	$\overline{A \cdot B}$		A	B	A + B
0	0	0	1		1	1	1
0	1	0	1		1	0	1
1	0	0	1		0	1	1
1	1	1	0		0	0	0



DeMorgan's Theorem #2

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

A	B	A + B	$\overline{A + B}$		A	B	$\overline{A} \cdot \overline{B}$
0	0	0	1		1	1	1
0	1	1	0		1	0	0
1	0	1	0		0	1	0
1	1	1	0		0	0	0

EQUAL

KARNAUGH MAPS (K-MAP)

- An n-variable k-map has two cell with each cell corresponding to an n-variable truth table value .
- K-Map cell are labeled with the corresponding truth table row .
- K-Map cells are arranged such that adjacent cells correspond to truth rows that differ in only one bit position (logical adjacency) .

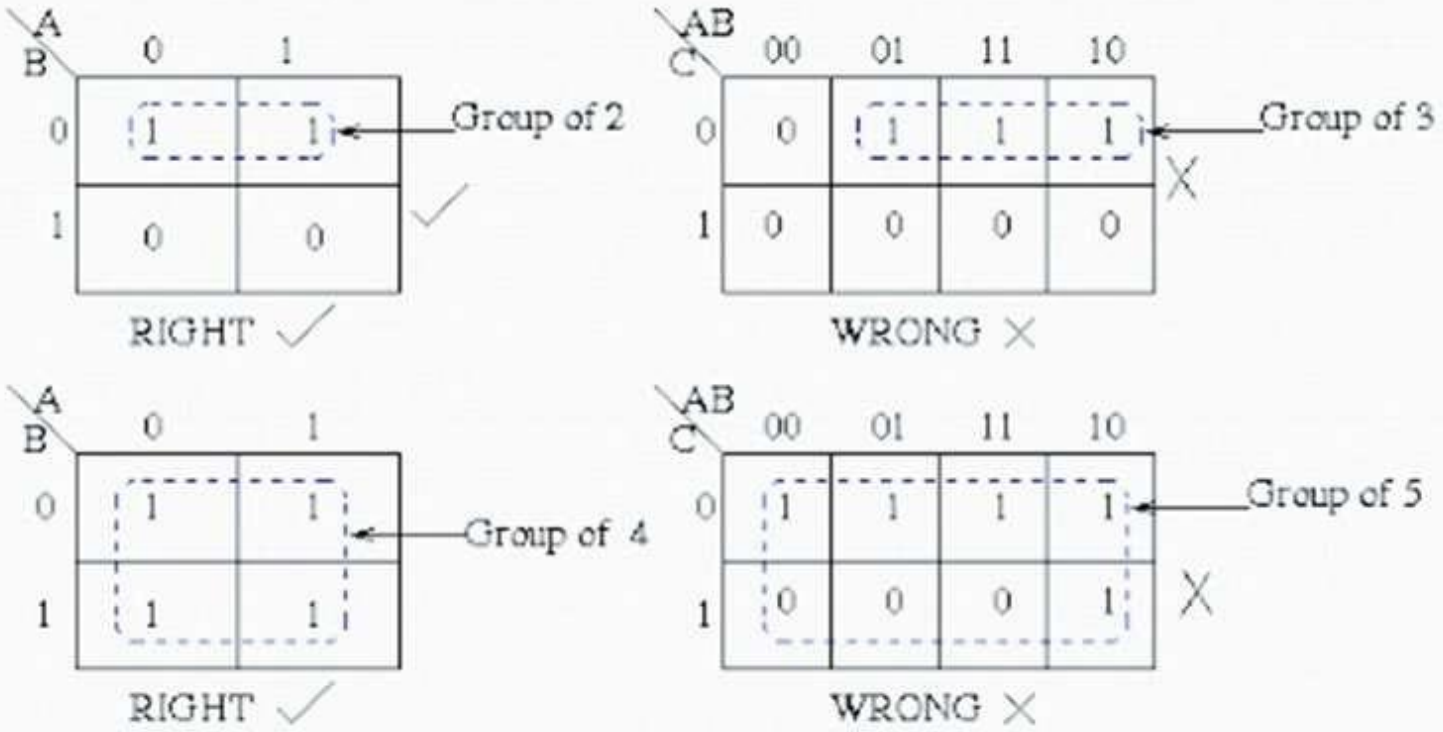
KARNAUGH MAPS (K-MAPS)

- K-Map – A tool for representing Boolean function up to six variables .
- K-Map are tables of row and columns with entries represent 1's or 0's of SOP and POS representation .

3. Groups must contain 1, 2, 4, 8, or in general 2^n cells.

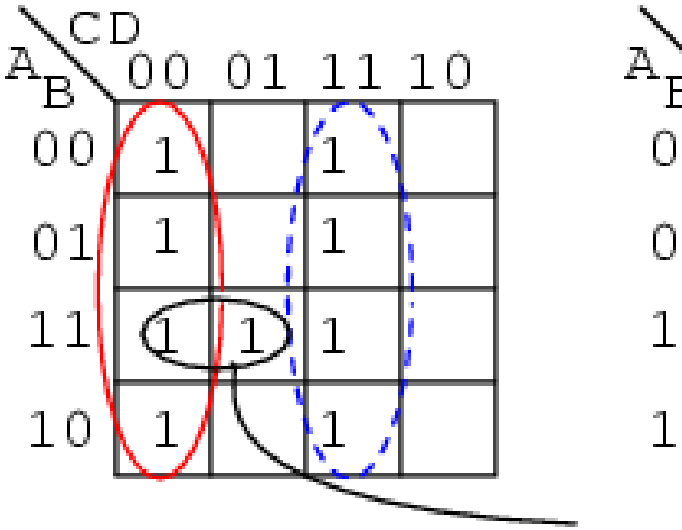
That is if n = 1, a group will contain two 1's since $2^1 = 2$.

If n = 2, a group will contain four 1's since $2^2 = 4$.

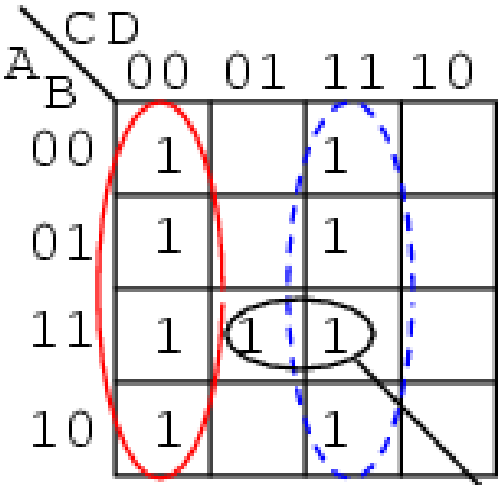


Example – 4 variable K-Map

$$Out = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C\,D + \overline{A}\,B\,\overline{C}\,\overline{D} + \overline{A}\,B\,C\,D + A\,B\,\overline{C}\,\overline{D} + A\,B\,\overline{C}\,D + A\,B\,C\,D + A\,\overline{B}\,\overline{C}\,\overline{D} + A\,\overline{B}\,C\,D$$



$$Out = \overline{C}\,\overline{D} + C\,D + A\,B\,\overline{C}$$



$$Out = \overline{C}\,\overline{D} + C\,D + A\,B\,D$$

Advantages and Disadvantages of K-Map

Advantages :

1. Minimizes boolean expressions without the need using various boolean theorems & computations.
2. Minimizes number of Logical gates used.

Disadvantages :

1. Become tedious for 5 variables.
2. It is not suitable for computer reduction.

ARITHMETIC CIRCUITS

CHAPTER- 6

Combinational Arithmetic Circuits

- Addition:
  - **Half Adder (HA).**
  - **Full Adder (FA).**
- Subtraction:
  - **Half Subtractor.**
  - **Full Subtractor.**
- 4 bit adder/subtracter
- Adder and Subtractor IC (7484)

HALF ADDER

Adding two single-bit binary values, X, Y produces a sum S bit and a carry out C-out bit. This operation is called half addition and the circuit to realize it is called a half adder.

Half Adder Truth Table

Inputs		Outputs	
X	Y	S	C-out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

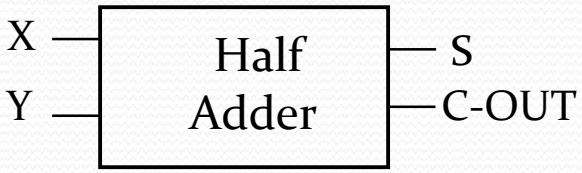
$S(X,Y) = \Sigma\ (1,2)$

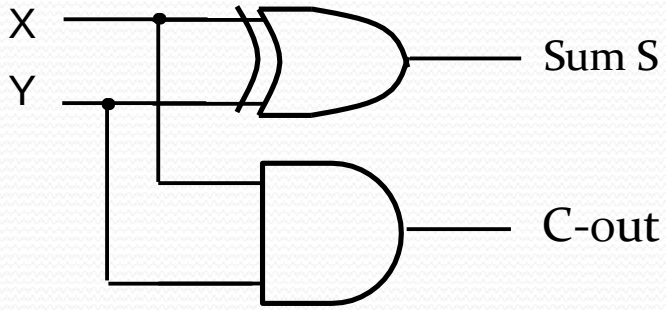
$S = X'Y + XY'$

$S = X \oplus Y$

$C\text{-out}(x, y, C\text{-in}) = \Sigma\ (3)$

$C\text{-out} = XY$





FULL ADDER

Adding two single-bit binary values, X, Y with a carry input bit C-in produces a sum bit S and a carry out C-out bit.

Full Adder

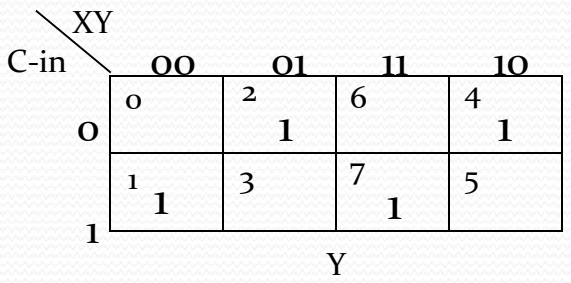
Full Adder Truth Table

Inputs			Outputs	
X	Y	C-in	S	C-out
0	0	0	0	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

$S(X,Y, C\text{-in}) = \Sigma\ (1,2,4,7)$

$C\text{-out}(x, y, C\text{-in}) = \Sigma\ (3,5,6,7)$

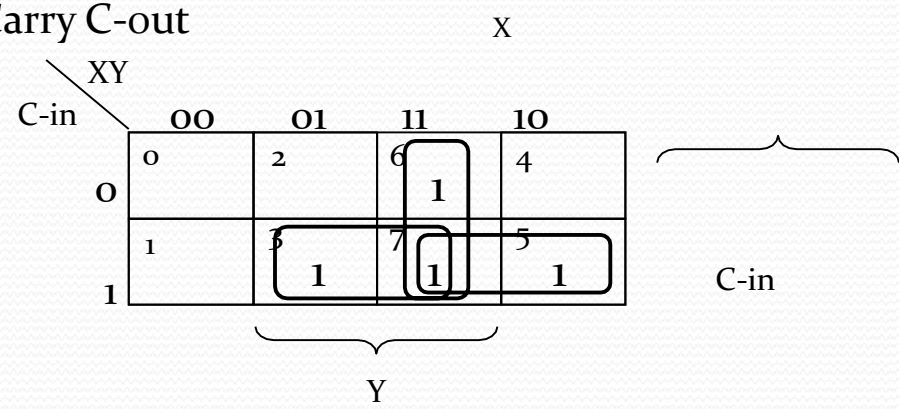
Sum S



$S = X'Y'(C\text{-in}) + XY'(C\text{-in})' + XY'(C\text{-in})' + XY(C\text{-in})$

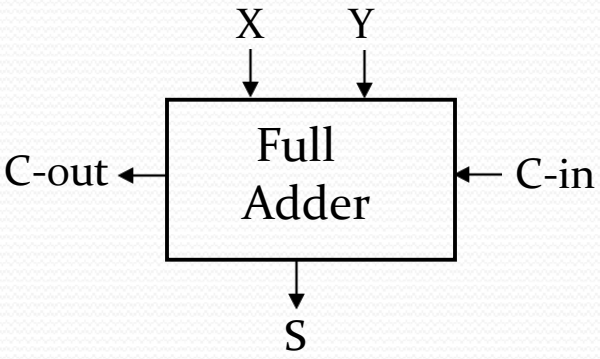
$S = X \oplus Y \oplus (C\text{-in})$

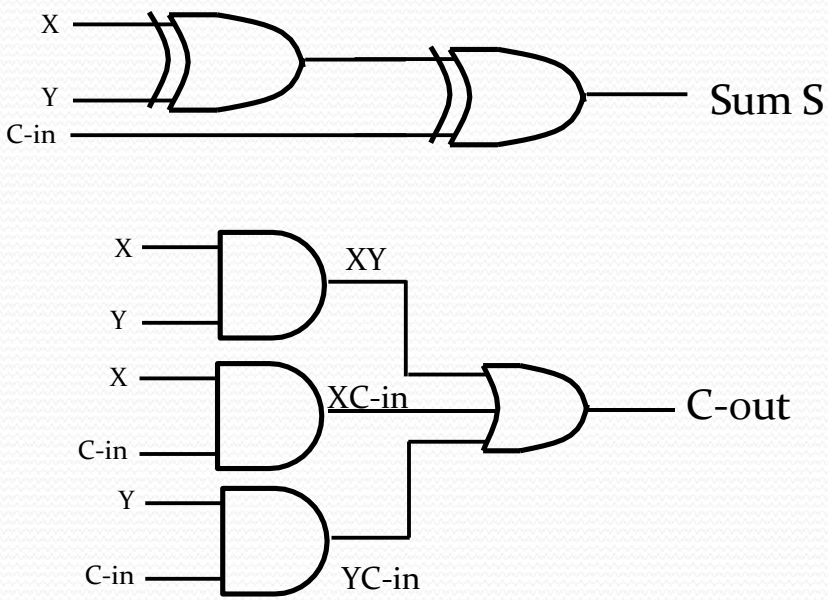
Carry C-out



$C\text{-out} = XY + X(C\text{-in}) + Y(C\text{-in})$

Full Adder Circuit Using XOR





HALF SUBTRACTOR

- Subtracting a single-bit binary value Y from anther X (I.e. X -Y) produces a difference bit D and a borrow out bit B-out.
- This operation is called half subtraction and the circuit to realize it is called a half subtractor.

Half Subtractor

Half Subtractor Truth Table

Inputs		Outputs	
X	Y	D	B-out
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

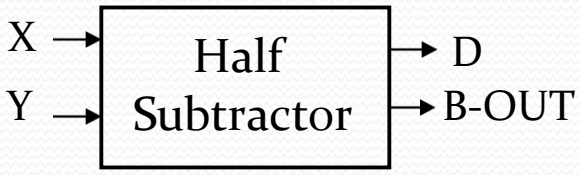
$D(X,Y) = \Sigma\,(1,2)$

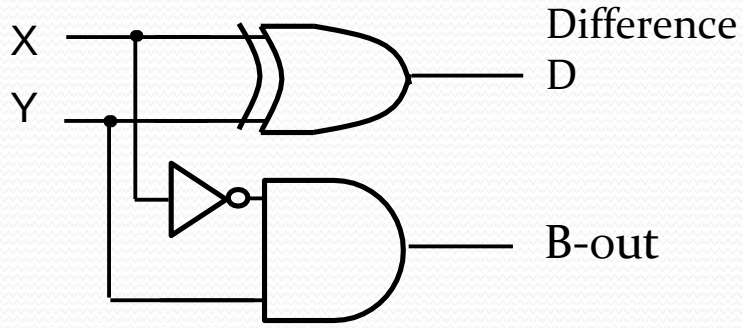
$D = X'Y + XY'$

$D = X \oplus Y$

$B\text{-out}(x, y, C\text{-in}) = \Sigma\,(1)$

$B\text{-out} = X'Y$





FULL SUBTRACTOR

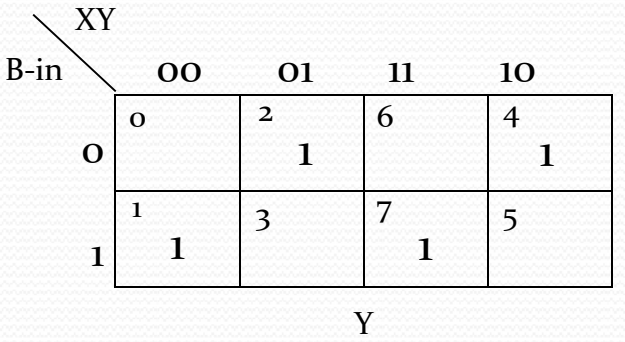
- Subtracting two single-bit binary values, Y, B-in from a single-bit value X produces a difference bit D and a borrow out B-out bit. This is called full subtraction.

Full Subtractor

Full Subtractor Truth Table

Inputs			Outputs	
X	Y	B-in	D	B-out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

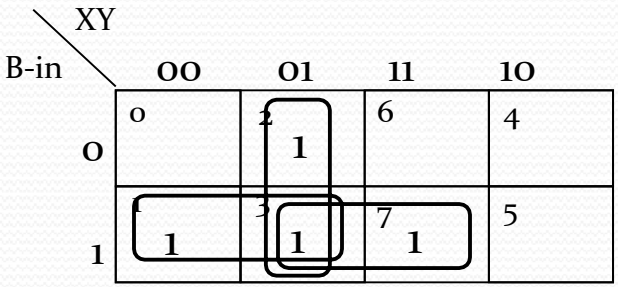
Difference D



$S = X'Y'(B\text{-in}) + XY'(B\text{-in})' + XY'(B\text{-in})' + XY(B\text{-in})$

$S = X \oplus Y \oplus (C\text{-in})$

Borrow B-out

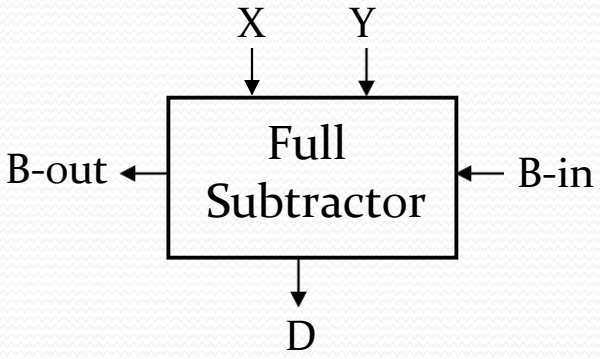


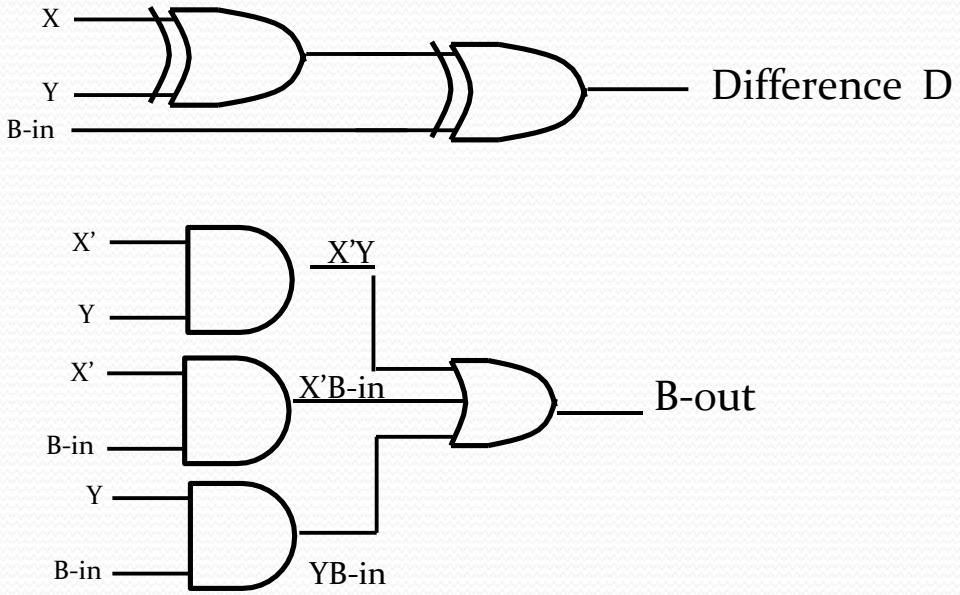
$S(X,Y, C\text{-in}) = \Sigma\,(1,2,4,7)$

$C\text{-out}(x, y, C\text{-in}) = \Sigma\,(1,2,3,7)$

$B\text{-out} = X'Y + X'(B\text{-in}) + Y(B\text{-in})$

Full Subtractor Circuit Using XOR

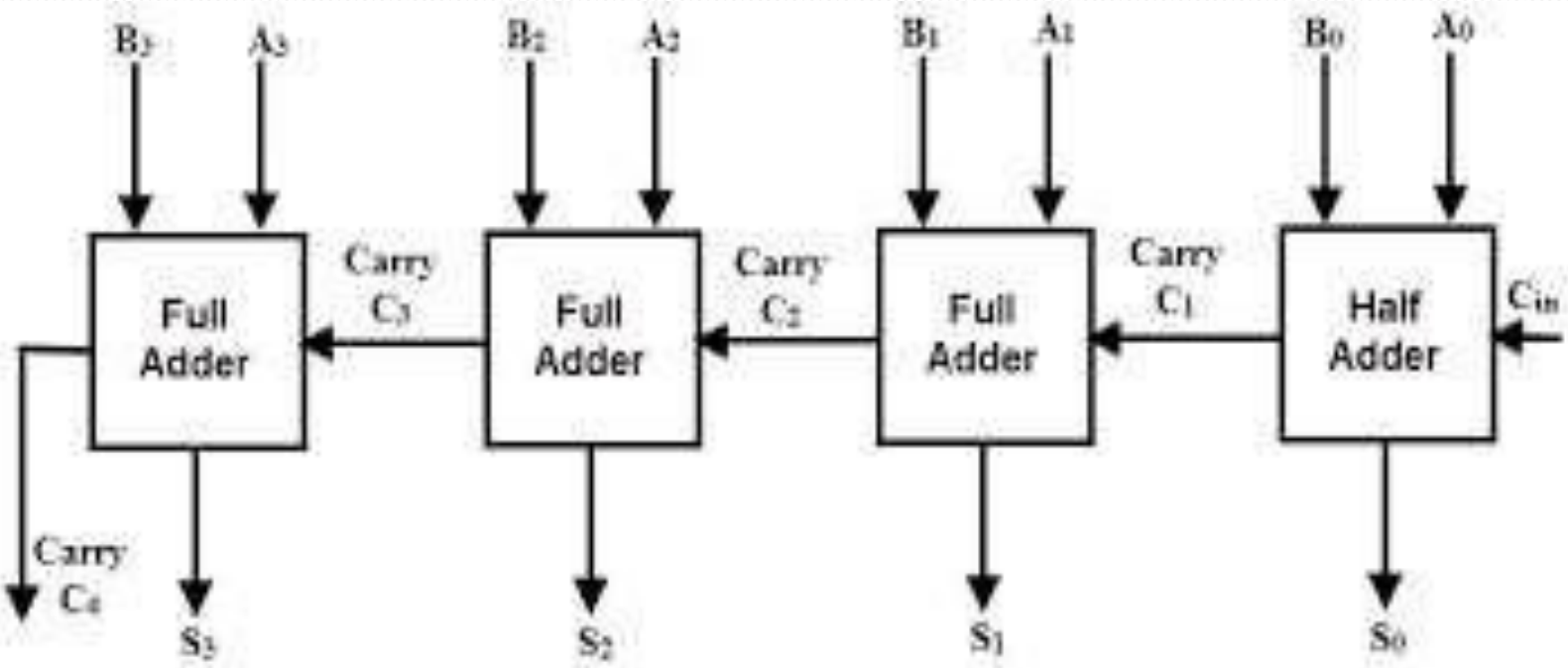




PARALLEL BINARY ADDER

- These adders are constructed by connecting two or more full adders.
- A single full adder is used for adding two one-bit binary numbers and an input carry.
- For the addition of binary numbers having more than one bit, additional full adders must be used.

Block diagram of 4- bit binary Parallel Adder



4-bit Parallel Adder

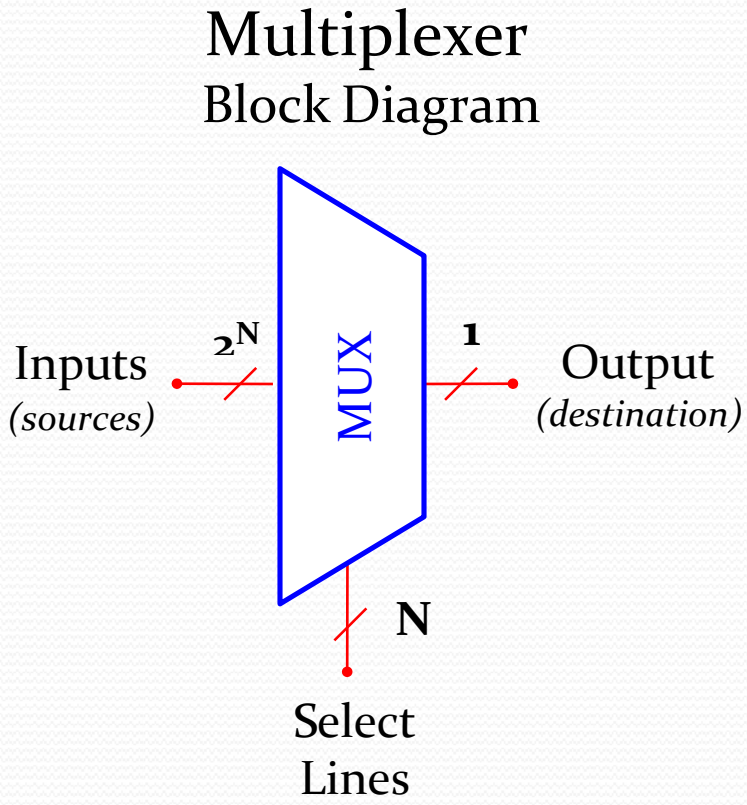
- For 4-bit binary full adder, we require four full adders(or three full adders and one half adder).
- The carry output of the rightmost full adder becomes the carry input of the second full adder.
- Let the two four bit binary numbers are A_3 A_2 A_1 A_0 and B_3 B_2 B_1 B_0 and five sum bits S_3 S_2 S_1 S_0 .
- The carry output of the leftmost full adder becomes the MSB in the sum which is represented by S_4.

Decoders, Multiplexers, De-Multiplexers and Encoder

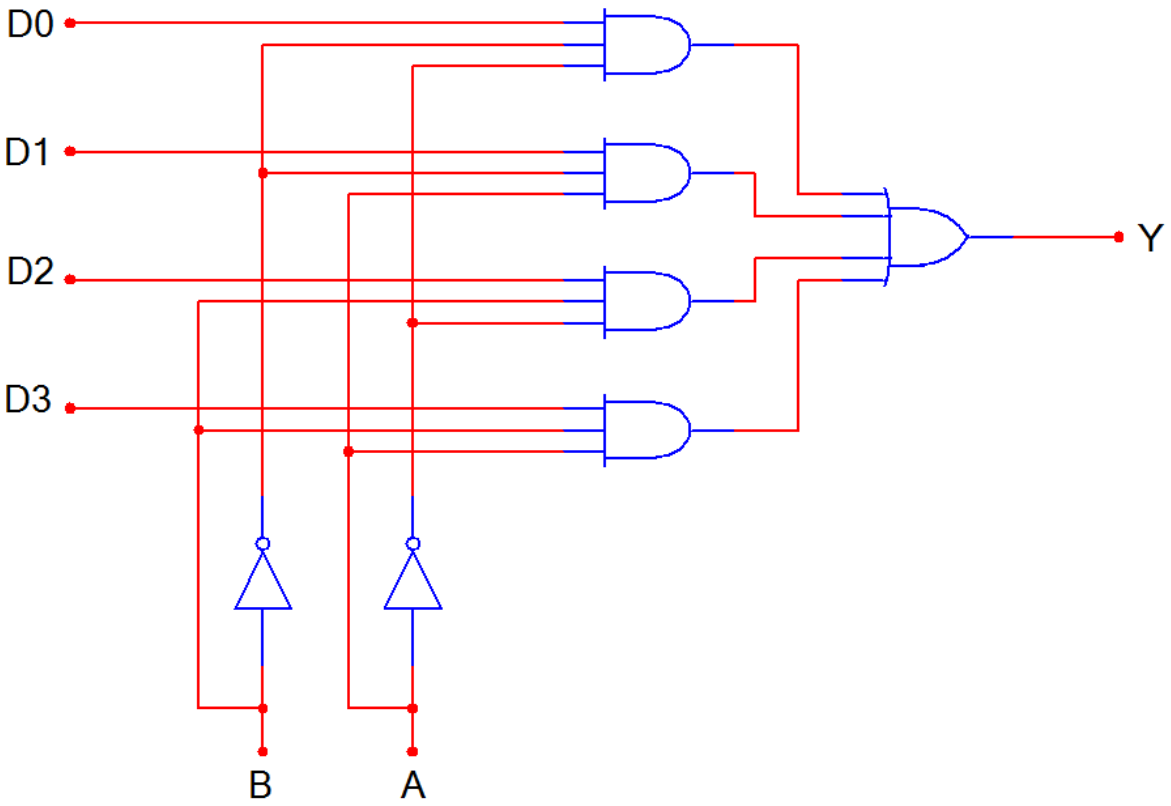
CHAPTER- 7

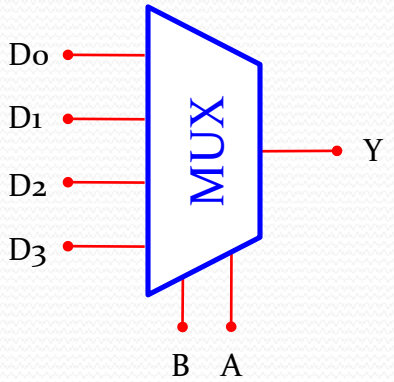
What is a Multiplexer (MUX)?

- A MUX is a digital switch that has multiple inputs (sources) and a single output (destination).
- The select lines determine which input is connected to the output.
- MUX Types
  - → 2-to-1 (1 select line)
  - → 4-to-1 (2 select lines)
  - → 8-to-1 (3 select lines)
  - → 16-to-1 (4 select lines)



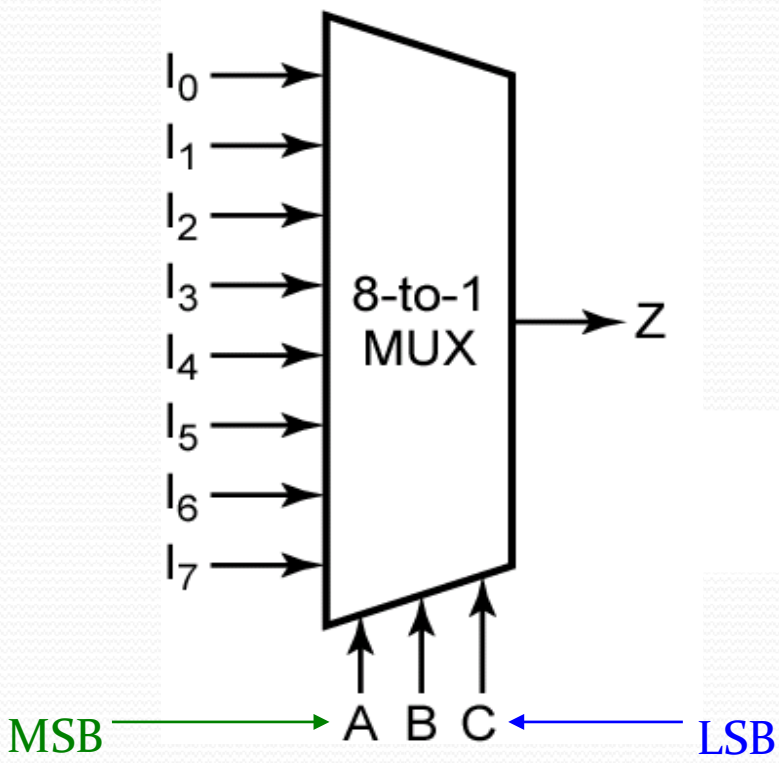
4-to-1 Multiplexer (MUX)





B	A	Y
0	0	Do
0	1	D1
1	0	D2
1	1	D3

Multiplexers

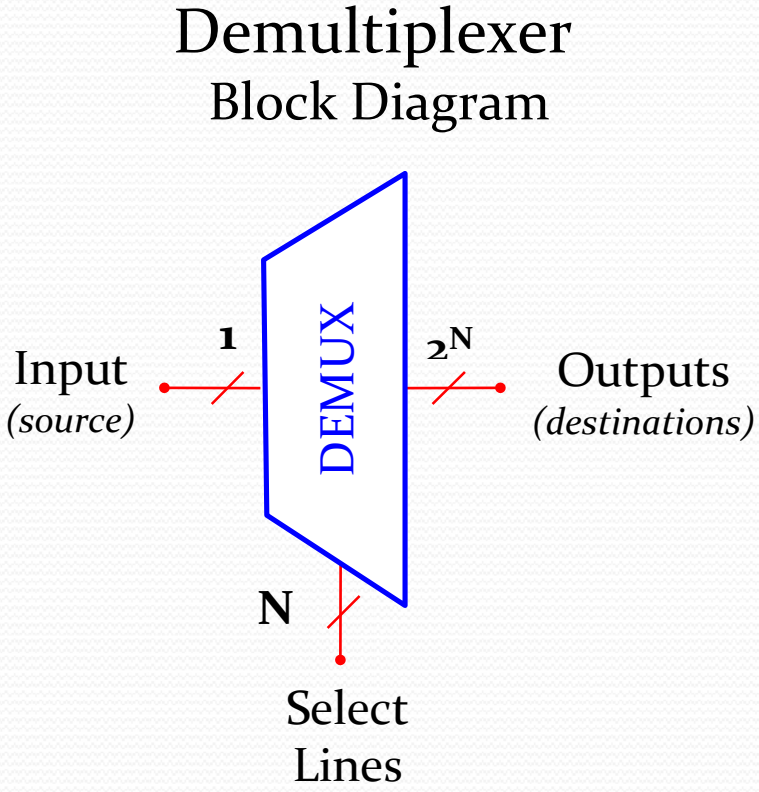


A	B	C	F
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

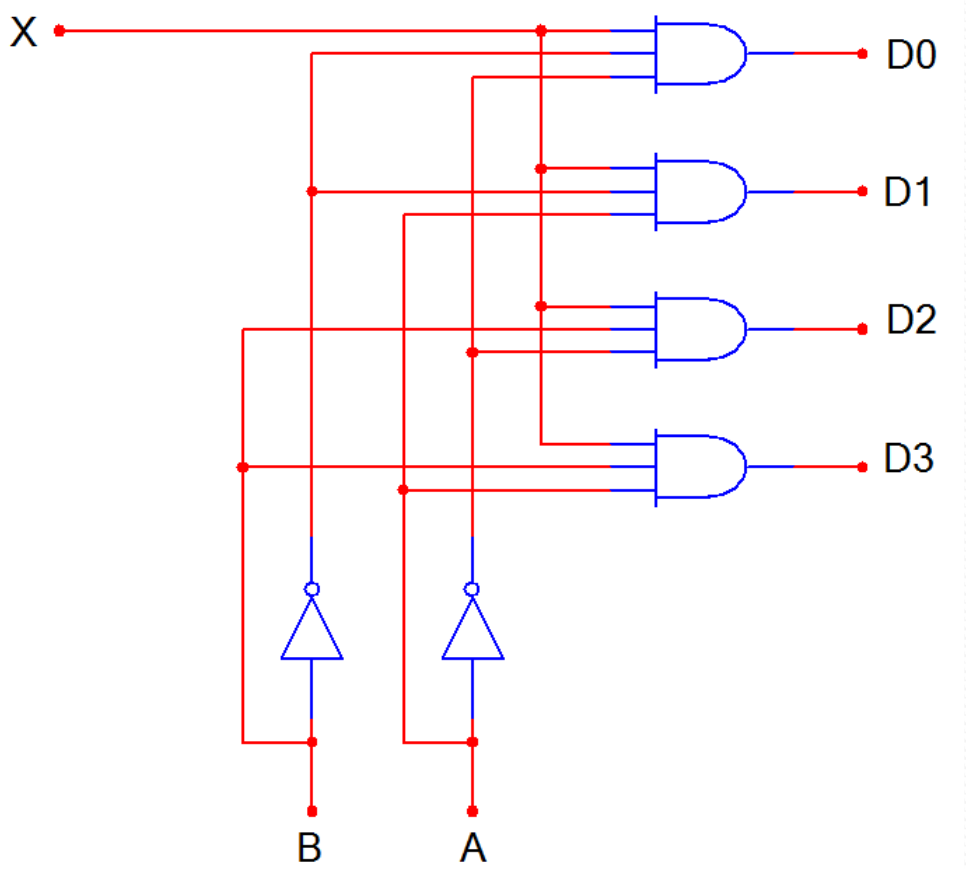
$$Z = A'.B'.C'.I_0 + A'.B'.C.I_1 + A'.B.C'.I_2 + A'.B.C.I_3 + A.B'.C'.I_0 + A.B'.C.I_1 + A'.B.C'.I_2 + A.B.C.I_3$$

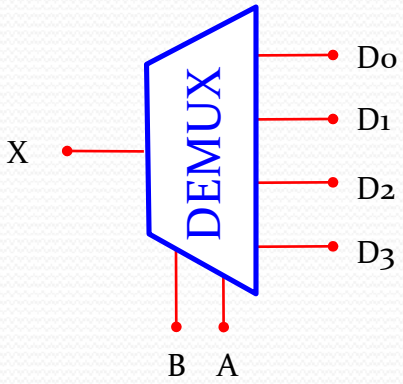
What is a Demultiplexer (DEMUX)?

- A DEMUX is a digital switch with a single input (source) and a multiple outputs (destinations).
- The select lines determine which output the input is connected to.
- DEMUX Types
  - → 1-to-2 (1 select line)
  - → 1-to-4 (2 select lines)
  - → 1-to-8 (3 select lines)
  - → 1-to-16 (4 select lines)



1-to-4 De-Multiplexer (DEMUX)





B	A	Do	D1	D2	D3
0	0	X	0	0	0
0	1	0	X	0	0
1	0	0	0	X	0
1	1	0	0	0	X

Decoders

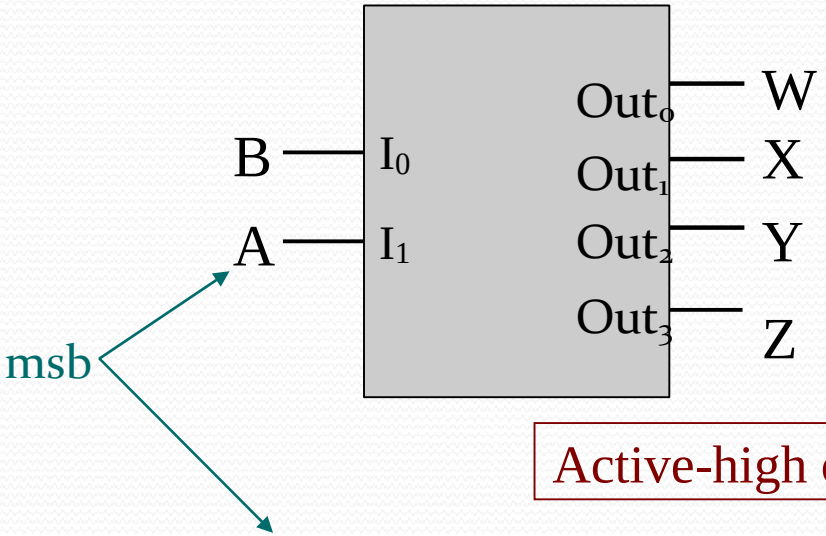
A decoder has

- N inputs
- 2^N outputs

A decoder selects one of 2^N outputs by decoding the binary value on the N inputs.

The decoder generates all of the minterms of the N input variables.

Decoders



W = A'.B'

X = A.B'

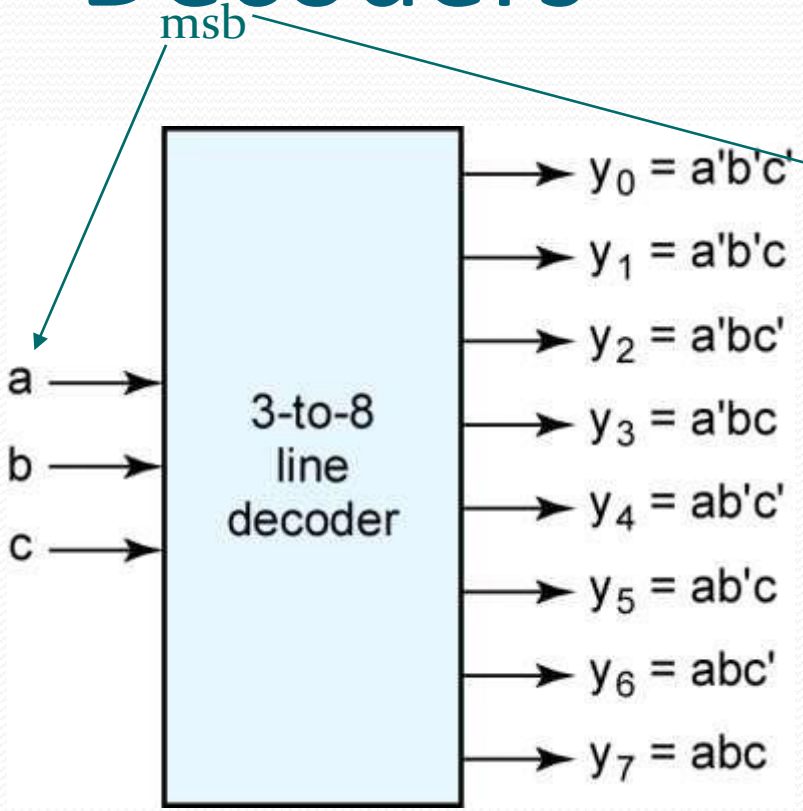
Y = A'.B

Z = A.B

Active-high outputs

A	B	W	X	Y	Z
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Decoders



a	b	c	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_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

Encoders

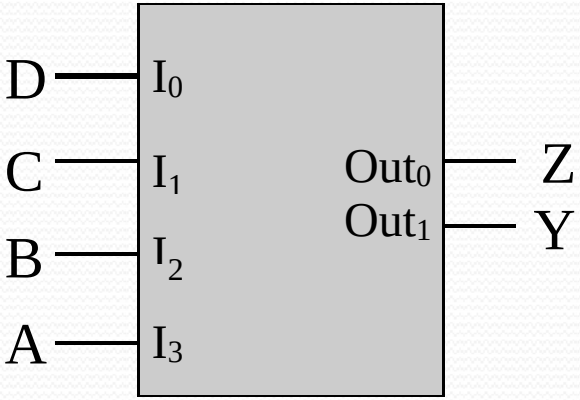
An encoder has

- 2^N inputs
- N outputs

An encoder outputs the binary value of the selected (or active) input.

An encoder performs the inverse operation of a decoder.

Encoders

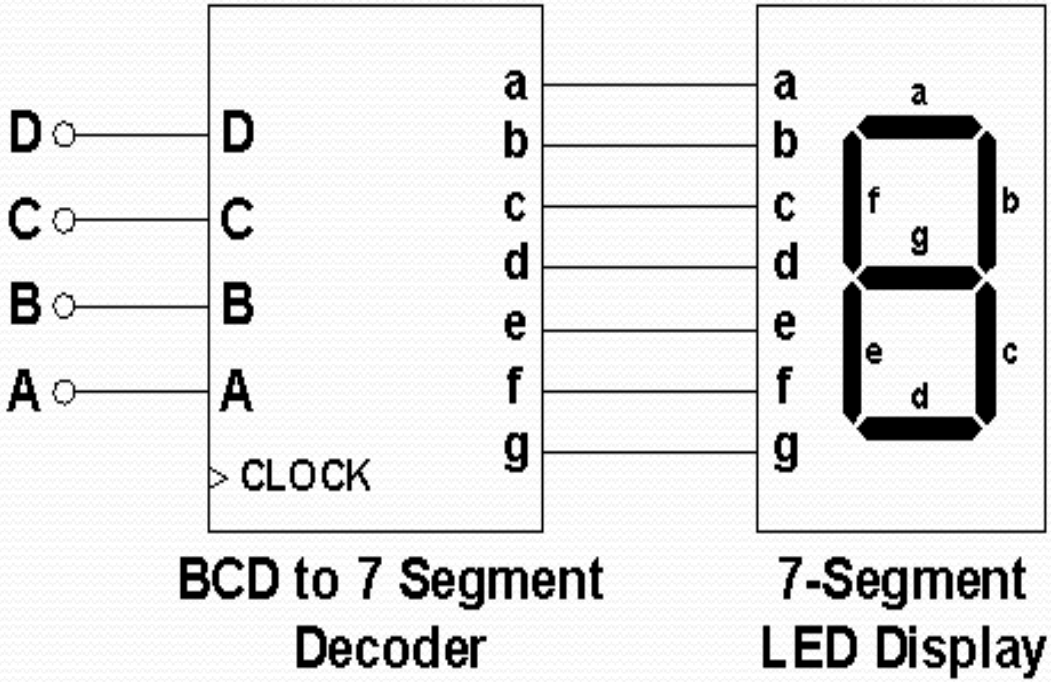


A	B	C	D	**Y**	**Z**
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

BCD TO SEVEN SEGMENT DECODER

- In this display device, the data which is in the BCD format has to be changed suitably.
- For this purpose, we require a BCD to 7- segment decoder.
- The circuit has four input lines for receiving the BCD inputs & seven output lines i.e. a,b,c,d,e,f,g to drive a 7-segment display.

BLOCK DIAGRAM OF BCD TO 7-SEGMENT DECODER



TRUTH TABLE OF BCD TO 7 SEVEN SEGMENT DECODER

Binary Inputs				Decoder Outputs							7-Segment Display Outputs
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

LATCHES AND FLIP - FLOPS

CHAPTER -8

What is latch ?

- A flip-flop or **latch** is a circuit that has two stable states and can be used to store state information.
- It is a sequential circuit.
- It is a temporary storage device.

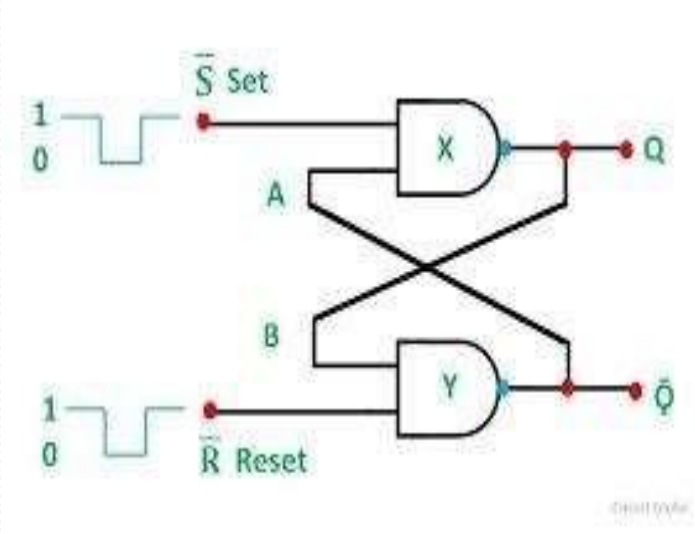
FLIP FLOP

- A flip-flop is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs.
- Flip Flops are sequential circuits.
- It also stores memory.

TYPES OF FLIP-FLOP

- SR flip-flop
- T flip-flop
- D flip-flop
- JK flip-flop

SR- LATCH



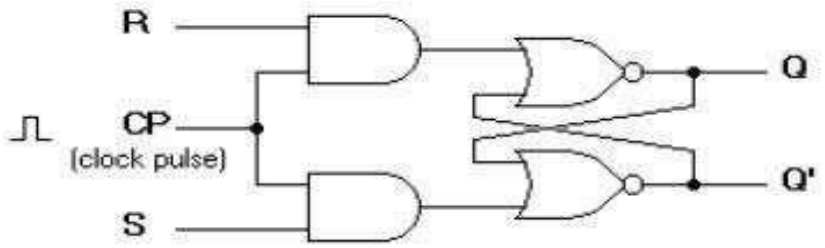
TRUTH TABLE OF SR-LATCH

Sno	S	R	Q	Q'	State
1	1	0	1	0	Q is set to 1
2	1	1	1	0	No change
3	0	1	0	1	Q' is set to 1
4	1	1	0	1	No change
5	0	0	1	1	Invalid

SR FLIP FLOP

- **SR Flip Flop** is an arrangement of logic gates that maintains a stable output even after the inputs are turned off.
- This simple **flip flop** circuit has a set input (S) and a reset input (R).

SR FLIP FLOP



(a) Logic diagram

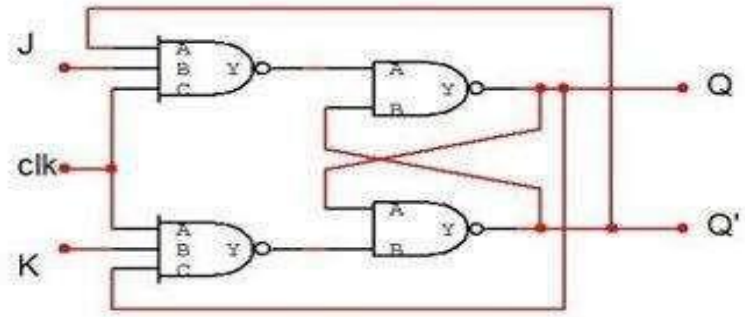
Q	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

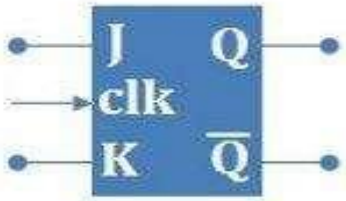
(b) Truth table

Clocked SR flip-flop

JK FLIP FLOP

- The **JK flip flop** is basically a gated SR **flip-flop** with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level “1”.
- There is no such thing as a J-K latch, only J-K **flip-flops**.
- Without the edge-triggering of the clock input, the circuit would continuously **toggle** between its two output states when both J and K were held high (1), making it an astable device instead of a bistable device in that circumstance.



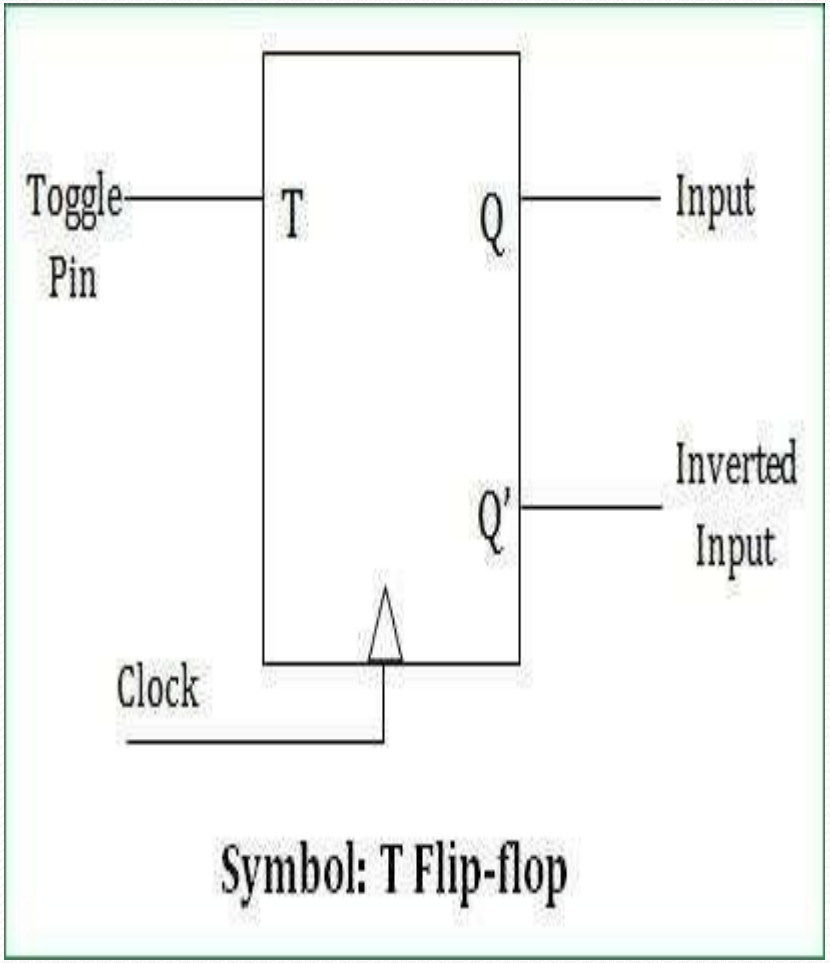


J	K	clk	Q
0	0	1	Önceki konum
0	1	1	0
1	0	1	1
1	1	1	toggle

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

T or Toggle flip -flop

- The **T or "toggle" flip-flop** changes its output on each clock edge, giving an output which is half the frequency of the signal **to** the **T** input.
- It is useful for constructing binary counters, frequency dividers, and general binary addition devices.
- It can be made from a J-K **flip-flop** by tying both of its inputs high.

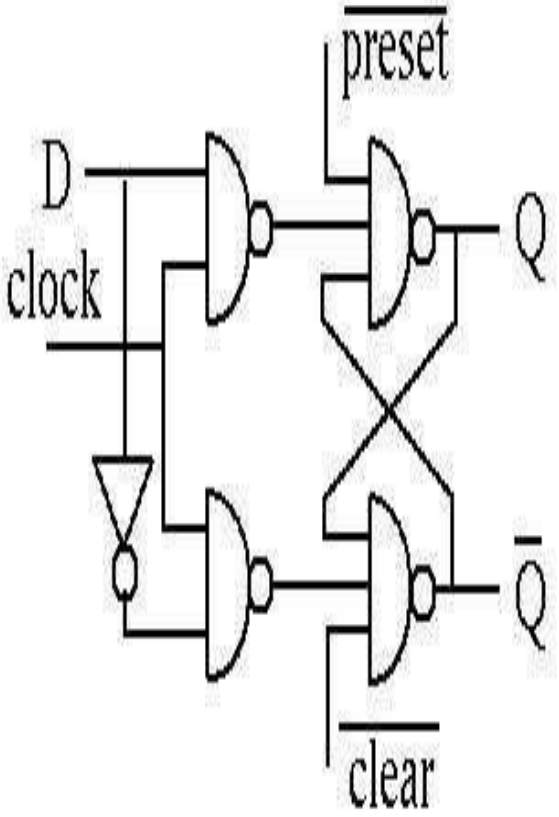


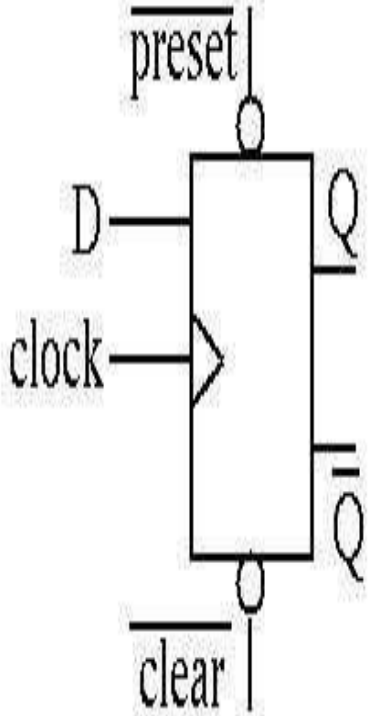
Symbol: T Flip-flop

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

D FLIP FLOP

- The **D flip-flop** tracks the input, making transitions with match those of the input **D**.
- The **D** stands for "data"; this **flip-flop** stores the value that is on the data line.
- It can be thought of as a basic memory cell. A D **flip-flop** can be made from a set/reset **flip-flop** by tying the set to the reset through an inverter.





D	Q_{t+1}
0	0
1	1

APPLICATIONS OF FLIP FLOP

- **flip flop** circuit mainly involves in bounce elimination switch.
- data storage, data transfer.
- latch, registers, counters.
- frequency division, memory, etc.

APPLICATIONS OF LATCHES

- Cascading of a positive latch and negative latch gives a negative edge-triggered flip-flop and cascading of negative and positve latch gives a positive edge-triggered flip-flop.
- A latch is used as a savior for scan hold timing closure in the form of lockup latch.
- Latch is used in pipelines.

DIFFERENCE

LATCH

- Gates are the building block of the latches.
- Latches does not have clock signal.
- It is a level triggered device.
- It is based on the enable function input.

FLIP FLOP

- Latches are the building block of Flip Flop.
- Flip Flop has clock signal.
- It is edge triggered.
- It works on the clock pulses.

COUNTERS

CHAPTER- 9

INTRODUCTION

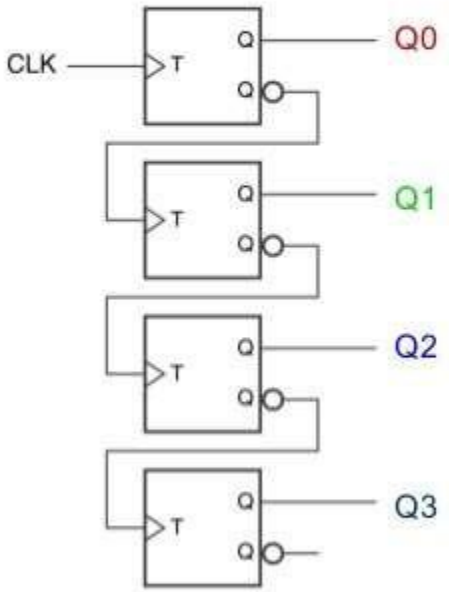
- Counter is the combination of flip-flop which is used to count the events of number of clock at input. Depending upon the manner by mean of which the flip-flop of counter triggered.
  - There is two types of counters:
    - Asynchronous Counter
    - Synchronous Counter

ASYNCHRONOUS COUNTER

- Asynchronous counters are **those whose output is free from the clock signal**. Because the flip flops in asynchronous counters are supplied with different clock signals, there may be delay in producing output. The required number of logic gates to design asynchronous counters is very less. So they are simple in design.

ASYNCHRONOUS RIPPLE COUNTER

Asynchronous Ripple Counter

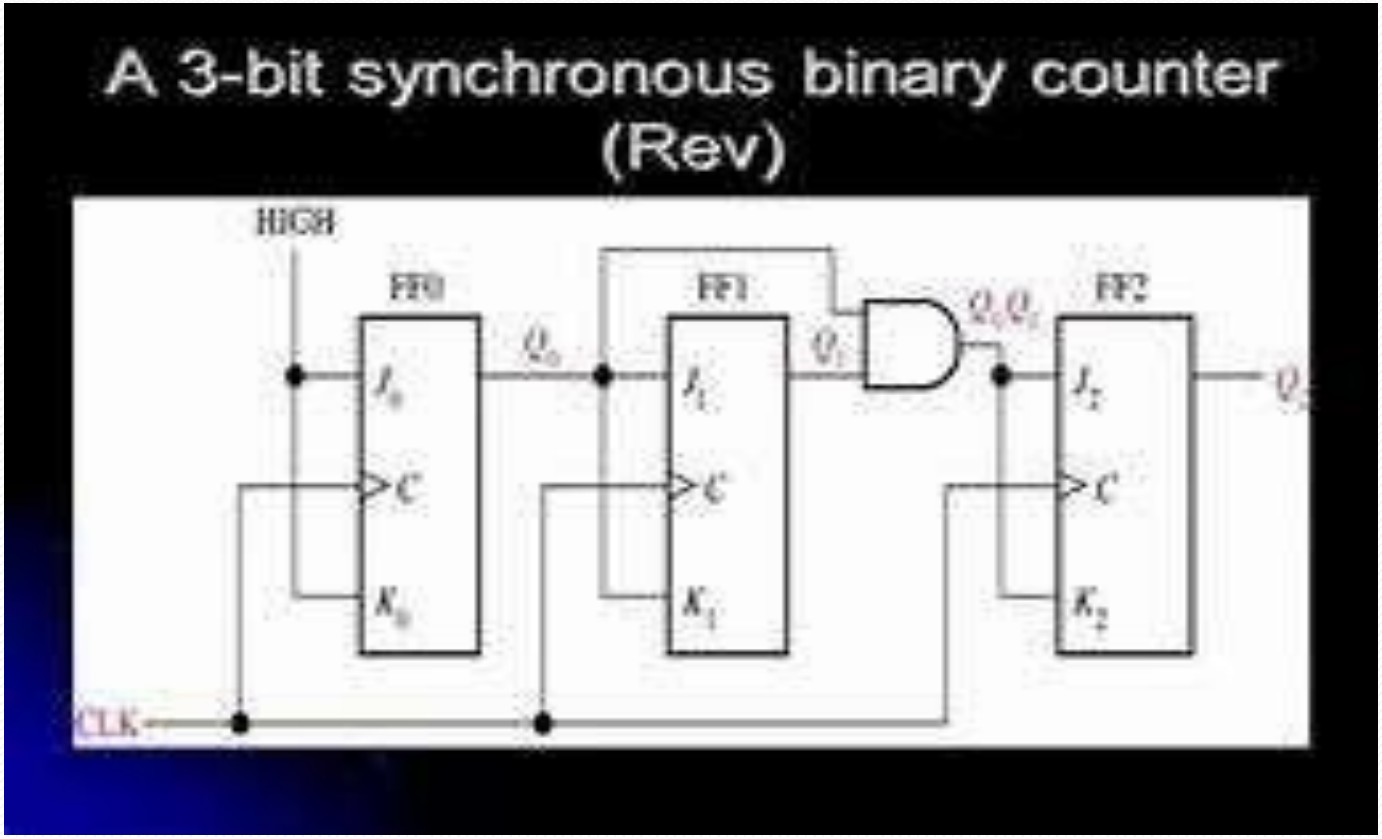


Q3	Q2	Q1	Q0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
⋮			

SYNCHRONOUS COUNTER

- In synchronous counters, the **clock inputs of all the flip-flops** are connected together and are triggered by the input pulses. Thus, all the flip-flops change state simultaneously (in parallel).

3 BIT SYNCHRONOUS COUNTER

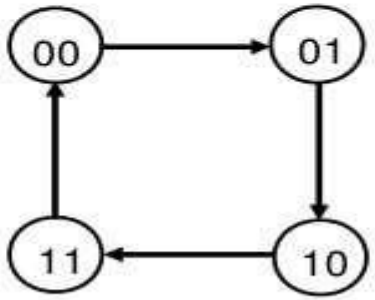


APPLICATION OF COUNTER

Counters

- Applications include:
  - system clock
  - timer, delays
  - watches, clocks, alarms
  - counting events
  - memory addressing
  - frequency division
  - sequence control
  - cycle control
  - protocols

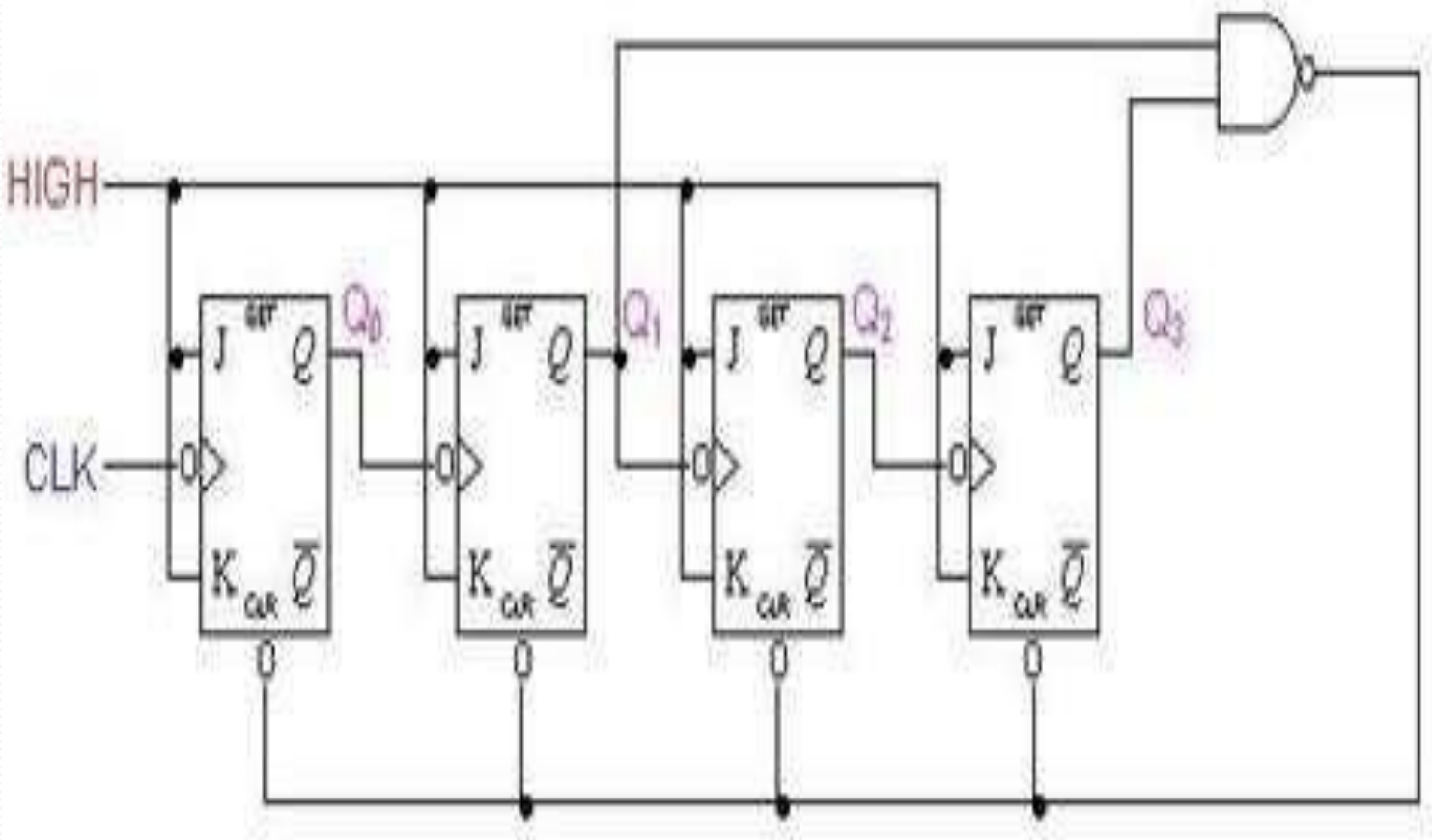
Present State		Next State	
A	B	A	B
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



DECADE COUNTER

- A decade counter is one that counts in decimal digits, rather than binary. It counts from **0 to 9** and then resets to **zero**. The counter output can be set to zero by pulsing the reset line low. The count then increments on each clock pulse until it reaches 1001 (decimal 9).

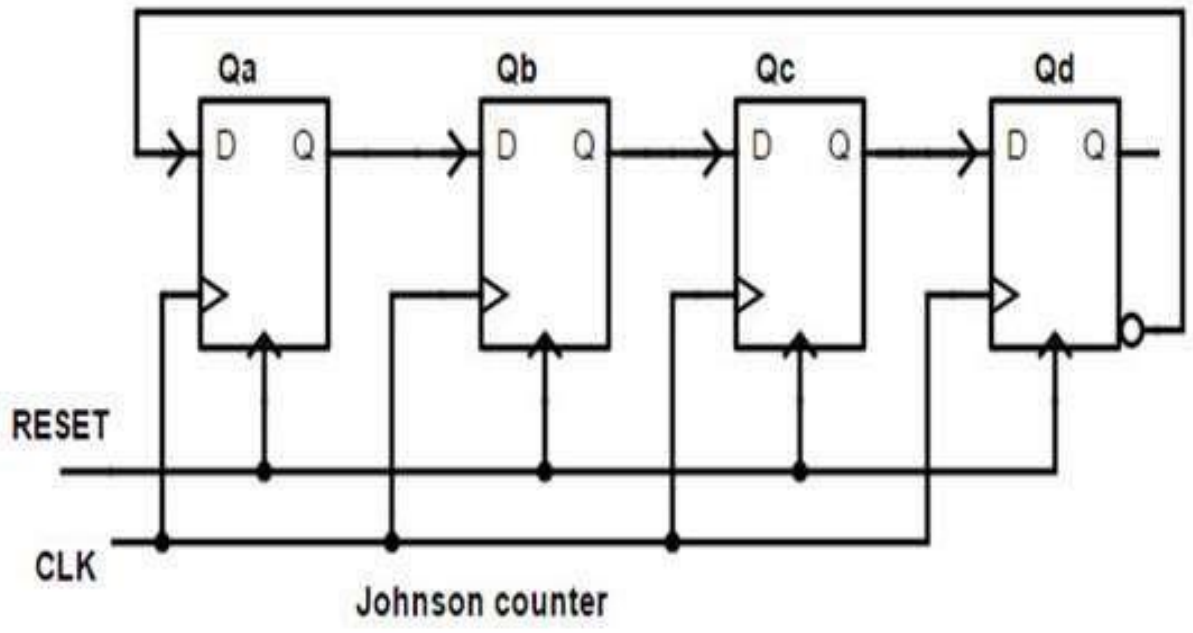
CIRCUIT DAIGRAM OF DECADE COUNTER



RING COUNTER

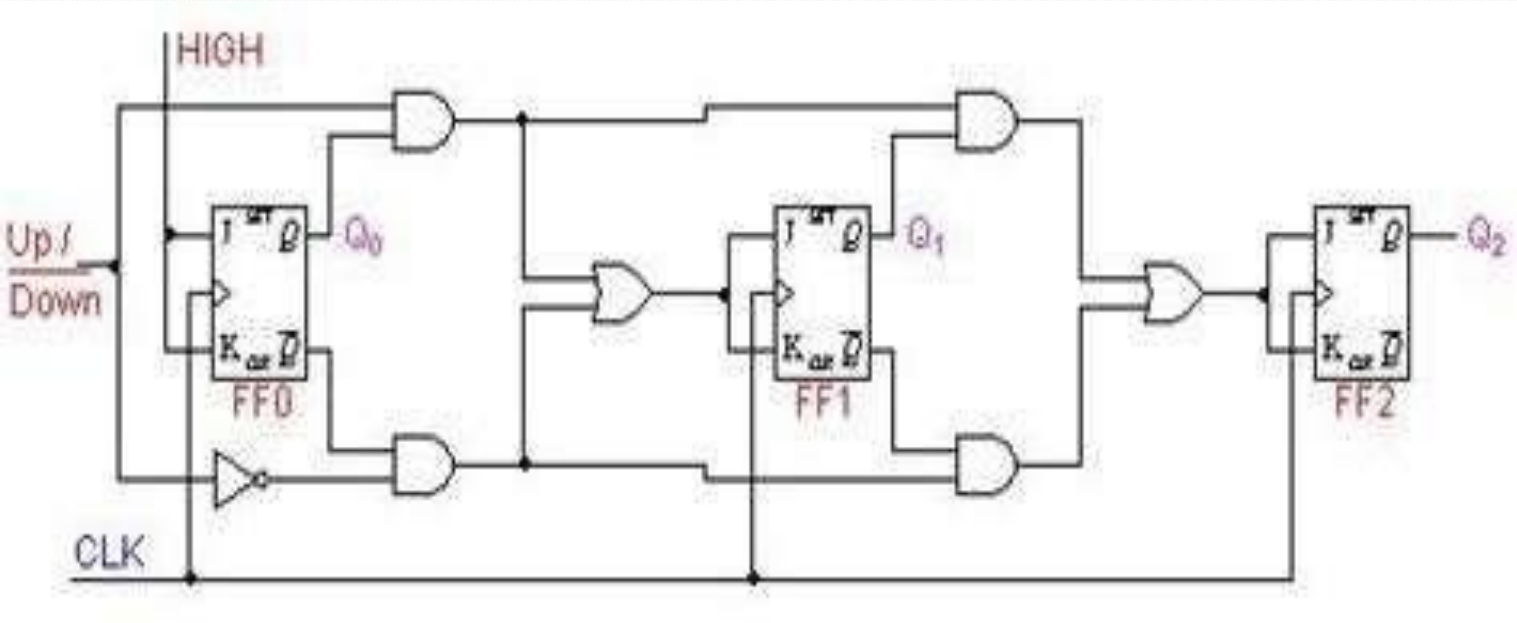
In a computer system, the standard ring counter form is a type of counter that is composed of a **shift register**, another type of counter that exists in the sequential logic form.

Q_A	Q_B	Q_C	Q_D
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
repeat			



UP/DOWN COUNTER

A circuit of a 3-bit synchronous up-**down counter** and a table of its sequence are shown below. Similar to an asynchronous up-**down counter**, a synchronous up-**down counter** also has an up-**down** control input. It is used to control the direction of the **counter** through a certain sequence



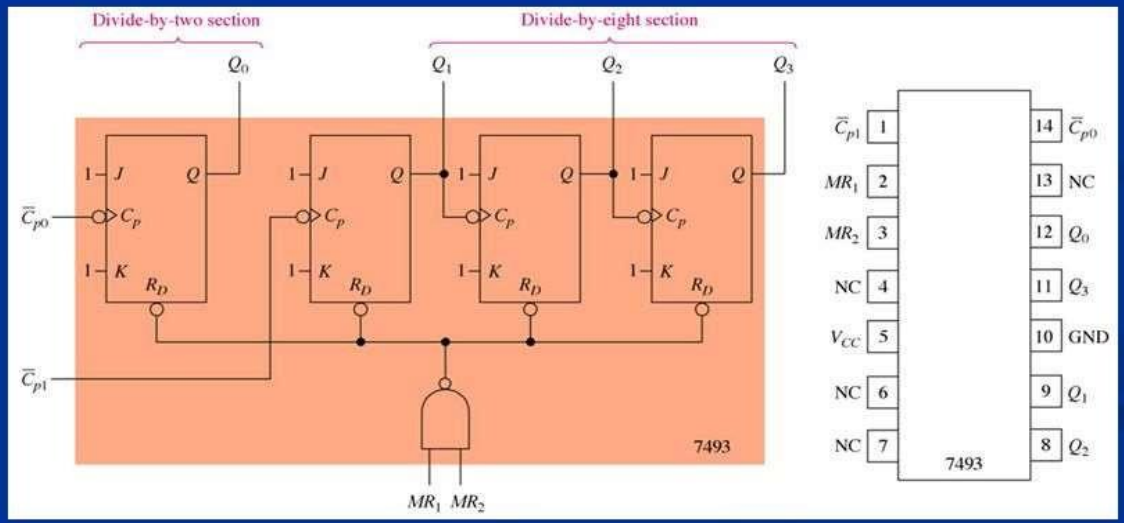
RIPPLE COUNTER

A **ripple counter** is an asynchronous **counter** where only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop. Asynchronous **counters** are also called **ripple-counters** because of the way the clock pulse ripples it way through the flip-flops.

CIRCUIT DAIGRAM OF RIPPLE COUNTER

Ripple Counter Integrated Circuits

- 7493 4-bit binary ripple counter logic diagram and pin configuration

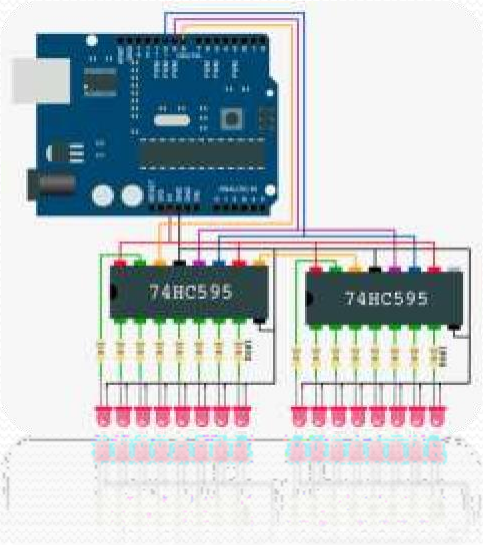


SHIFT REGISTER

CHAPTER-10

SHIFT REGISTER

- Introduction and basic concepts including shift left and shift right.
- (a) Serial in parallel out, serial in serial out, parallel in serial out, parallel in parallel out
- (b)Universal shift register
- (c)Buffer register, Tristate Register
- (d)IC 7495



DEFINATION

- A register is a digital circuit with two basic functions: Data Storage and Data Movement
  - A shift register provides the data movement function
  - A shift register "shifts" its output once every clock cycle

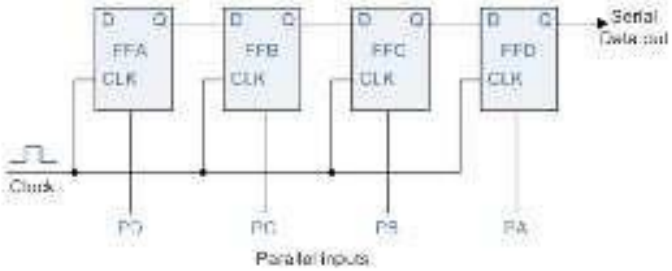
- A shift register is a group of flip-flops set up in a linear fashion with their inputs and outputs connected together in such a way that the data is shifted from one device to another when the circuit is active

TYPES OF SHIFT REGISTER

- SERIAL IN SERIAL OUT (SISO)
- SERIAL IN PARALLEL OUT (SIPO)
- PARALLEL IN SERIAL OUT (PISO)
- PARALLEL IN PARALLEL OUT (PIPO)

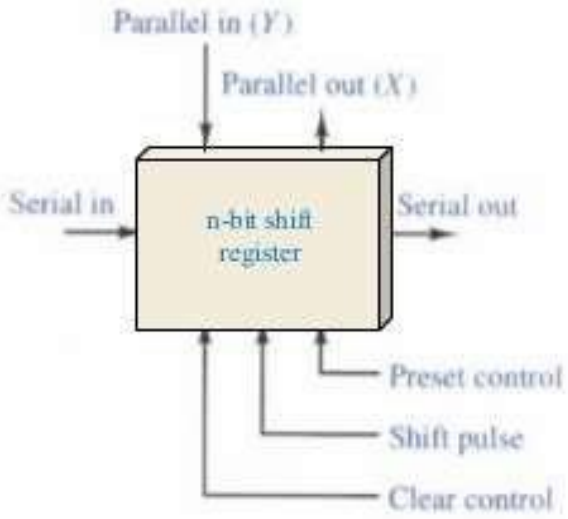
Shift Register Applications

- converting between serial data and parallel data
- temporary storage in a processor
  - scratch-pad memories
- some arithmetic operations
  - multiply, divide
- communications
  - UART
- some counter applications
  - ring counter
  - Johnson counter
  - Linear Feedback Shift Register (LFSR) counters
- time delay devices
- more …



Shift Register Characteristics

- Types
  - Serial-in, Serial-out
  - Serial-in, Parallel-out
  - Parallel-in, Serial-out
  - Parallel-in, Parallel-out
  - Universal
- Direction
  - Left shift
  - Right shift
  - Rotate (right or left)
  - Bidirectional

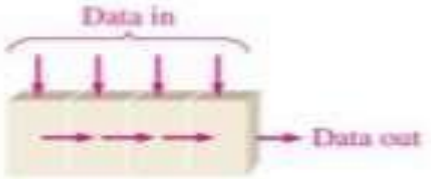


Data Movement

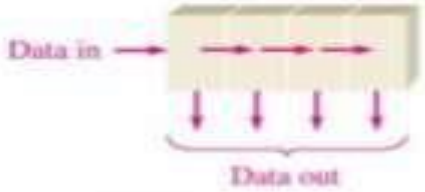
- The bits in a shift register can move in any of the following manners

Serial in/shift right/serial out

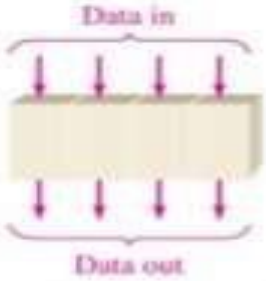
Serial in/shift left/serial out



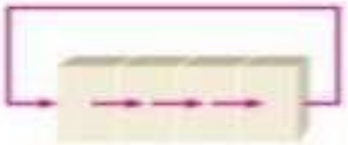
Parallel in/serial out



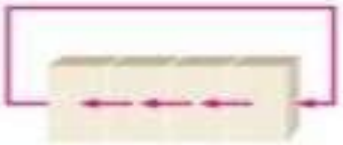
Serial in/parallel out



Parallel in/parallel out



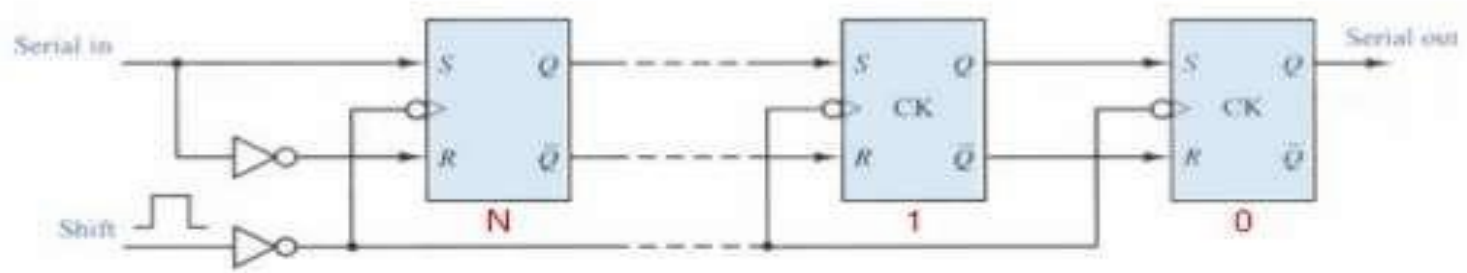
Rotate right



Rotate left

Serial-In Serial-Out

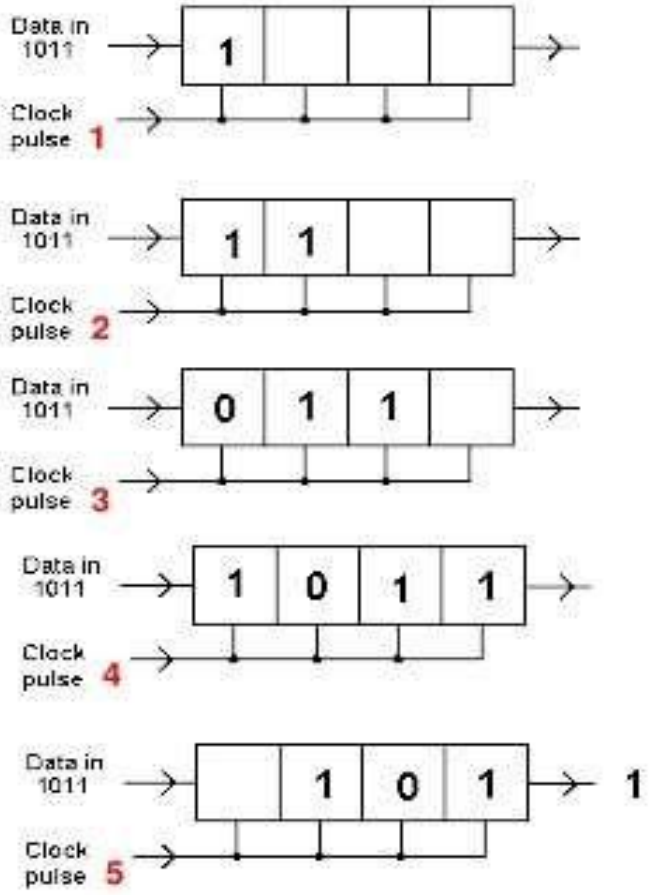
- The logic circuit diagram below shows a generalized serial-in serial-out shift register
  - SR Flip-Flops are shown
  - Connected to behave as D Flip-Flops
  - Input values moved to outputs of each Flip-Flop with the clock (shift) pulse



N-Bit Shift Register

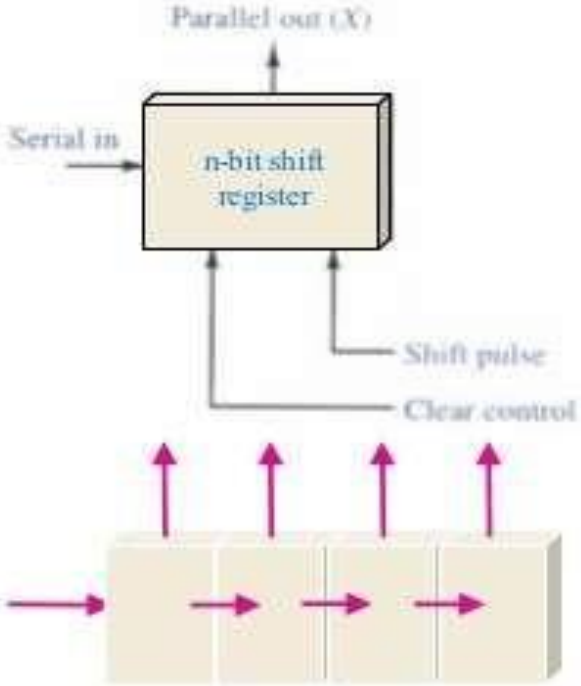
Serial-In Serial-Out

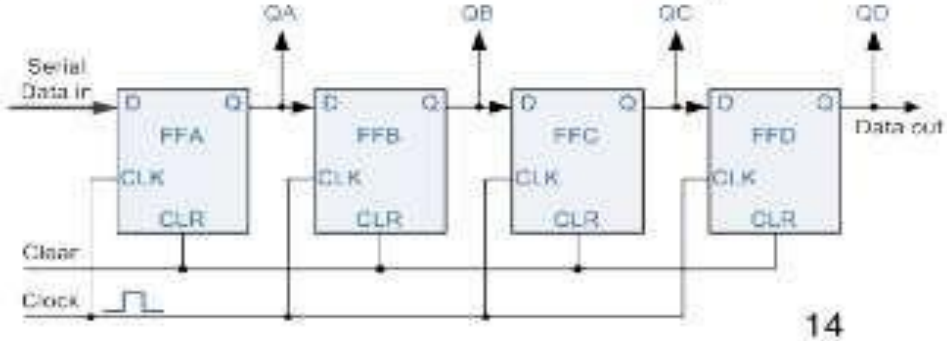
- A simple way of looking at the serial shifting operation, with a focus on the data bits, is illustrated at right
- The 4-bit data word "1011" is to be shifted into a 4-bit shift register
- One shift per clock pulse
- Data is shown entering at left and shifting right



Serial-to-Parallel Conversion

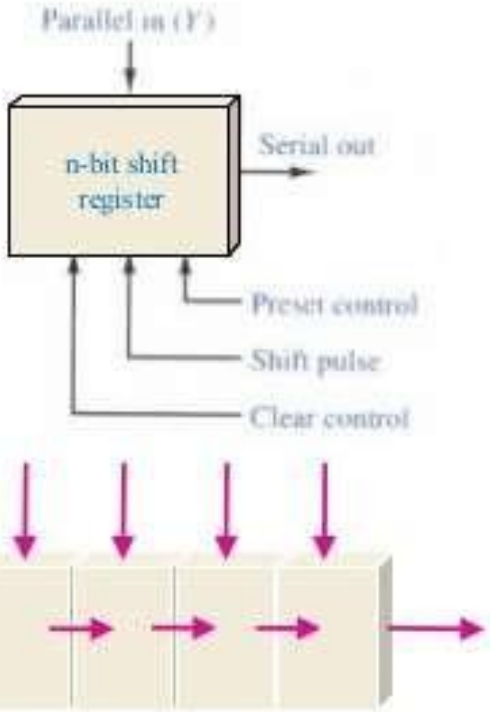
- We often need to convert from serial to parallel
  - e.g., after receiving a series transmission
- The diagrams at the right illustrate a 4-bit serial-in parallel-out shift register
- Note that we could also use the Q of the right-most Flip-Flop as a serial-out output

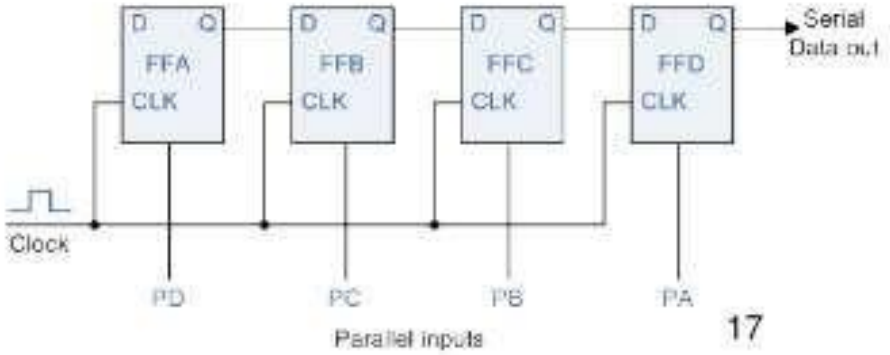




Parallel-to-Serial Conversion

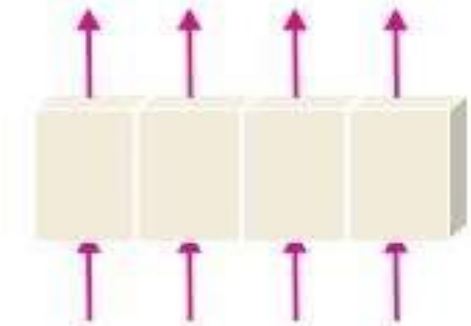
- We use a Parallel-in Serial-out Shift Register
- The DATA is applied in parallel form to the parallel input pins PA to PD of the register
- It is then read out sequentially from the register one bit at a time from PA to PD on each clock cycle in a serial format
- One clock pulse to load
- Four pulses to unload

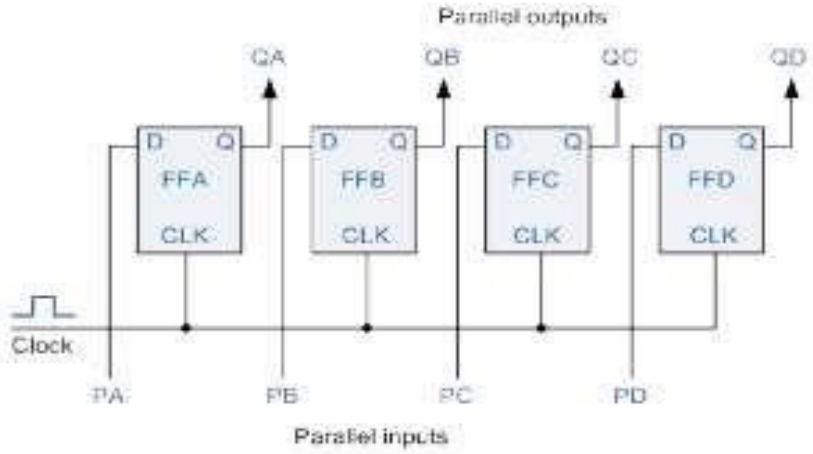




Parallel-In Parallel-Out

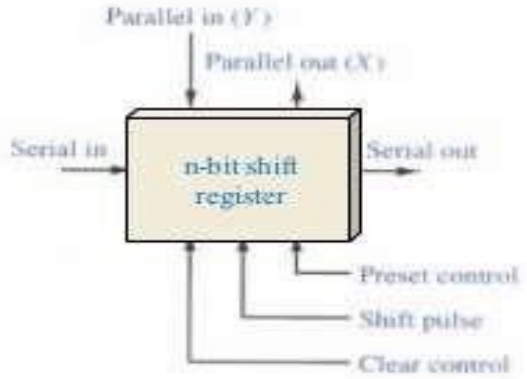
- Parallel-in Parallel-out Shift Registers can serve as a temporary storage device or as a time delay device
- The DATA is presented in a parallel format to the parallel input pins PA to PD and then shifted to the corresponding output pins QA to QD when the registers are clocked
- One clock pulse to load
- One pulse to unload

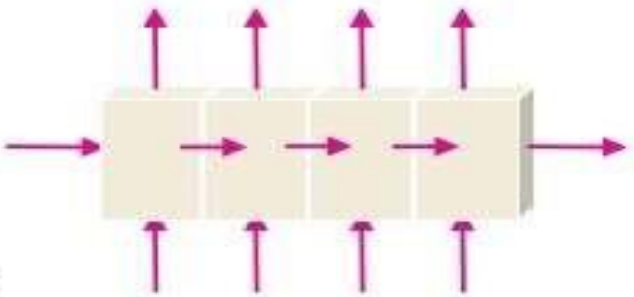


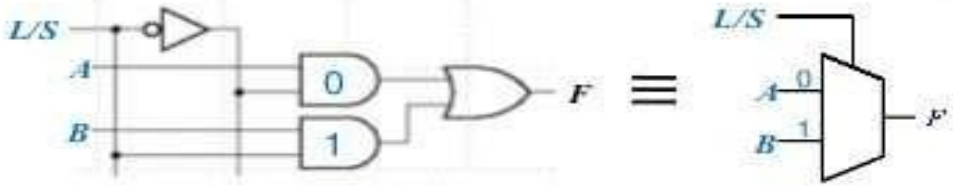


Universal Shift Register

- Universal shift register
- Can do any combination of parallel and serial input/output operations
- Requires additional inputs to specify desired function
- Uses a Mux-like input gating

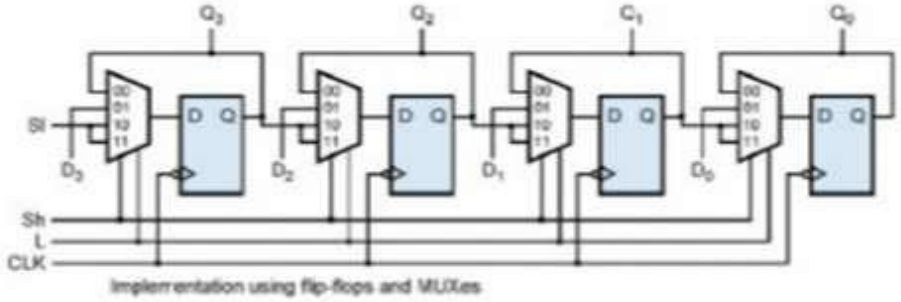




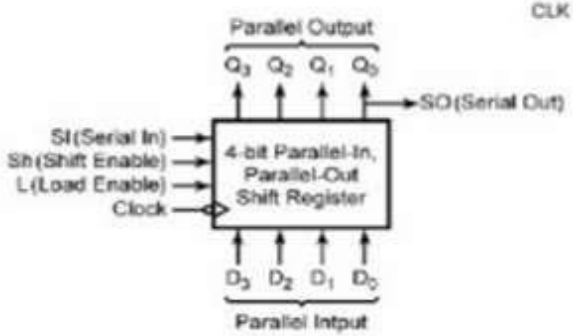


Universal Shift Register

- Parallel shift register (can serve as converting parallel-in to serial-out shifter):



Implementation using flip-flops and MUXes



Inputs		Next State				Action
Sh (Shift)	Ld (Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	
0	0	Q_3	Q_2	Q_1	Q_0	no change
0	1	D_3	D_2	D_1	D_0	load
1	X	SI	Q_3	Q_2	Q_1	right shift

BUFFER REGISTER

Buffer registers are a type of registers used to store a binary word. These can be constructed using a series of flip-flops as each flip-flop can store a single bit.

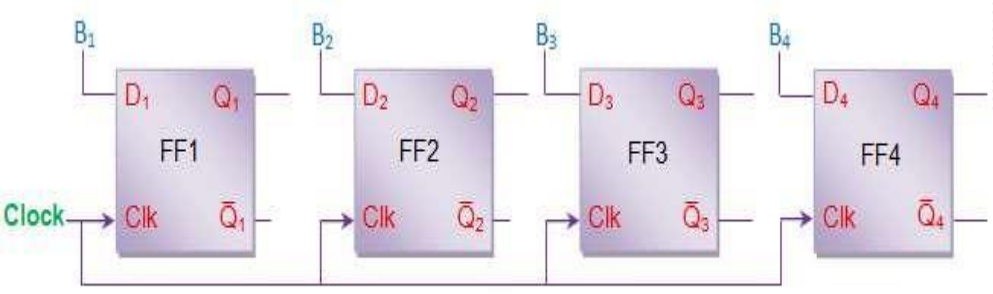


Figure 1 4-bit Buffer Register

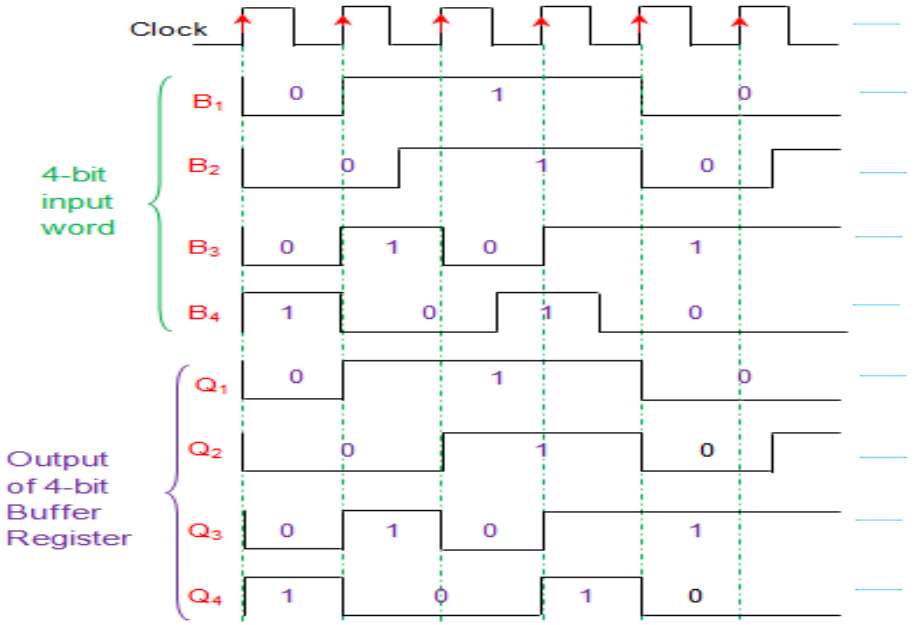


Figure 2 Input-Output Waveform for 4-bit

CONTROLLED BUFFEER REGISTER

Buffer registers offer no means of control over the inputs which in turn leads to uncontrolled outputs. In order to overcome this drawback one can resort to controlled buffer registers.

- Tri-state switches are used to control the operation of loading and/or retrieval of the data to/from the buffer register. Here one has to pull the LD or WR control line (blue line) low in order to store the data into the register, while RD control line (red line) should be made low to read the data.

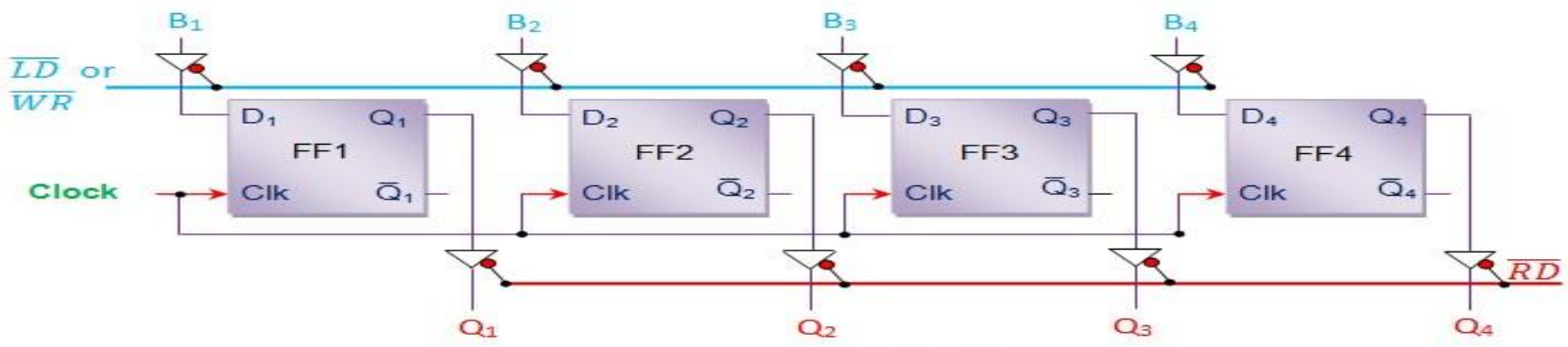


Figure 3 4-bit Controlled Buffer Register

BIDIRECTIONAL SHIFT REGISTER

Bidirectional shift registers are the storage devices which are capable of shifting the data either right or left depending on the mode selected.

it bidirectional shift register with serial data loading and retrieval capacity. Initially all the flip-flops in the register are reset by driving their clear pins high.Next R/LÌ... control line is made either low or high in order to opt for either left-shift or right-shift of the data bits, respectively.

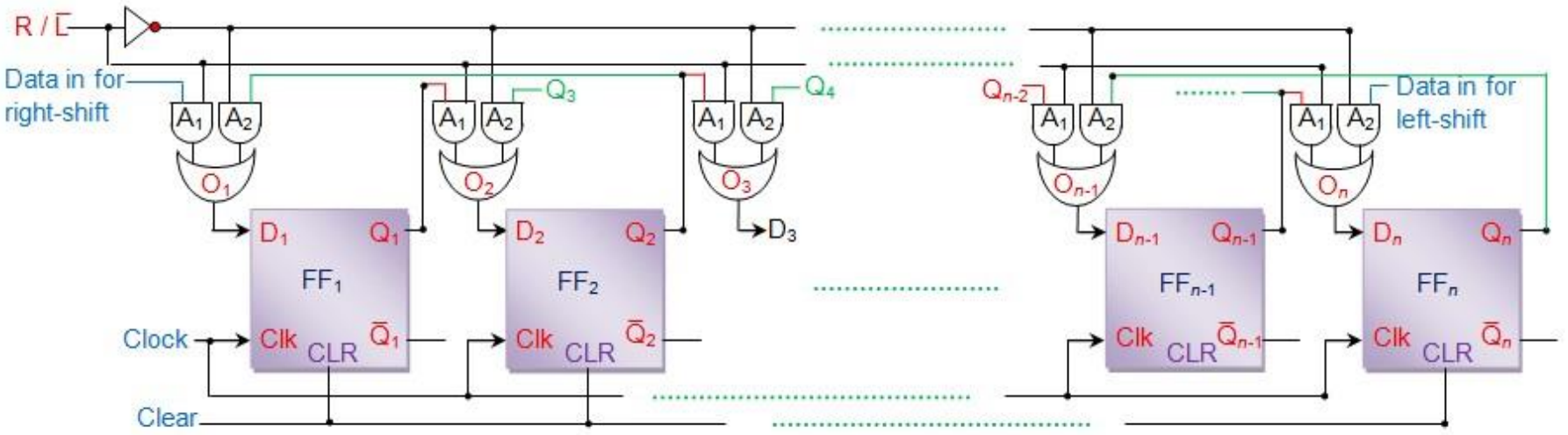
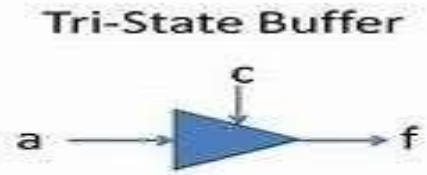


Figure 1 *n*-bit Bidirectional Shift Register

TRISTATE BUFFER

A tri-state buffer is similar to a buffer, but it adds an additional "enable" input that controls whether the primary input is passed to its output or not. If the "enable" inputs signal is true, the tri-state buffer behaves like a normal buffer. If the "enable" input signal is false, the tri-state buffer passes a *high impedance* (or hi-Z) signal, which effectively disconnects its output from the circuit



c	a	f
0	0	Z
0	1	Z
1	0	0
1	1	1

APPLICATIONS OF SHIFT REGISTER

- (A) TIME DELAY
- (B) SERIAL/PARALLEL DATA CONVERSION
- (C) RING COUNTERS
- (D)UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER

A/D AND D/A CONVERTERS

CHAPTER- **11**

WHAT IS A/D CONVERTER?

- An electronic integrated circuit which transforms a signal from analog (continuous) to digital (discrete) form.
- Analog signals are directly measurable quantities.
- Digital signals only have two states. For digital computer, we refer to binary states, 0 and 1.

WHY A/D CONVERTER IS NEEDED

- Microprocessors can only perform complex processing on digitized signals.
- When signals are in digital form they are less susceptible to the deleterious effects of additive noise.
- ADC Provides a link between the analog world of transducers and the digital world of signal processing and data handling.

APPLICATION OF A/D CONVERTERS

- ADC are used virtually everywhere where an analog signal has to be processed, stored, or transported in digital form.
- Some examples of ADC usage are digital volt meters, cell phone, thermocouples, and digital oscilloscope.

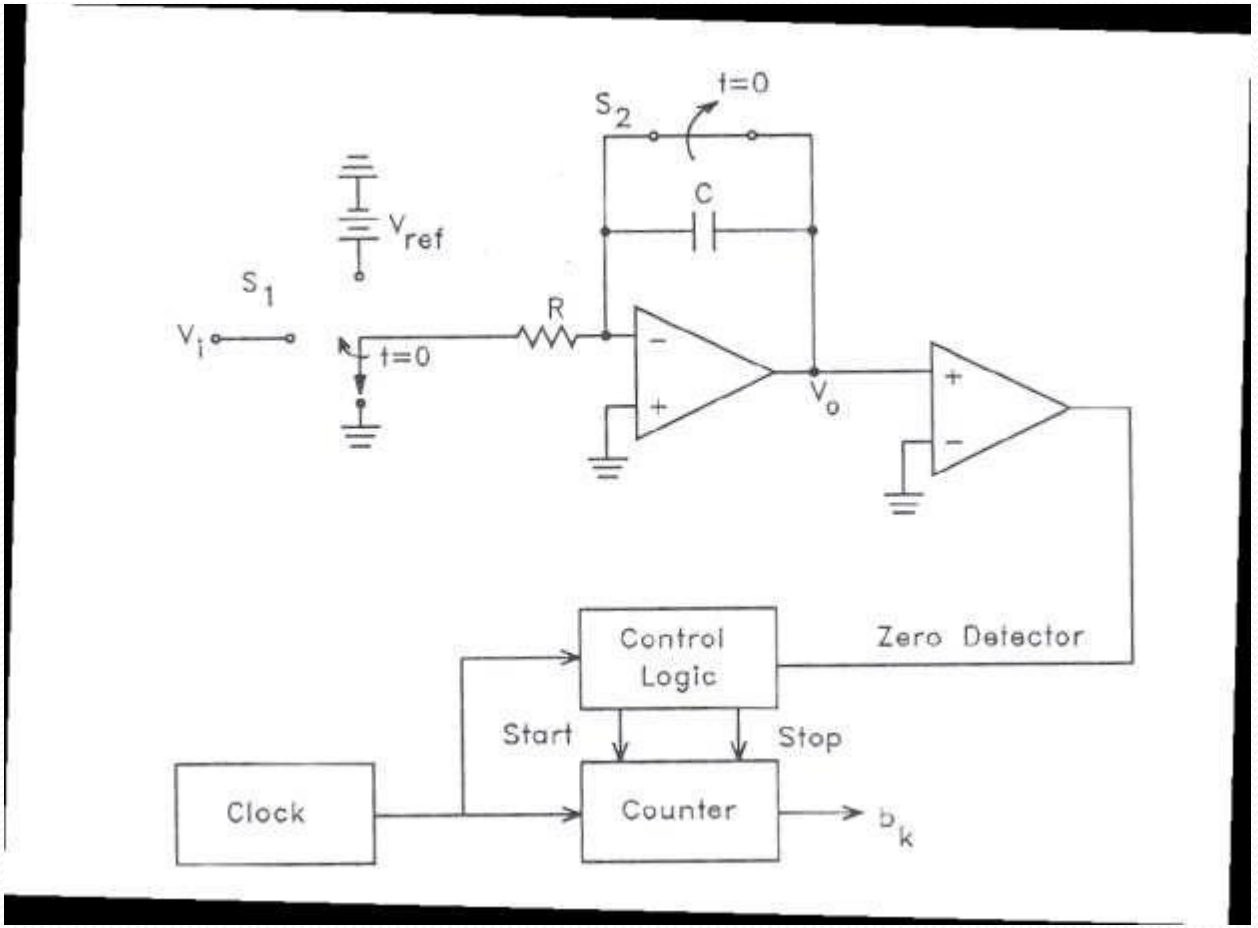
TYPES OF A/D CONVERTERS

- Dual Slope A/D Converter
- Successive Approximation A/D Converter
- Staircase ramp or Single Slope A/D Converter
- Parallel comparator A/D Converter

DUAL SLOPE A/D CONVERTER

- **Fundamental components**
- Integrator
- Electronically Controlled Switches
- Counter
- Clock
- Control Logic
- Comparator

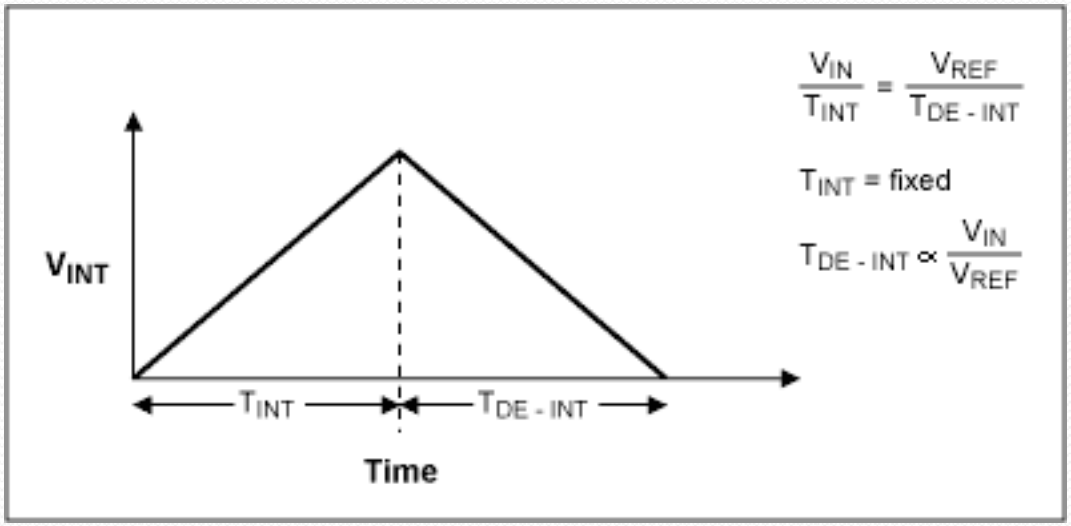
Block diagram of A/D Converter



How does it work

- A dual-slope A/D (DS-ADC) integrates an unknown input voltage (V_{IN}) for a fixed amount of time (T_{INT}), then "de-integrates" (T_{DEINT}) using a known reference voltage (V_{REF}) for a variable amount of time.
- The key advantage of this architecture over the single-slope is that the final conversion result is insensitive to errors in the component values. That is, any error introduced by a component value during the integrate cycle will be cancelled out during the de-integrate phase.

Waveform of DS-ADC



DS –ADC ADV. & DISADVANTAGES

- ADVANTAGES
  - Conversion result is insensitive to errors in the component values.
  - Fewer adverse affects from “noise”
  - High Accuracy
- DISADVANTAGES
  - Slow
  - Accuracy is dependent on the use of precision external components
  - Cost

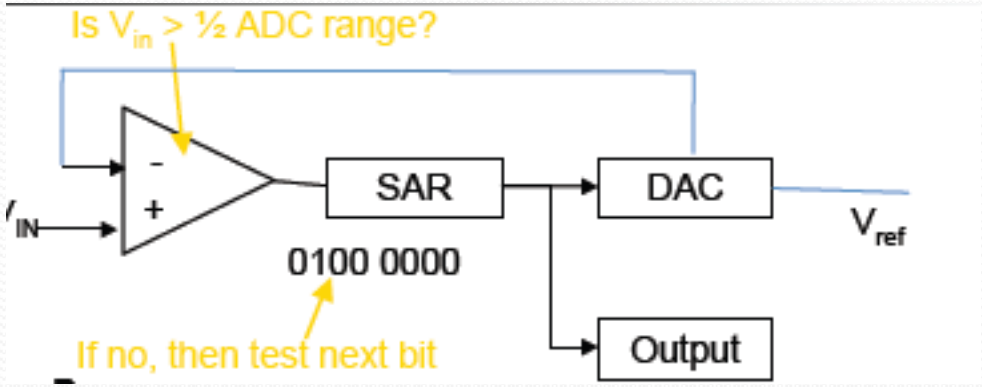
SUCCESSIVE APPROXIMATION A/D CONVERTER

- Uses a n-bit DAC to compare DAC and original analog results.
- Uses Successive Approximation Register (SAR) supplies an approximate digital code to DAC of Vin.
- Comparison changes digital output to bring it closer to the input value.
- Uses Closed-Loop Feedback Conversion.

PROCESS OF SA-ADC

- MSB initialized as 1
- Convert digital value to analog using DAC
- Compares guess to analog input
- Is $V_{in} > V_{DAC}$
    - Set bit 1
    - If no, bit is 0 and test next bit

BLOCK DIAGRAM OF SA-ADC



ADVANTAGES OF SA-ADC

Advantages

- Capable of high speed and reliable .
- Medium accuracy compared to other ADC types.
- Good tradeoff between speed and cost.
- Capable of outputting the binary number in serial.

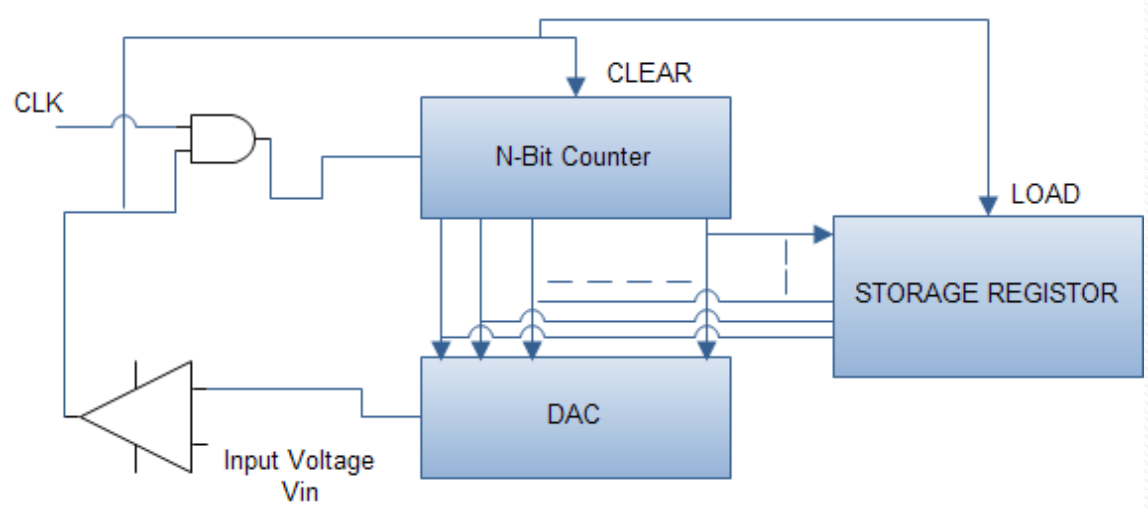
DISADVANTAGES OF SA-ADC

Disadvantages

- ADC's will be slower
- Speed limited to ~5Msps

COUNTER TYPE ADC

THE COUNTER TYPE ADC IS THE BASIC TYPE OF ADC WHICH IS ALSO CALLED AS DIGITAL RAMP TYPE ADC OR STAIR CASE APPROXIMATION ADC. THIS CIRCUIT CONSISTS OF N BIT COUNTER, DAC AND OP-AMP COMPARATOR AS SHOWN IN BELOW FIGURE.



ADVANTAGES & DISADVANTAGES OF COUNTER TYPE ADC

Advantages

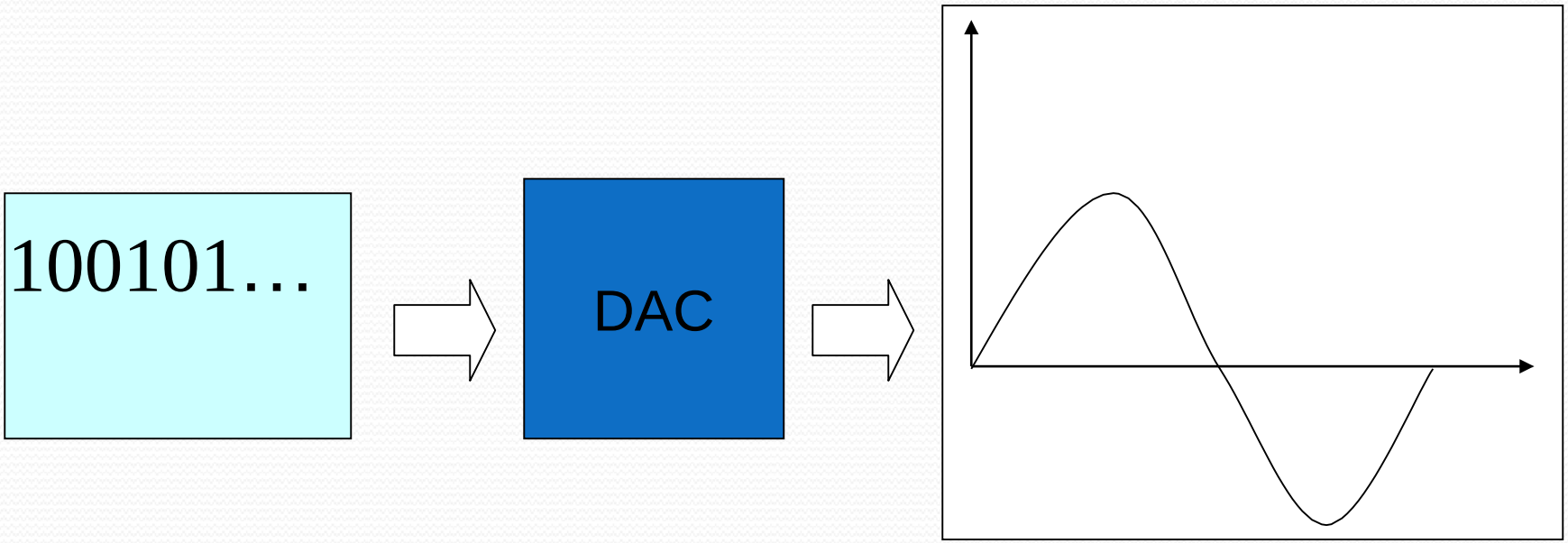
- Simple to understand and operate.
- Cost is less because of less complexity in design.

Disadvantages

- Speed is less because every time the counter has to start from ZERO.
- There may be clash or aliasing effect if the next input is sampled before completion of one operation ost is less because of less complexity in design.

What is a DAC?

- A digital to analog converter (DAC) converts a digital signal to an analog voltage or current output.



Types of DACs

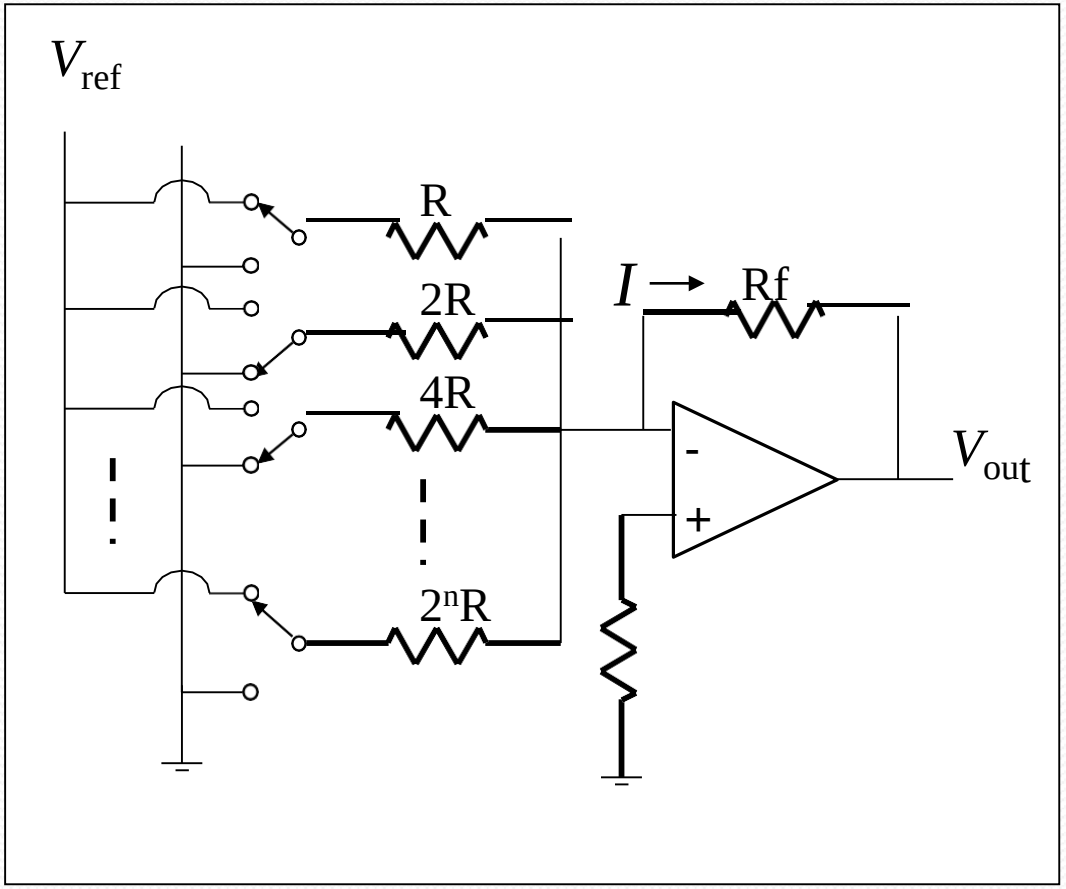
- Many types of DACs available.
- Usually switches, resistors, and op-amps used to implement conversion
- Two Types:
  - Binary Weighted Resistor
  - R-2R Ladder

Binary Weighted Resistor

- Utilizes a summing op-amp circuit
- Weighted resistors are used to distinguish each bit from the most significant to the least significant
- Transistors are used to switch between V_{ref} and ground (bit high or low)

Binary Weighted Resistor

- Assume IdealOp-amp
- No current int op-amp
- Virtual ground at inverting input
- $V_{out} = -IR_f$

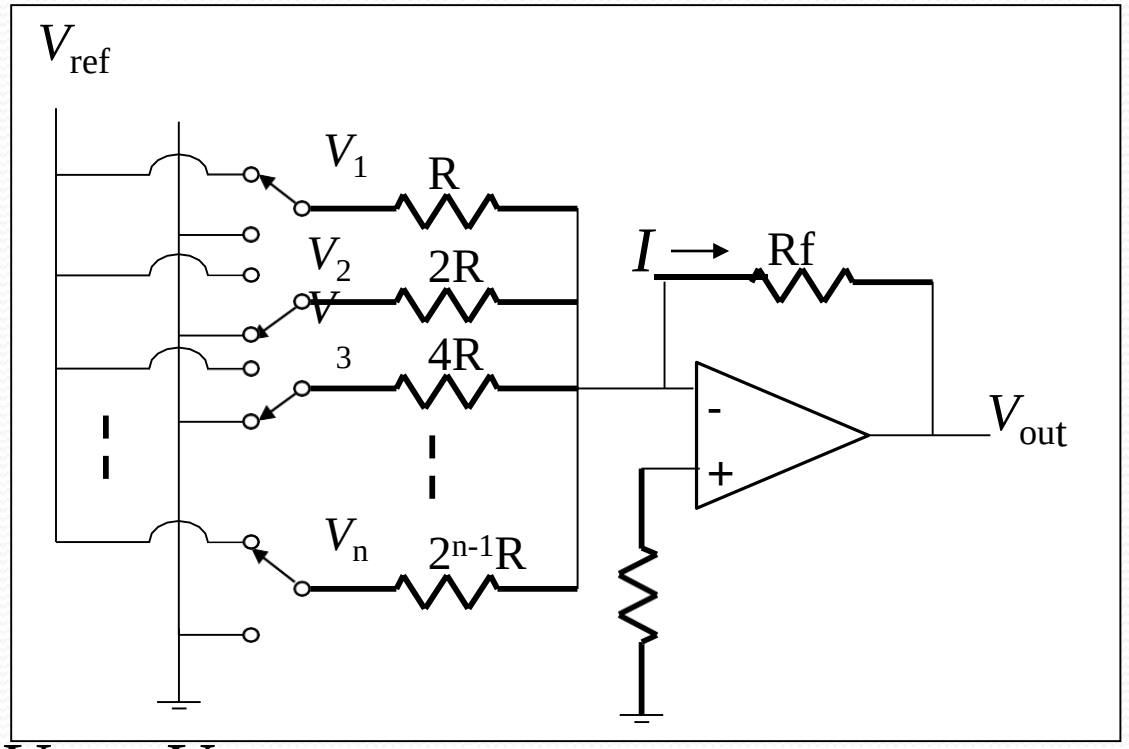


Binary Weighted Resistor

Voltages V_1 through V_n are either V_{ref} if corresponding bit is high or ground if corresponding bit is low

V_1 is most significant bit

V_n is least significant bit



MSB

LSB

$$V_{out} = -IR_f = -R_f\left(\frac{V_1}{R} + \frac{V_2}{2R} + \frac{V_3}{4R} + \cdots \frac{V_n}{2^{n-1}R}\right)$$

Binary Weighted Resistor

If $R_f = R/2$

$$V_{out} = -IR = -\left(\frac{V_1}{2} + \frac{V_2}{4} + \frac{V_3}{8} + \cdots \frac{V_n}{2^n}\right)$$

For example, a 4-Bit converter yields

$$V_{out} = -V_{ref}\left(b_3\frac{1}{2} + b_2\frac{1}{4} + b_1\frac{1}{8} + b_0\frac{1}{16}\right)$$

Where b_3 corresponds to Bit-3, b_2 to Bit-2, etc.

Binary Weighted Resistor

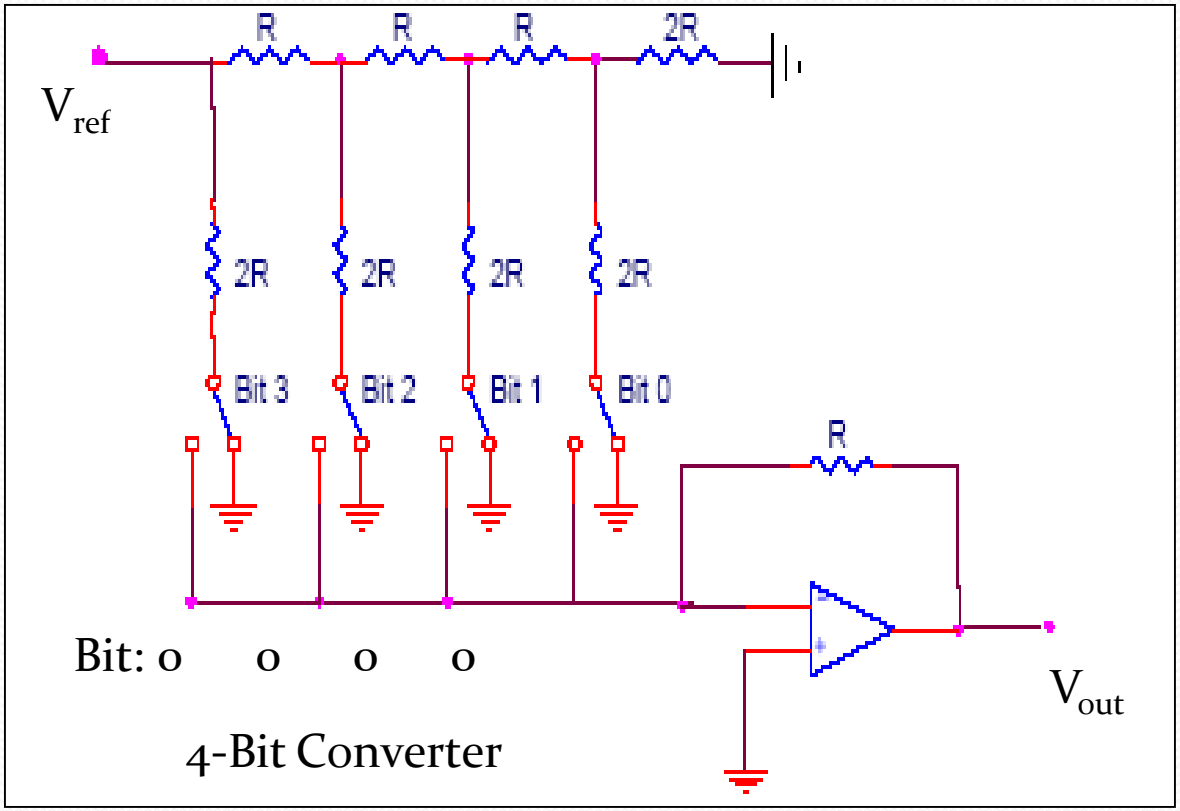
Advantages

- Simple Construction/Analysis
- Fast Conversion

Disadvantages

- Requires large range of resistors (2000:1 for 12-bit DAC) with necessary high precision for low resistors
- Requires low switch resistances in transistors
- Can be expensive. Therefore, usually limited to 8-bit resolution.

R-2R Ladder

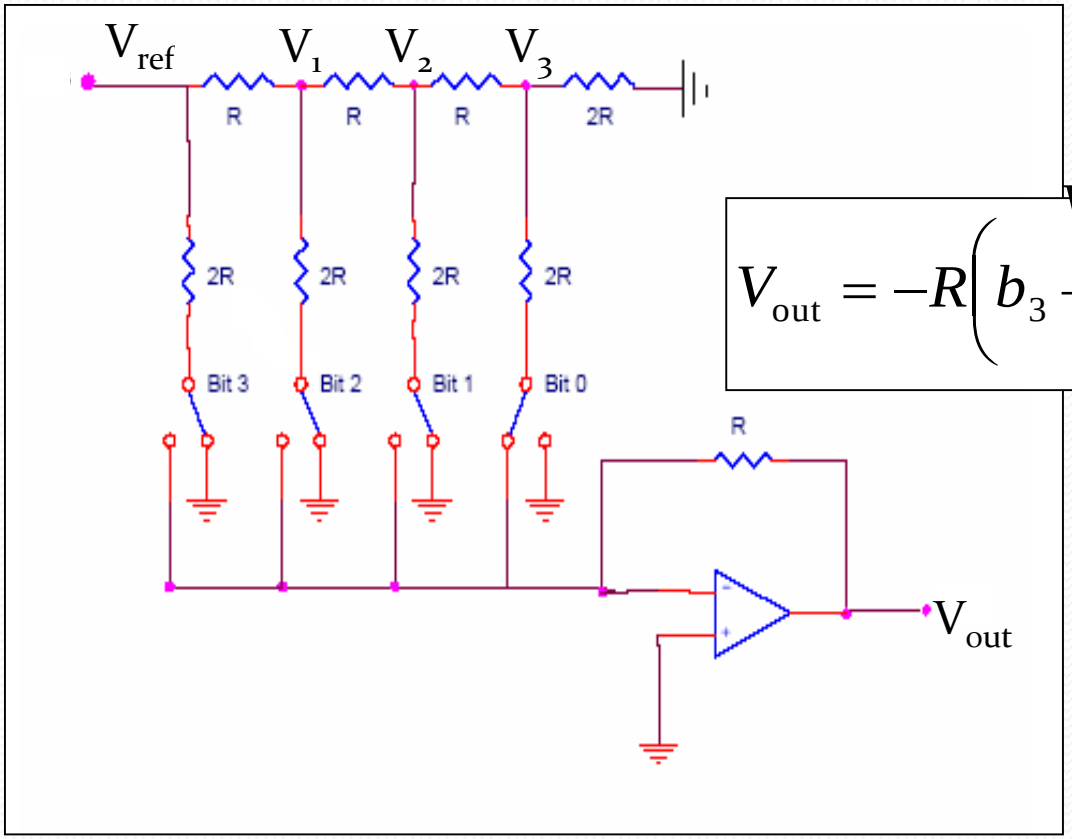


Each bit corresponds to a switch:

If the bit is high, the corresponding switch is connected to the inverting input of the op-amp.

If the bit is low, the corresponding switch is connected to ground.

R-2R Ladder



Results:

$$V_3 = \frac{1}{8} V_{ref}, \quad V_2 = \frac{1}{4} V_{ref}, \quad V_1 = \frac{1}{2} V_{ref}$$

$$V_{out} = -R\left(b_3 \frac{V_{ref}}{2R} + b_2 \frac{V_{ref}}{4R} + b_1 \frac{V_{ref}}{8R} + b_0 \frac{V_{ref}}{16R} \right)$$

Where b_3 corresponds to bit 3, b_2 to bit 2, etc.

If bit n is set, b_n=1

If bit n is clear, b_n=0

R-2R Ladder

For a 4-Bit R-2R Ladder

$$V_{out} = -V_{ref}\left(b_3\frac{1}{2} + b_2\frac{1}{4} + b_1\frac{1}{8} + b_0\frac{1}{16}\right)$$

For general n-Bit R-2R Ladder or Binary Weighted Resister DAC

$$V_{out} = -V_{ref}\sum_{i=1}^{n} b_{n-i}\frac{1}{2^i}$$

R-2R Ladder

Advantages

- Only two resistor values (R and 2R)
- Does not require high precision resistors

Disadvantage

- Lower conversion speed than binary weighted DAC

APPLICATIONS

- Digital Motor Control
- Computer Printers
- Sound Equipment (e.g. CD/MP3 Players, etc.)
- Electronic Cruise Control
- Digital Thermostat

SEMICONDUCTOR MEMORIES

CHAPTER-12

Memory Organization

- It provides spaces for storing instruction and data, space for intermediate results and spaces for final results.
- Memory is primarily of two types;
- (a) INTERNAL MEMORY: Primary/Main Memory And Cache Memory.
- (b) EXTERNAL MEMORY: Secondary Storage.

INTERNAL MEMORY

- PRIMARY

Primary memory is computer memory that a processor or computer accesses first or directly. It allows a processor to access running execution applications and services that are temporarily stored in a specific memory location. Primary memory is also known as primary storage or main memory

- CACHE MEMORY

Cache memory is a small-sized type of volatile computer **memory** that provides high-speed data access to a processor and stores frequently used computer programs, applications and data. It is the fastest **memory** in a computer

EXTERNAL MEMORY

- The storage capacity of the main memory or the primary memory of the computer is limited. Sometimes we have to store millions or billions bytes of data and primary memory of this computer is not able to store this data. Therefore, we require additional memory called auxiliary memory or secondary storage.

WHAT IS MEMORY?

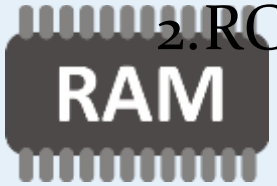
- The ability to store and retrive the digital information in a microcomputer system is called MEMORY.
- In past years, magnetic tape is used as memory element . But now days, with advancement in semi-conductor technology, semiconductor or memories of different types and sizes are used

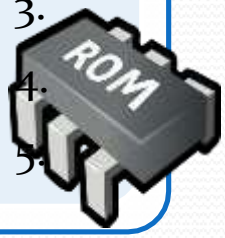
MEMORY

SEMICONDUCTOR MEMORY

MAGNETIC MEMORY

1. RAM
2. ROM

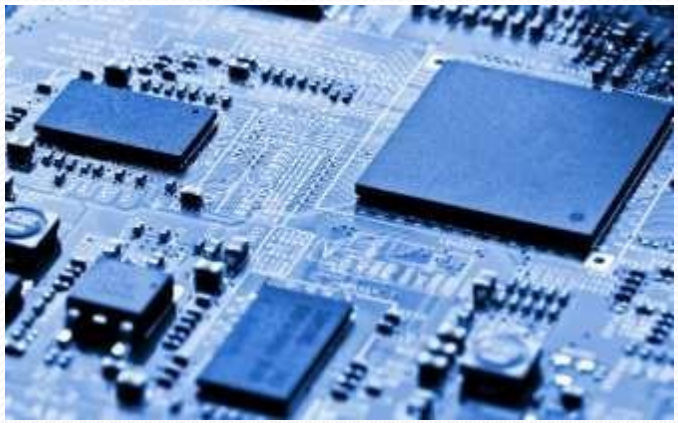




1. MAGNETIC TAPE
2. FLOPPY DISK
3. HARD DISK
4. OPTICAL DISC
5. MAGNETIC BUBBLE MEMORY

SEMICONDUCTOR MEMORY

semiconductor memories are small in size, have low cost, high speed of operation, high reliability and memory size can be expanded according to their requirements.



ROM

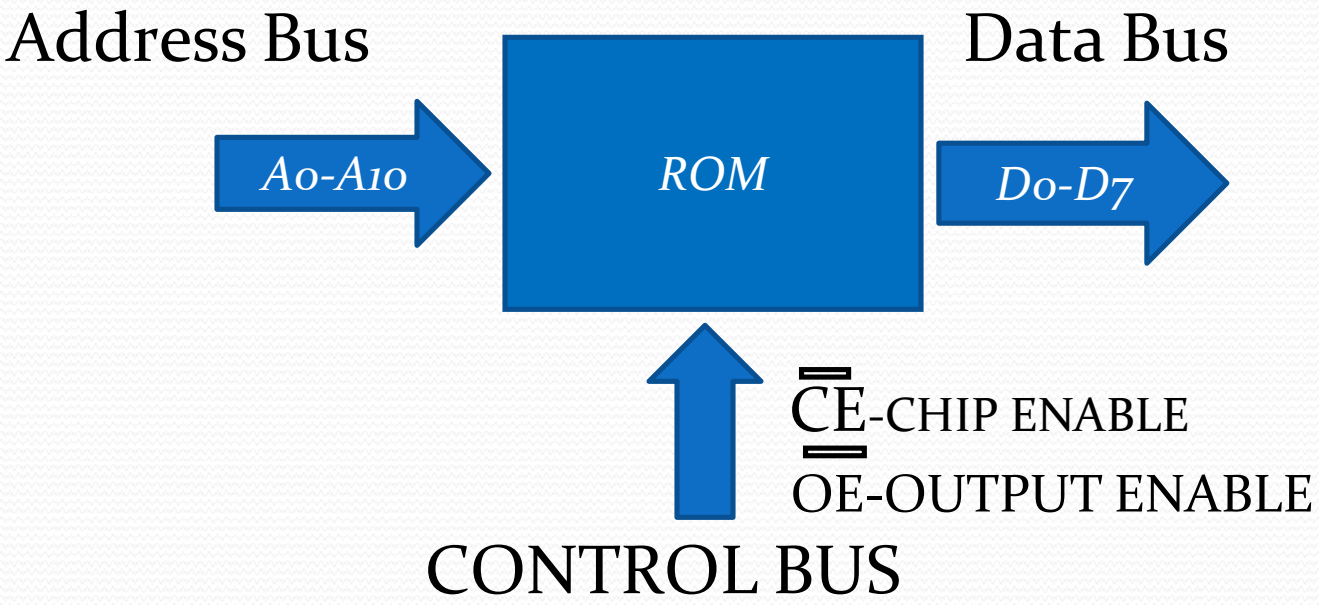
Bipolar Rom

MOS

WHAT IS ROM?

Read-only memory (**ROM**) is a type of storage medium that permanently stores data on personal computers (PCs) and other electronic devices. It contains the programming needed to start a PC, which is essential for boot-up; it performs major input/output tasks and holds programs or software instructions.

BLOCK DIAGRAM OF ROM



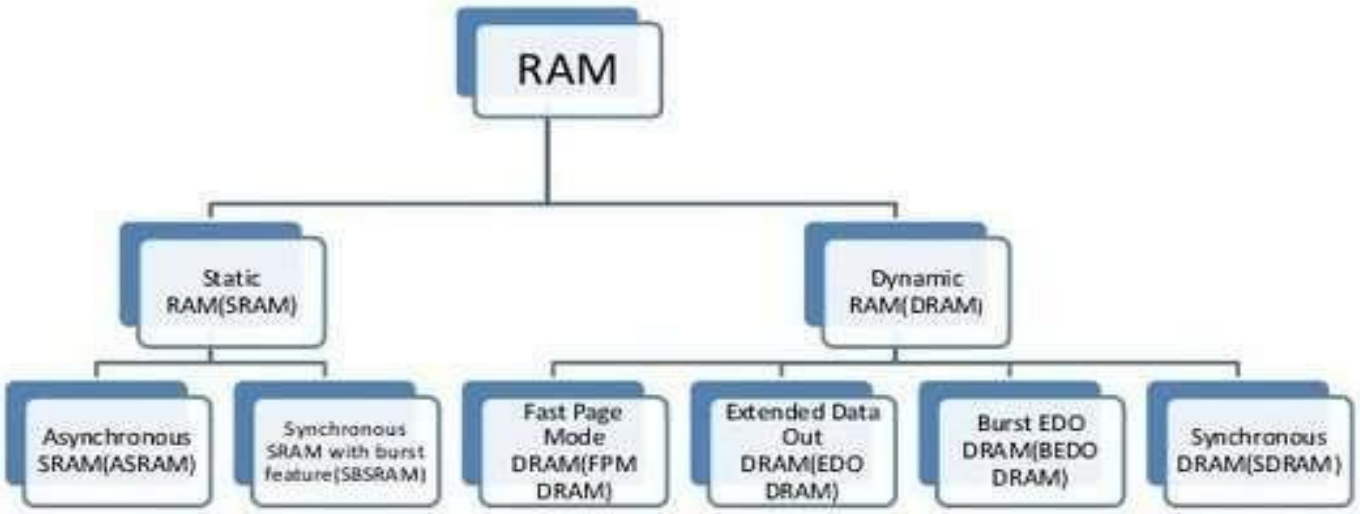
TYPES OF ROM:-

1. Programmable Read Only Memory[PROM]
2. Erasable Programmable Read Only Memory [EROM]
3. Electrical Erasable Programmable Memory [EEPROM]

WHAT IS RAM?

Random Access Memory (**RAM**) is the hardware in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. **RAM** is the main memory in a computer, and this type of memory is volatile and all information that was stored in **RAM** is lost when computer is turned off.

Types of Random-Access Memory(RAM)



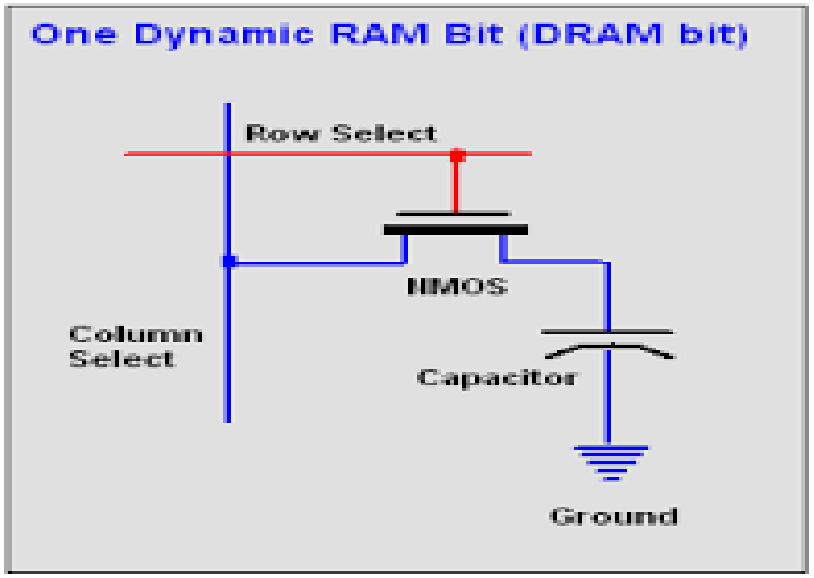
TYPES OF RAM

1.STATIC RAM

SRAM (**Static RAM**) is random access memory (**RAM**) that retains data bits in its memory as long as power is being supplied.

2.DYNAMIC RAM

Dynamic RAM (DRAM), which stores bits in cells consisting of a capacitor and a transistor



DIFFERENCE BETWEEN STATIC AND DYNAMIC RAM

Static RAM	Dynamic RAM
➢ SRAM uses transistor to store a single bit of data	➢ DRAM uses a separate capacitor to store each bit of data
➢ SRAM does not need periodic refreshment to maintain data	➢ DRAM needs periodic refreshment to maintain the charge in the capacitors for data
➢ SRAM's structure is complex than DRAM	➢ DRAM's structure is simplex than SRAM
➢ SRAM are expensive as compared to DRAM	➢ DRAM's are less expensive as compared to SRAM
➢ SRAM are faster than DRAM	➢ DRAM's are slower than SRAM
➢ SRAM are used in Cache memory	➢ DRAM are used in Main memory

"DIFFERENCE"

RAM	ROM
1. Temporary memory	1. Permanent memory
1. RAM enables data read & write to memory	1. Instructions written in ROM can only be read
1. Data can be changed or deleted	1. Data cannot be changed or deleted
1. Instructions are written into the RAM at the time of execution	1. Instructions are written into ROM at manufacturing time