SYNERGY SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING

LECTUER NOTES

Name of the Faculty: Prajnaparamita Kabi
Course: Diploma
Branch : CS
Semester: 3RD
Name of the subject: DIGITAL ELECTRONICS
Subject Code: Th-3

NUMBERSYSTEMANDCODES

INTRODUCTION:-

- Thetermdigitalreferstoaprocessthatisachievedbyusingdiscreteunit.
- Innumbersystemthereardifferentsymbolsandeachsymbolhasanabsolutevalueandalsohas place value.

RADIXORBASE:-

Theradixorbaseofanumbersystemisdefinedasthenumberofdifferentdigitswhichcanoccurin each position in the number system.

RADIXPOINT:-

The generalized form ofa decimal point is known as radix point. Inany positional numbersystemtheradix point divides the integer and fractional part.

N_r=[Integerpart▪Fractionalpart]

↑

Radixpoint

NUMBERSYSTEM:-

Ingeneralanumberinasystemhavingbaseorradix'r'canbewrittenas

$$a_n a_{n-1} a_{n-2} \ldots\ldots\ldots\ldots a_0 . a_{-1} a_{-2} \ldots\ldots\ldots\ldots a_{-m}$$

Thiswillbeinterpretedas

$$Y = a_n x r^n + a_{n-1} x r^{n-1} + a_{n-2} x r^{n-2} + \ldots\ldots + a_0 x r^0 + a_{-1} x r^{-1} + a_{-2} x r^{-2} + \ldots\ldots\ldots\ldots + a_{-m} x r^{-m}$$

where Y= value of theentire number

a_n=thevalueof thenthdigit r =

radix

TYPESOFNUMBERSYSTEM:-

Thereare fourtypes of numbersystems.Theyare

1. Decimalnumbersystem
2. Binarynumbersystem
3. Octalnumbersystem
4. Hexadecimalnumbersystem

DECIMALNUMBERSYSTEM:-

- Thedecimalnumbersystemcontaintenuniquesymbols0,1,2,3,4,5,6,7,8and9.
- Indecimalsystem10symbolsareinvolved,sothebaseorradixis10.
- Itisapositionalweightedsystem.
- Thevalueattachedtothesymboldependsonitslocationwithrespecttothedecimal point.

Ingeneral,

$d_n \; d_{n-1}d_{n-2}\ldots\ldots\ldots\ldots d_0.d_{-1}d_{-2}\ldots\ldots\ldots\ldots d_{-m}$

isgivenby

$(d_n x10^n)+(d_{n-1}x10^{n-1})+(d_{n-2}x10^{n-2})+ \ldots+(d_0x10^0)+(d_{-1}x10^{-1})+(d_{-2}x10^{-2})+\ldots+(d_{-m}x10^{-m})$

For example:-

9256.26=9x1000+ 2x100+5 x10+6x 1+ 2x(1/10)+ 6x(1/100)

$= 9x10^3+2x10^2+5x10^1+6x10^0+2x10^{-1}+6x10^{-2}$

BINARYNUMBERSYSTEM:-

- Thebinarynumbersystemisapositionalweighted system.
- The base orradixofthis numbersystem is 2.
- Ithastwoindependentsymbols.
- Thesymbols usedare0 and 1.
- Abinarydigitiscalledabit.
- Thebinarypointseparatestheintegerandfractionparts.

Ingeneral,

$d_n \; d_{n-1}d_{n-2}\ldots\ldots\ldots\ldots d_0.d_{-1}d_{-2}\ldots\ldots\ldots\ldots d_{-k}$

isgivenby

$(d_n x2^n)+(d_{n-1}x2^{n-1})+ (d_{n-2}x2^{n-2})+\ldots.+(d_0x2^0)+(d_{-1}x2^{-1})+(d_{-2}x2^{-2})+\ldots.+(d_{-k}x2^{-k})$

OCTALNUMBERSYSTEM:-

- Itisalsoapositionalweightedsystem.
- Itsbaseorradixis8.
- Ithas8independentsymbols0,1,2,3,4,5,6and7.
- Itsbase 8= 2^3, every3-bitgroup of binary canbe representedby anoctal digit.

HEXADECIMALNUMBERSYSTEM:-

- Thehexadecimalnumbersystemisapositionalweightedsystem.
- The base orradixofthis numbersystem is 16.
- Thesymbolsusedare0,1,2,3,4,5,6,7,8,9,A,B,C,D,EandF
- Thebase16=24,every4-bitgroupofbinarycanberepresentedby anhexadecimal digit.

CONVERSIONFROMONENUMBERSYSTEMTOANOTHER:-

1. **BINARYNUMBERSYSTEM:-**

(a) Binarytodecimalconversion:-

In this method, each binary digit of the number is multiplied by its positional weight and the product termsare added to obtain decimal number.

For example:

(i) Convert$(10101)_2$todecimal.

Solution :

(Positionalweight) $2^4\ 2^3\ 2^2 2^1 2^0$

Binarynumber 10101

$=(1x2^4)+(0x2^3)+(1x2^2)+(0x2^1)+(1x2^0)$

$=16+0+4+0+1$

$=(21)_{10}$

(ii) Convert$(111.101)_2$todecimal.

Solution:

$(111.101)_2$ $=(1x2^2)+(1x2^1)+(1x2^0)+(1x2^{-1})+(0x2^{-2})+(1x2^{-3})$

$=4+2+1+0.5+0+0.125$

$=(7.625)_{10}$

(b) BinarytoOctalconversion:-

Forconversionbinarytooctalthebinarynumbersaredividedintogroupsof3bitseach,startingatthe binary point and proceeding towards left and right.

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

For example:

(i) Convert$(101111010110.110110011)_2$intooctal.

Solution :

Groupof3bitsare	101	111	010	110.		110	110	011
Convert eachgroupintooctal=	5	7	2	6	.	6	6	3

Theresultis**$(5726.663)_8$**

(ii) Convert$(10101111001.0111)_2$intooctal.

Solution :

Binarynumber		10	101	111	001.		011	1
Groupof3bitsare	=010		101	111	001.		011	100
Converteachgroupintooctal=		2	5	7	1	.	3	4

Theresultis**$(2571.34)_8$**

(c) BinarytoHexadecimalconversion:-

For conversion binary to hexadecimal number the binary numbers starting from the binarypoint, groups are made of 4 bits each, on either side of the binary point.

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

For example:
(i) Convert$(1011011011)_2$intohexadecimal.

Solution:

GivenBinarynumber 10 1101 1011

Groupof4bitsare 0010 1101 1011

Converteachgroupintohex = 2 D B

The result is**$(2DB)_{16}$**

(ii) Convert$(01011111011.011111)_2$intohexadecimal.

Solution:

GivenBinarynumber 010 1111 1011 . 0111 11

Groupof3bitsare =0010 1111 1011 . 0111 1100

Convert eachgroupintooctal= 2 F B . 7 C

The result is**$(2FB.7C)_{16}$**

2. DECIMALNUMBERSYSTEM:-

(a) Decimaltobinaryconversion:-

In the conversion the integer number are converted to the desired base using successive division by the base or radix.

For example:
(i) Convert$(52)_{10}$intobinary.

Solution:

Divide the given decimal number successively by 2 read the integer part remainder upwards to get equivalent binarynumber. Multiply thefractionpart by 2. Keepthe integer in the product as it is andmultiply the newfraction in the product by 2. The process is continued and the integer are read in the products from top to bottom.

2|52
2|26 —0
2|13 —0
2|6 —1
2|3 —0
2|1 —1
0 —1

Resultof$(52)_{10}$is**$(110100)_2$**

(ii) Convert$(105.15)_{10}$intobinary.

Solution:

Integerpart		**Fraction part**
2|105		0.15x2=0.30
2|52	1	0.30x2=0.60
2|26	0	0.60x2=1.20
2|13	0	0.20x2=0.40
2 | 6	1	0.40x2=0.80
2 | 3	0	0.80x2=1.60
2 | 1	1	
0	1	

Resultof$(105.15)_{10}$is**$(1101001.001001)_2$**

(b) Decimaltooctalconversion:-

To convert the given decimal integer number to octal, successively divide the given number by 8 till the quotient is 0. To convert the given decimal fractions to octal successively multiply the decimal fraction and the subsequent decimal fractions by 8 till the product is 0 or till the required accuracy is obtained.

For example:

(i) Convert$(378.93)_{10}$intooctal.

Solution:

8|378		0.93x8=7.44
8|47	2	0.44x8=3.52
8|5	7	0.52x8=4.16
0	5	0.16x8=1.28

Resultof$(378.93)_{10}$is**$(572.7341)_8$**

(c) Decimaltohexadecimalconversion:-

Thedecimaltohexadecimalconversionissameasoctal.

For example:

(i) Convert$(2598.675)_{10}$intohexadecimal.

Solution:

	Remainder			
	Decimal	**Hex**		**Hex**
16|2598			0.675 x16=10.8	A
16|162	6	6	0.800 x16=12.8	C
16|10	2	2	0.800 x16=12.8	C
0	10	A	0.800 x16=12.8	C

Resultof$(2598.675)_{10}$is**$(A26.ACCC)_{16}$**

3. OCTALNUMBERSYSTEM:-

(a) Octaltobinaryconversion:-

Toconvertagivenaoctalnumbertobinary,replaceeachoctaldigitbyits3-bitbinary equivalent.

For example:

Convert$(367.52)_8$intobinary.

Solution:

GivenOctalnumber is		3	6	7	.	5	2
Converteachgroupoctal to binary	=011	110111	.	101010			

Resultof$(367.52)_8$ is**$(011110111.101010)_2$**

(b) Octaltodecimal conversion:-

For conversion octal to decimal number, multiply each digit in the octal number by the weight of its position and add all the product terms

Forexample:-

Convert$(4057.06)_8$todecimal

Solution:

$(4057.06)_8 = 4x8^3+0x8^2+5x8^1+7x8^0+0x8^{-1}+6x8^{-2}$

$= 2048+0+40+7+0+0.0937$

$= (2095.0937)_{10}$

Resultis**$(2095.0937)_{10}$**

(c) Octaltohexadecimal conversion:-

ForconversionofoctaltoHexadecimal,firstconvertthegivenoctalnumbertobinaryandthenbinary number to hexadecimal.

Forexample:-

Convert$(756.603)_8$tohexadecimal.

Solution :-

Givenoctalno.		7	5	6	.	6	0	3
Converteachoctaldigittobinary	=	111	101	110	.	110	000	011
Groupof4bits are	=	0001	1110	1110	.	1100	0001	1000
Convert4 bitsgroupto hex.	=	1	E	E	.	C	1	8

Result is **$(1EE.C18)_{16}$**

(4)HEXADECIMALNUMBERSYSTEM:-

(a) Hexadecimaltobinary conversion:-

Forconversionofhexadecimaltobinary,replacehexadecimaldigit byits4bitbinarygroup. **For example:**

Convert$(3A9E.B0D)_{16}$intobinary.

Solution:

GivenHexadecimalnumberis		3	A	9	E	.	B	0	D
Convert each hexadecimal bitbinary	=0011101010011110.101100001101digit to 4								

Resultof$(3A9E.B0D)_8$is**$(0011101010011110.101100001101)_2$**

(b) Hexadecimaltodecimalconversion:-

Forconversionofhexadecimaltodecimal,multiplyeachdigitinthehexadecimalnumberbyitsposition weight and add all those product terms.

Forexample:-
Convert$(A0F9.0EB)_{16}$todecimal

Solution:

$(A0F9.0EB)_{16} = (10x16^3)+(0x16^2)+(15x16^1)+(9x16^0)+(0x16^{-1})+(14x16^{-2})+(11x16^{-3})$
$= 40960+0+240+9+0+0.0546+0.0026$
$= (41209.0572)_{10}$

Resultis**$(41209.0572)_{10}$**

(c) Hexadecimal toOctal conversion:-

Forconversionofhexadecimaltooctal,firstconvertthegivenhexadecimalnumbertobinaryandthen binary number to octal.

Forexample:-
Convert$(B9F.AE)_{16}$tooctal.

Solution:-

Givenhexadecimalno.is		B	9	F	.	A	E			
Converteachhex.digittobinary	=	1011	1001	1111	.	1010	1110			
Groupof3bitsare	=	101	110	011	111	.	101	011	100	
Convert3 bitsgrouptooctal.	=	5	6	3	7	.	5	3	4	

Resultis(**$5637.534)_8$**

BINARYARITHEMATICOPERATION:-

1. BINARYADDITION:-

The binaryaddition rulesare asfollows
0 + 0 = 0; 0 + 1 = 1; 1 + 0 = 1; 1 + 1 =10 , i.e 0with a carry of 1 **For**

example :-

Add$(100101)_2$and$(1101111)_2$.
Solution :-

```
      10010 1
+   1 1 0 1 1 1 1
    10010 100
```

Resultis**$(10010100)_2$**

2. BINARYSUBTRACTION:-

Thebinarysubtractionrulesareasfollows
0-0=0;1 -1=0;1-0=1;0-1 =1,withaborrowof1

Forexample:-
Substract$(111.111)_2$from$(1010.01)_2$.
Solution :-

$$\begin{array}{rr} & 1010\,.\,0\,10 \\ - & \underline{\;\;111.1\,11} \\ & \underline{0010.011} \end{array}$$

Resultis**$(0010.011)_2$**

3. BINARYMULTIPLICATION:-

Thebinarymultiplicationrulesareasfollows 0 x 0 = 0 ; 1 x 1 = 1 ; 1 x0 = 0 ; 0x 1 = 0

Forexample:-

Multiply$(1101)_2$by$(110)_2$.
Solution :-

$$\begin{array}{rr} & 1101 \\ \times & \underline{\;\;110} \\ & 0000 \\ & 1101 \\ + & \underline{1101\;\;\;\;} \\ & \underline{1001110\;\;} \end{array}$$

Resultis**$(1001110)_2$**

4. BINARYDIVISION:-

The binary division is very simple and similar to decimal number system. The division by '0' is meaningless. So we have only 2 rules

0 ÷ 1 = 0

1 ÷ 1 = 1

Forexample:-
Divide$(10110)_2$by$(110)_2$. Solution

:-

$$\begin{array}{l} 110)\ 101101\ (\ 111.1 \\ \quad - \ \underline{110\;\;\;} \\ \qquad\ 1010 \\ \qquad\ \underline{110\;\;\;} \\ \qquad\quad 1001 \\ \qquad\ \underline{110\;\;\;} \\ \qquad\quad\ 110 \\ \qquad\ \underline{110\;\;\;\;} \\ \qquad\ \underline{000\;\;\;\;} \end{array}$$

Resultis**$(111.1)_2$**

1'sCOMPLEMENTREPRESENTATION:-

The1'scomplementofabinarynumberisobtainedbychangingeach0to1andeach1to 0.

Forexample:-

Find$(1100)_2$1'scomplement.

Solution :-

Given	1	1	0	0
1'scomplementis	0	0	1	1

Resultis**$(0011)_2$**

2'sCOMPLEMENTREPRESENTATION:-

The2'scomplementofabinarynumberisabinarynumberwhichisobtainedbyadding1tothe1's complement of a number i.e.

2'scomplement=1'scomplement+1

Forexample:-

Find$(1010)_2$2'scomplement.

Solution :-

Given	1	0	1	0
1'scomplementis	0	1	0	1
+				1
2'scomplement	0	1	1	0

Resultis**$(0110)_2$**

SIGNEDNUMBER:-

In sign –magnitude form, additional bit called the sign bit is placed in front of the number. If the sign bit is 0, the number is positive. If it is a 1, the number is negative.

For example:-

0 1 0 1 0 0 1= +41
↑
Signbit

1 1 0 1 0 0 1= -41
↑
Signbit

SUBSTRACTIONUSINGCOMPLEMENTMETHOD:-

1'sCOMPLEMENT:-

In 1's complement subtraction, add the 1's complement of subtrahend to the minuend. If there is a carry out, then the carry is added to the LSB. This is called end around carry. If the MSB is 0, the result is positive. If the MSB is 1, the result is negative and is in its 1's complement form. Then take its 1's complement to get the magnitude in binary.

For example:-

Subtract$(10000)_2$from$(11010)_2$using1'scomplement.

Solution:-

11010		11010		= 26
-10000	=>	+0 1111	(1'scomplement)	= - 16
	Carry	→ **1**01001		+ 10
		+ 1		
		01010	=+10	

Resultis**+10**

2'sCOMPLEMENT:-

In 2's complement subtraction, add the 2's complement of subtrahend to the minuend. If there is a carry out, ignore it. If the MSB is0, the result is positive. If the MSB is 1, the result is negative and is in its 2'scomplement form. Then take its 2's complement to get the magnitude in binary.

For example:-

Subtract$(1010100)_2$from$(1010100)_2$using2'scomplement.

Solution:-

1010100		1010100		= 84
- 1010100	=>	+0 101100(2'scomplement)		= -84
		10000000(Ignorethecarry)		0
		= 0(result =0)		

HenceMSBis0.Theanswerispositive.Soitis+0000000= **0**

DIGITAL CODES:-

In practice the digital electronics requires to handle data which may be numeric, alphabets and special characters. This requires the conversion of the incoming data into binary format before it can be processed. There is various possible ways of doing this and this process is called encoding. To achieve the reverse of it,we use decoders.

WEIGHTEDANDNON-WEIGHTEDCODES:-

Therearetwotypesofbinary codes

1) Weightedbinarycodes
2) Non-weightedbinarycodes

Inweightedcodes,foreachposition(orbit),thereisspecificweight attached.

Forexample,inbinarynumber,eachbitisassignedparticularweight2nwhere'n'isthebitnumberforn= 0,1,2,3,4 the weights are 1,2,4,8,16 respectively.

Example:- BCD

Non-weightedcodesarecodeswhicharenotassignedwithanyweighttoeachdigitposition,i.e.,eachdigit position within the number is not assigned fixed value.

Example:-Excess-3(XS-3)codeandGraycodes

BINARYCODEDDECIMAL(BCD):-

BCDis a weighted code. In weighted codes, each successive digit from right to left represents weights equal to some specified value and to get the equivalent decimal number add the products of the weights by the correspondingbinary digit. 8421 is themost commonbecause 8421 BCD is themostnatural amongstthe other possible codes.

For example:-

$(567)_{10}$isencodedinvarious4bit codes.

Solution:-

Decimal	→	5	6	7
8421code	→	0101	0110	0111
6311code	→	0111	1000	1001
5421code	→	1000	0100	1010

BCDADDITION:-

Addition of BCD(8421) is performedbyadding two digits of binary,startingfrom least significant digit. In caseif the result is an illegal code (greater than 9) or if there is a carry out of one then add 0110(6) and add the resulting carry to the next most significant.

For example:-

Add679.6from536.8usingBCDaddition.

Solution:-

679.6 011001111001.0110 (679.6in BCD)

+536.8 =>+01010011 0110.1000 (536.8 in BCD)

1216 .4 10111010 1111.1110 (Allareillegalcodes)

+0110+0110+0110.+0110 (Add 0110to each)

0001 0010 00010110.0100

1 2 1 6 . 4 (correctedsum = 1216.4)

Resultis**1216.4**

BCDSUBTRACTION:-

The BCD subtraction is performed by subtracting the digits of each 4 –bit group of the subtrahend from corresponding 4 –bit group of the minuend in the binary starting from the LSD. If there is no borrow from the next higher group[then no correction is required. If there is a borrow from the next group, then 6_{10} (0110) is subtracted from the difference term of this group.

For example:-

Subtract147.8from206.7using8421BCD code.

Solution:-

206.7 00100000 0110.0111 (206.7in BCD)

-147.8 =>-00010100 0111.1000 (147.8 in BCD)

58 .9 000010111110.1111 (Borrowsarepresent)

-0110-0110.-0110

01011000.1001

5 8 . 9 (correcteddifference=58.9)

Resultis**$(58.9)_{10}$**

EXCESSTHREE(XS-3)CODE:-

The Excess-3 code, also called XS-3, is a non- weighted BCD code. This derives it name from the fact thateach binary code word is the corresponding 8421 code word plus 0011(3). It is a sequential code. It is a self complementing code.

XS-3 ADDITION:-

In XS-3 addition, add the XS-3 numbers by adding the 4 bit groups in each column starting from the LSD. If there is no carry out from the addition of any of the 4 bit groups, subtract 0011 from the sum term of those groups. If there is a carry out, add 0011 to the sum term of those groups

For example:-

Add37and28usingXS-3 code.

Solution:-

37 01101010 (37 inXS-3)

+28 => + 01011011 (28inXS-3)

65 10111010 (Carryisgenerated)

+ 1 (Propagatecarry)

1100 0101 (Add 0110tocorrect0101and

-0011+0011 subtract0011tocorrect1100)

1001 1000 (CorrectedsuminXS-3= 65_{10})

XS-3 SUBTRACTION:-

To subtract in XS-3 number by subtracting each 4-bit group of the subtrahend from the corresponding 4-bit group of the minuend starting from the LSD. If there is no borrow from the next 4-bit group. add 0011 to the difference term of such groups. If there is a borrow, subtract 0011 from the differenceterm.

Forexample:-

. **Subtract175from267usingXS-3code**.

Solution :-`

267 01011010 1010 (267 inXS-3)

-175 => -01001010 1000(175in XS-3)

092 000011110010 (Correct 0010and0000byadding0011and

+0011-0011+0011 correct1111bysubtracting0011)

0011 11000101 (CorrecteddifferenceinXS-3=92_{10})

ASCIICODE:-

The American Standard Code for Information Interchange (ASCII) pronounced as 'ASKEE' is widely used alphanumeric code. This is basically a 7 bit code. The number of different bit patterns that can be created with7 bits is 27 = 128 ,the ASCII can be used to encode both the uppercase and lowercase characters of the alphabet (52 symbols) and some special symbols in addition tothe 10 decimal digits.Itisusedextensively for printers and terminals that interface with small computer systems. The table shown below shows the ASCII groups.

TheASCII code

LSBs	**MSBs**							
	000	001	010	011	100	101	110	111
0000	NUL	DEL	Space	0	@	P	P	
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s

0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DLE

EBCDICCODE:-

The Extended Binary Coded Decimal Interchange Code (EBCDIC) pronounced as ‘eb –si- dik’ is an 8 bit alphanumeric code. Since 28 =256 bitpatternscan be formed with 8 bits.Itis used by mostlarge computersto communicate in alphanumeric data. The table shown below shows the EBCDIC code.

TheEBCDICcode

LSD (Hex)	**MSD(Hex)**															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	DS		SP	&							[	]	\	0
1	SOH	DC1	SOS				/		a	j	~		A	J		1
2	STX	DC2	FS	SYN					b	k	s		B	K	S	2
3	ETX	DC3							c	l	t		C	L	T	3
4	PF	RES	BYP	PN					d	m	u		D	M	U	4
5	HT	NL	LF	RS					e	n	v		E	N	V	5
6	LC	BS	EOB	YC					f	o	w		F	O	W	6
7	DEL	IL	PRE	EOT					g	p	x		G	P	X	7
8		CAN							h	q	y		H	Q	Y	8
9		EM							i	r	z		I	R	Z	9
A	SMM	CC	SM		Ø	!	I	:								
B	VT				.	$	,	#								
C	FF	IFS		DC4	<	*	%	@								
D	CR	IGS	ENQ	NAK	(	)	_	‘								
E	SO	IRS	ACK		+	;	>	=								
F	SI	IUS	BEL	SUB	I	‘	?	‘								

GRAYCODE:-

The gray code is a non-weighted code. It is not a BCD code. It is cyclic code because successive words in this differ in one bit position only i.e it is a unit distance code.

Gray code is used in instrumentation and data acquisition systems where linear or angulardisplacement is measured. They are also used in shaft encoders, I/O devices, A/D converters and other peripheral equipment.

BINARY-TO–GRAYCONVERSION:-

Ifan n-bit binary numberis representedbyB_nB_{n-1}- -- --$B_1$and its gray code equivalent byG_nG_{n-1}----------G_1, where B_n and G_n are the MSBs , thengray code bits are obtainedfrom thebinary code asfollows

$G_n = B_n$

$G_{n-1} = B_n \oplus B_{n-1}$

.

.

.

.

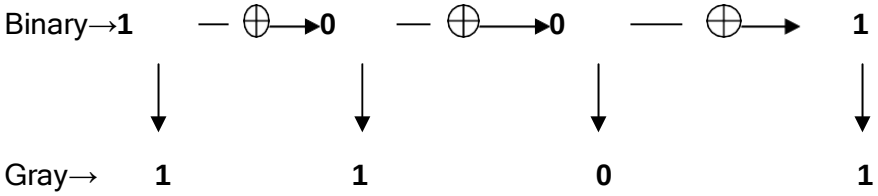
$G_1 = B_2 \oplus B_1$

Wherethesymbol$\oplus$standsforExclusive OR (X-OR)

Forexample:-

Convertthebinary1001totheGraycode.

Solution:-`



Thegraycodeis **1101**

GRAY-TO-BINARYCONVERSION:-

Ifan n-bit graynumber isrepresentedbyG_nG_{n-1}-------G_1andits binaryequivalent byB_nB_{n-1}----------B_1, thenbinarybitsareobtainedfromGraybitsasfollows: B_n

$= G_n$

$B_{n-1} = B_n \oplus G_{n-1}$

.

.

.

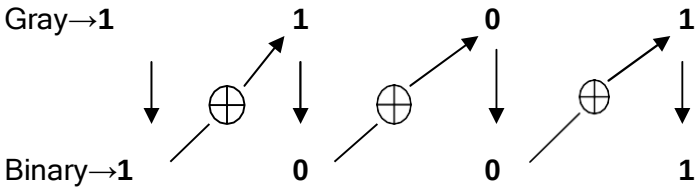
.

$B_1 = B_2 \oplus G_1$

Forexample:-

ConverttheGraycode1101tothebinary.

Solution :-



The binarycodeis**1001**

LOGICGATES

LOGICGATES:-

- Logicgatesarethefundamentalbuildingblocksofdigitalsystems.
- Thereare3 basictypesofgatesAND,ORand NOT.
- Logicgatesareelectroniccircuitsbecausetheyaremadeupofanumberofelectronicdevicesand components.
- Inputsandoutputsoflogicgatescanoccuronlyin2levels.ThesetwolevelsaretermedHIGHand LOW, or TRUE and FALSE, or ON andOFF or simply 1 and 0.
- The table which lists all the possible combinations of input variables and the corresponding outputs iscalled a truth table.

LEVELLOGIC:-

A logic inwhichthe voltage levels representslogic 1 andlogic0. Level logic maybe positive or negative logic.

Positive Logic:-

Apositivelogic systemistheone in which thehigher ofthe twovoltagelevels represents thelogic1and the lower of the two voltages level represents the logic 0.

NegativeLogic:-

Anegativelogic system is theone in whichthelowerofthe twovoltagelevels represents thelogic1and the higher of the two voltages level represents the logic 0.

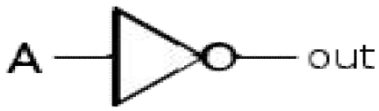
DIFFERENTTYPESOFLOGICGATES:-

NOT GATE (INVERTER):-

- ANOT gate,alsocalledandinverter, hasonlyone inputandoneoutput.
- Itisadevicewhoseoutputisalwaysthecomplementofits input.
- The output of a NOT gate is the logic 1 state when its input is in logic 0 state and the logic 0 state when its inputs is in logic 1 state.

ICNo.:-7404

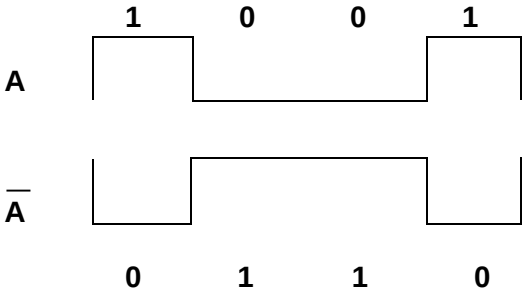
LogicSymbol



Truthtable

INPUT A	OUTPUT $\overline{A}$
0	1
1	0

TimingDiagram

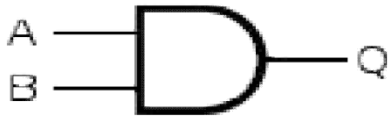


ANDGATE:-

- AnANDgate hastwo ormoreinputsbut onlyone output.
- Theoutput is logic1 state onlywheneach one ofits inputsis atlogic 1 state.
- The output is logic0 state even if oneof its inputs isat logic 0 state.

ICNo.:-7408

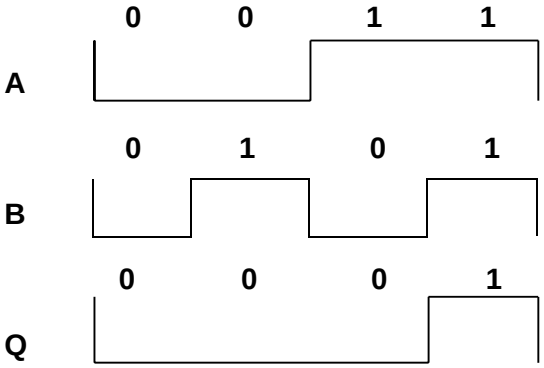
LogicSymbol



TruthTable

		OUTPUT
A	B	Q=A.B
0	0	0
0	1	0
1	0	0
1	1	1

TimingDiagram



ORGATE:-

- AnORgatemayhave two or more inputs but onlyone output.
- The output is logic1 state,even if one of its inputis inlogic 1 state.
- Theoutput islogic0 state,onlywhen each oneof its inputsisin logicstate.

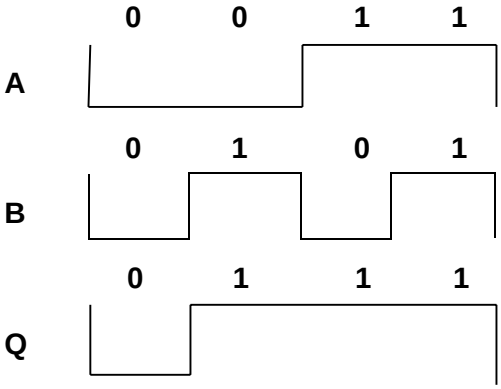
ICNo.:-7432

LogicSymbol

TruthTable

INPUT		OUTPUT
A	B	Q=A+B
0	0	0
0	1	1
1	0	1
1	1	1

TimingDiagram



NANDGATE:-

- NANDgateisacombination of an ANDgate anda NOTgate.
- Theoutputislogic0wheneachoftheinputislogic1andforanyothercombinationofinputs,the output is logic 1.

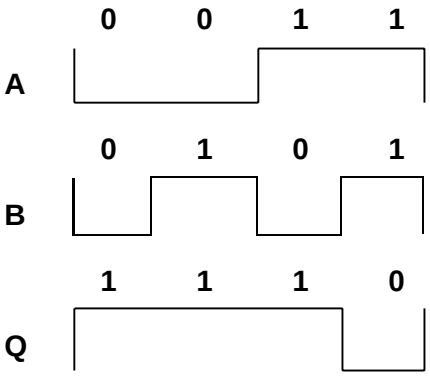
IC No.:- 7400 two input NAND gate
7410threeinputNANDgate
7420 four input NAND gate
7430eightinputNANDgate

LogicSymbol

TruthTable

INPUT		OUTPUT
A	B	$Q=\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

TimingDiagram



NORGATE:-

- NORgate is acombination of an ORgateanda NOT gate.
- The output is logic 1, only when each one of its input is logic 0 and for any other combination of inputs, the output is a logic 0 level.

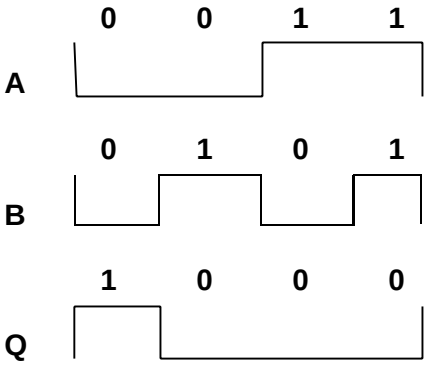
IC No.:- 7402 two input NOR gate
7427threeinputNORgate
7425 four input NOR gate

LogicSymbol

TruthTable

INPUT		OUTPUT
A	B	$Q=\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

TimingDiagram



EXCLUSIVE–OR(X-OR)GATE:-

- AnX-ORgateisatwoinput,one outputlogic circuit.
- The output is logic 1 when one and only one ofits two inputs is logic 1. When both the inputs is logic 0or when both the inputs is logic 1, the output is logic 0.

ICNo.:-7486

LogicSymbol

INPUTS are **A** and **B**

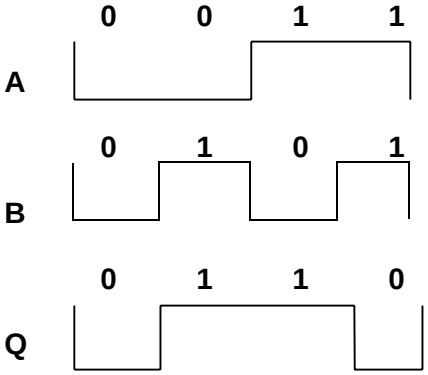
OUTPUT is**Q**= A⊕B

$= A\bar{B} + \bar{A}B$

TruthTable

INPUT		OUTPUT
A	B	Q =A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

TimingDiagram

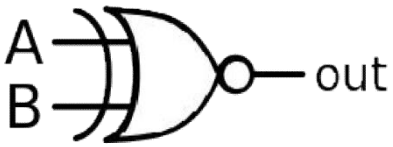


EXCLUSIVE–NOR(X-NOR)GATE:-

- AnX-NORgateis thecombinationof anX-ORgate anda NOT gate.
- AnX-NORgateis atwoinput, oneoutputlogic circuit.
- Theoutput is logic1onlywhen both the inputsarelogic0 or when boththe inputs is1.
- Theoutputis logic0 whenone ofthe inputsislogic0and otheris 1.

ICNo.:-74266

LogicSymbol

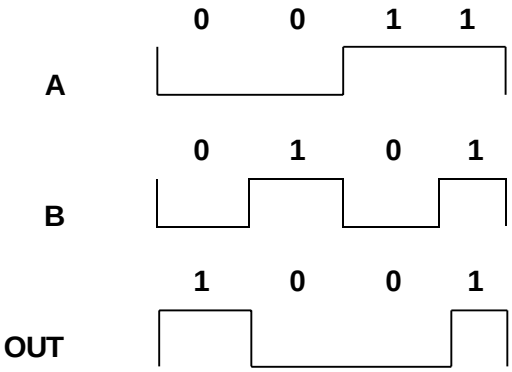


INPUT		OUTPUT
A	B	OUT=AXNOR B
0	0	1
0	1	0
1	0	0
1	1	1

OUT=AB $\bar{}$ +A$\bar{B}$

=AXNORB

TimingDiagram

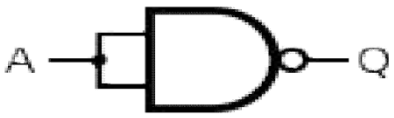


UNIVERSALGATES:-

There are 3 basic gates AND, OR and NOT, there are two universal gates NAND and NOR, each of which can realize logic circuits single handedly. The NAND and NOR gates are called universal building blocks. Both NAND and NOR gates can perform all logic functions i.e. AND, OR, NOT, EXOR and EXNOR.

NANDGATE:-

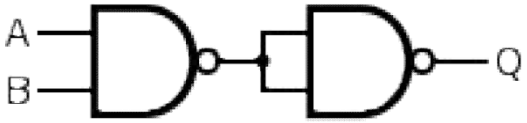
a) **InverterfromNANDgate**



Input **=A**
Output**Q=A** $\bar{}$

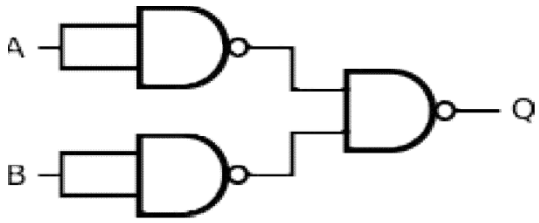
b) **ANDgatefromNANDgate**

Inputsare **A**and**B** Output
Q = A.B



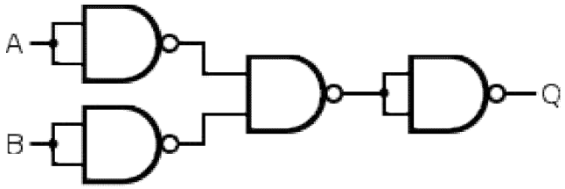
c) **ORgatefromNANDgate**

Inputsare**A**and**B**
Output **Q = A+B**



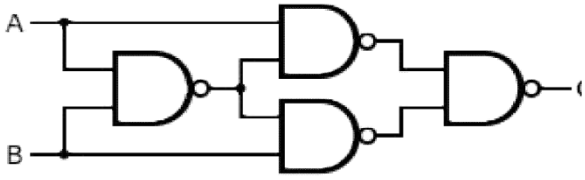
d) NORgatefromNANDgate

Inputsare**A**and**B** Output **Q = A+B**



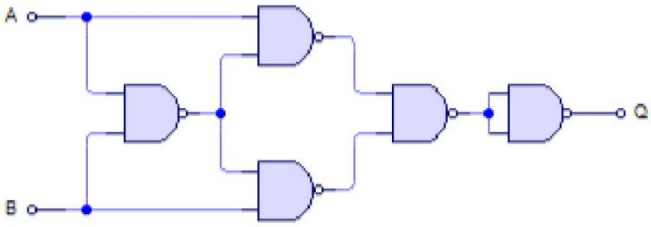
e) EX-ORgatefromNANDgate

Inputs are **A** and **B**
Output**Q=AB+AB**



f) EX-NORgateFromNANDgate

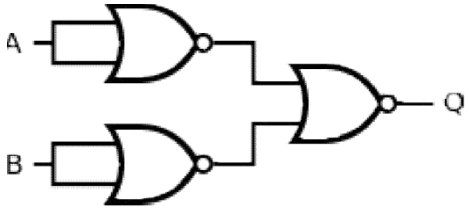
Inputs are **A**and **B**
Output**Q=AB+AB**



NORGATE:-

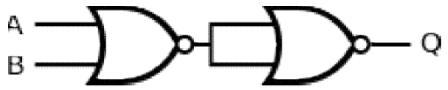
a) InverterfromNORgate Input **= A**
Output**Q=A**

b) ANDgatefromNORgate Input s are **A** and **B**Output **Q = A.B**



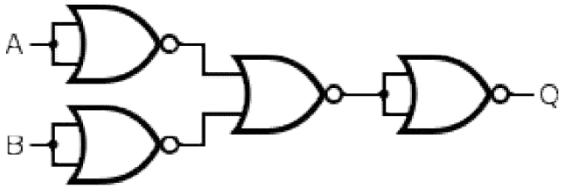
c) ORgatefromNORgate

InputsareAandB Output
Q = A+B



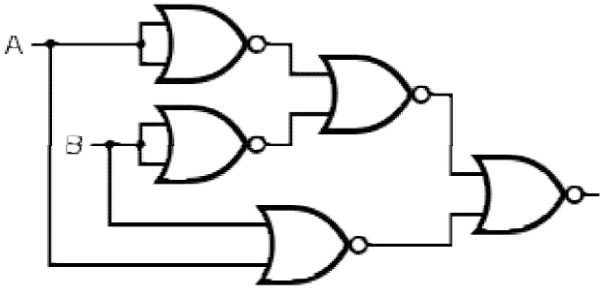
d) NANDgatefromNORgate

InputsareAandB Output
Q = A.B



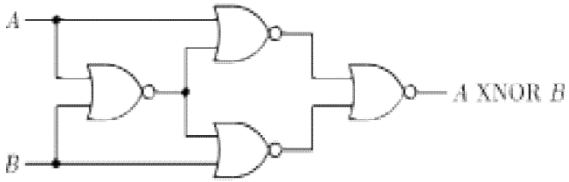
e) EX-ORgatefromNORgate

Inputs are **A** and **B**
Output**Q=AB+AB**



f) EX-NORgateFromNORgate

Inputs are **A**and **B**
Output**Q=AB+AB**



THRESHOLDLOGIC:-

INTRODUCTION:-

- The threshold element, also calledthe threshold gate (T-gate) is a much more powerful device thanany of the conventional logic gates such as NAND, NOR and others.
- Complex,largeBooleanfunctionscanberealizedusingmuchfewerthresholdgates.
- Frequently a single threshold gate can realize a very complex function which otherwise might require a large number of conventional gates.
- T-gateoffersincomparablyeconomical realization; it has not foundextensive use with the digital system designers mainly because of the following limitations.
  1. Itisverysensitivetoparameter variations.
  2. ItisdifficulttofabricateitinICform.

3. Thespeedofswitchingofthresholdelementsinmuchlowerthanthatofconventionalgates.

THETHRESHOLDELEMENTS:-

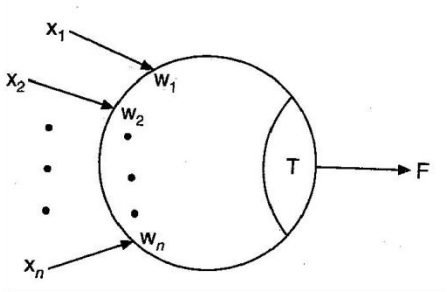
- A threshold element or gate has 'n' binary inputs x_1, x_2, ….., x_n; and a single binary output F. But in additiontothose,ithastwomoreparameters.
- Itsparametersareathresholdandweightsw_1,w_2,….,w_n.Theweightsw_1,w_2,…,$w_n$areassociated with the input variables x_1, x_2, …,x_n.
- Thevalueof thethreshold (T)andweightsmaybereal,positive ornegative number.
- Thesymbolofthethresholdelementisshowninfig.(a).
- Itisrepresentedbyacircle partitionedintotwoparts,onepartrepresentstheweightsandother represents T.
- Itis definedas

  $F(x_1,x_2,.....,x_n)=1$ ifandonlyif $\sum_{i=1}^{n} w_i x_i \geq T$

  otherwise

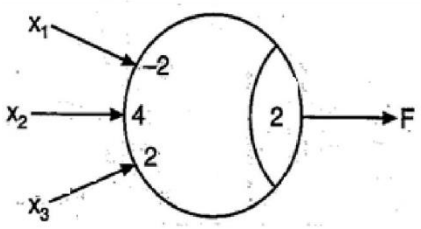
  $F(x_1,x_2,.....,x_n)=0$

- Thesumandproductoperationarenormalarithmeticoperationsandthesum $\sum_{i=1}^{n} w_i x_i \geq T$ iscalledtheweightedsumoftheelementor gate.



Example:-

ObtaintheminimalBooleanexpressionfromthethresholdgateshowninfigure.



Solution:-

The threshold gate with three inputs x_1, x_2, x_3 with weights -2(w_1) , 4(w_2) and 2(w_3) respectively. The value of threshold is 2(T). The table shown is the weighted sums and outputs for all input combinations. For this threshold gate, the weighted sum is

$w=w_1x_1+w_2x_2+w_3x_3$

$= (-2)x_1 +(4)x_2 +(2)x_3$

$=-2x_1+4x_2+2x_3$

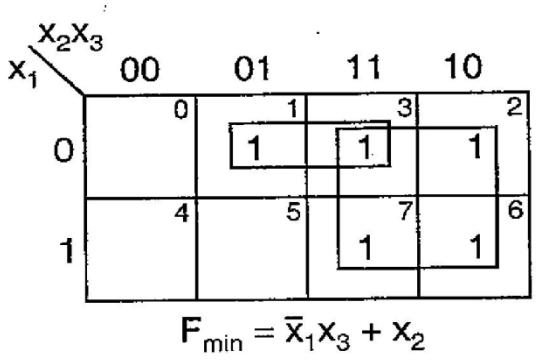
The output F is logic1for w≥2andit islogic0 forw<2

InputVariables			WeightedSum	Output
x_1	x_2	x_3	$w=-2x_1+4x_2+2x_3$	F
0	0	0	0	0
0	0	1	2	1
0	1	0	4	1
0	1	1	6	1
1	0	0	-2	0
1	0	1	0	0
1	1	0	2	1
1	1	1	4	1

From the input – output relation is given in the table, the Boolean expression for the output is

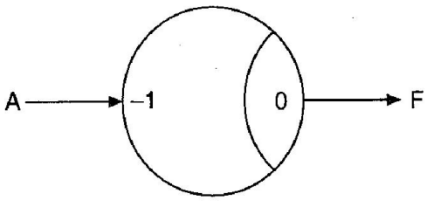
$$F=\sum m\ (1, 2, 3, 6, 7)$$

The K-mapforFis



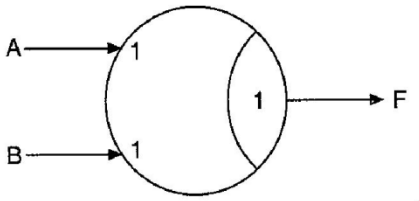
UNIVERSALITYOFAT-GATE:-

- AsingleT-gatecanrealizealargenumberoffunctionsbymerelychangingeithertheweightsorthe threshold or both, which can be done by altering the value of the corresponding resistors.
- Since a threshold gate can realize universal gates, i.e.,NAND gates and NOR gates,a threshold gateis also a universal gate.
- SinglethresholdgatecannotrealizebyasingleT-gate
- RealizationoflogicgatesusingT-gatesisshowninthebelowfigure.



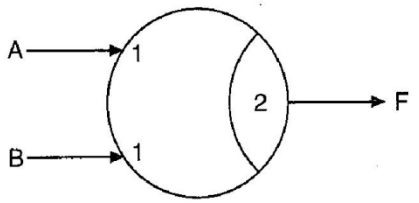
(a) NOT gate $F = \overline{A}$

Input	Weighted sum	Output
A	W = −A	F
0	0	1
1	−1	0



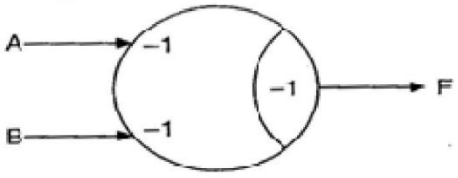
(b) OR gate F = A + B

Inputs		Weighted sum	Output
A	B	w = A + B	F
0	0	0	0
0	1	1	1
1	0	1	1
1	1	2	1



(c) AND gate F = AB

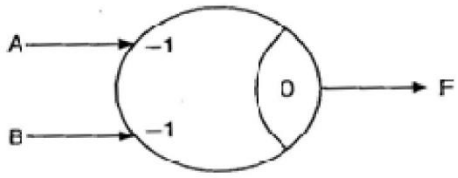
Inputs		Weighted sum	Output
A	B	w = A + B	F
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1



NAND gate

(d) $F = \overline{A} + \overline{B} = \overline{AB}$

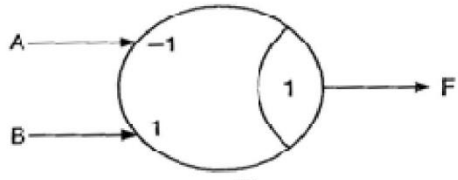
Inputs		Weighted sum	Output
A	B	w = −A − B	F
0	0	0	1
0	1	−1	1
1	0	−1	1
1	1	−2	0



NOR gate

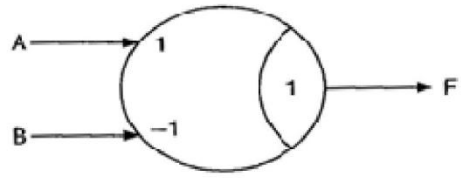
(e) $F = \overline{A} \cdot \overline{B} = \overline{A + B}$

Inputs		Weighted sum	Output
A	B	w = −A − B	F
0	0	0	1
0	1	−1	0
1	0	−1	0
1	1	−2	0



(f) $F = \overline{A}B$

Inputs		Weighted sum	Output
A	B	w = −A + B	F
0	0	0	0
0	1	1	1
1	0	−1	0
1	1	0	0



(g) $F = A\overline{B}$

Inputs		Weighted sum	Output
A	B	w = A − B	F
0	0	0	0
0	1	−1	0
1	0	1	1
1	1	0	0

BOOLEANALGEBRA

INTRODUCTION:-

- Switchingcircuitsarealsocalledlogiccircuits,gatescircuitsanddigitalcircuits.
- SwitchingalgebraisalsocalledBooleanalgebra.
- Booleanalgebraisasystemofmathematicallogic.Itisanalgebraicsystemconsistingofthesetof elements (0,1), two binary operators called OR and AND and unary operator called NOT.
- Itisthebasicmathematicaltoolintheanalysisandsynthesisofswitchingcircuits.
- Itis awaytoexpress logicfunctionsalgebraically.
- AnycomplexlogiccanbeexpressedbyaBooleanfunction.
- TheBooleanalgebraisgovernedbycertainwelldevelopedrulesandlaws.

AXIOMSANDLAWSOFBOOLEANALGEBRA:-

Axioms or postulates of Boolean algebra are set of logical expressions that are accepted without proof and upon which we can build a set of useful theorems. Actually, axioms are nothing more than the definitions of the three basic logic operations AND, OR and INVERTER. Each axiom can be interpreted as the outcome of an operation performed by a logic gate.

ANDoperation	**ORoperation**	**NOToperation**
Axiom1:0.0=0	Axiom5:0+0=0	Axiom9:$\overline{1}$=0
Axiom2:0.1=0	Axiom6:0+1=1	Axiom10:$\overline{0}$=1
Axiom3:1.0=0	Axiom7:1+0=1	
Axiom2:1.1=1	Axiom8:1+1=1	

1.ComplementationLaws:-

The term complement simply means to invert, i.e. to changes 0s to 1s and 1s to 0s. The five laws of complementation are as follows:

Law1:0 = $\overline{1}$

Law2:1= $\overline{0}$

Law3:ifA=0,thenA=1 **Law 4:**

if $\underline{A}$ = 1,thenA = 0

Law5: $\overline{A}$=0(doublecomplementationlaw)

2. ORLaws:-

The four OR laws are as follows

Law1:A+0=0(Nulllaw)

Law2:A+1=1(Identitylaw) **Law 3:**A + A = A

Law4:A+A=$\overline{1}$

3. ANDLaws:-

The four AND laws are as follows

Law 1:A . 0 = 0(Null law)

Law2:A.1=1(Identitylaw) **Law 3:**A . A = A

Law4:A .A=$\overline{0}$

4. CommutativeLaws:-

CommutativelawsallowchangeinpositionofANDorORvariables.Therearetwocommutativelaws.

Law 1:A+B=B+A

Proof

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

=

B	A	B+A
0	0	0
0	1	1
1	0	1
1	1	1

Law2:A.B=B.A

Proof

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

=

B	A	B.A
0	0	0
0	1	0
1	0	0
1	1	1

Thislawcanbeextendedtoanynumberofvariables.Forexample
A.B.C=B.C.A=C.A.B=B.A.C

5. AssociativeLaws:-

Theassociativelawsallowgroupingofvariables.Thereare2associativelaws.

Law1:(A+B)+ C=A+(B+C)

Proof

A	B	C	A+B	(A+B)+C
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

=

A	B	C	B+C	A+(B+C)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Law 2:(A.B)C=A(B.C)

Proof

A	B	C	AB	(AB)C
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

=

A	B	C	B.C	A(B.C)
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Thislawcanbeextendedtoanynumberofvariables.Forexample A(BCD) =
(ABC)D = (AB) (CD)

6. DistributiveLaws:-

Thedistributivelawsallowfactoringormultiplyingoutofexpressions.Therearetwodistributivelaws.

Law1:A(B+C)=AB+AC

Proof

A	B	C	B+C	A(B+C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

=

A	B	C	AB	AC	A+(B+C)
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Law2:A+BC=(A+B)(A+C)

Proof **RHS** = (A+B) (A+C)
=AA +AC+BA+BC
=A+AC+AB+BC
=A(1+C+B)+BC
=A. 1+BC (1+C+B=1+ B=1)
=A+BC
=**LHS**

7. RedundantLiteralRule(RLR):-

Law 1: $A + \bar{A}B = A + B$

Proof

$A+A\bar{B}=(A+A)\ (\bar{A}+B)$

$=1.(A+B)$

$=A+B$

Law2: $A(A\bar{+}B)=AB$ A(A +

Proof

B) $\bar{=}$ AA + $A\bar{B}$

=0 +AB

=AB

8. Idempotence Laws:-

Idempotencemeanssamevalue.

Law1: A**.** A=A

Proof

If A = 0, then A**.** A = 0**.** 0 =0 = A

If A= 1, thenA**.**A = 1**.** 1= 1 = A

ThislawstatesthatANDofa variablewithitself isequaltothatvariableonly.

Law2: A+A=A

Proof

If A= 0, thenA+ A = 0+ 0= 0 = A IfA

=1,thenA+A=1+1 =1=A

ThislawstatesthatORofavariablewithitselfisequaltothatvariableonly.

9. AbsorptionLaws:-

There are two laws:

Law 1: A+A·B=A

Proof

A+A ·B=A(1+B)=A ·1=A

A	B	AB	A+AB
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Law2: A(A+B)=A

Proof

A(A +B) =A ·A+A ·B=A+AB=A(1+ B)=A· 1=A

A	B	A+B	A(A+B)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

10. ConsensusTheorem(IncludedFactorTheorem):-

Theorem 1:

$AB+\bar{A}C+BC=AB+\bar{A}C$

Proof

LHS= $AB+\bar{A}C+BC$

$=AB+AC+BC(A+A)$

$=AB+AC+BCA+BCA$

$=AB(1+C)+\bar{A}C(1+B)$

$=AB(1)+\bar{A}C(1)$

$=AB+\bar{A}C$

=**RHS**

Theorem2:

$(A+B)(\bar{A}+C)(B+C)=(A+B)(\bar{A}+C)$

Proof

LHS $=(A+B)(\bar{A}+C)(B+C)$

$=(A\bar{A}+AC+BA+\bar{B}C)(B+C)$

$=(AC+BC+AB)(\bar{B}+C)$

$=ABC+BC+AB+AC+BC+ABC$

$=AC+BC+AB$

RHS $=(A+B)(\bar{A}+C)$

$=A\bar{A}+AC+BC+AB$

$=AC+BC+AB$

=LHS

11. TranspositionTheorem:-

Theorem:

$AB+\bar{A}C=(A+C)(\bar{A}+B)$

Proof

RHS= $(A+C)(\bar{A}+B)$

$=A\bar{A}+CA+\bar{A}B+CB$

$=0+AC+AB+BC$

$=AC+AB+BC(A+A)$

$=AB+ABC+AC+ABC$

$=AB+AC$

=LHS

12. DeMorgan'sTheorem:-

DeMorgan'stheoremrepresentstwolawsinBooleanalgebra. **Law**

1: $\overline{A+B}=\bar{A}\cdot\bar{B}$

Proof

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

=

A	B	$\bar{A}$	$\bar{B}$	$\bar{A}\bar{B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Thislawstatesthatthecomplementofasumofvariablesisequaltotheproductoftheirindividual complements.

Law2: $A \cdot B = \overline{A} + \overline{B}$

Proof

A	B	A.B	$\overline{A.B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

=

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

Thislawstatesthatthecomplementofaproductofvariablesisequaltothesumoftheirindividual complements.

DUALITY:-

The implication of the duality concept is that once a theorem or statement is proved, the dual also thus standproved. This is called the principle of duality.

$[f(A,B,C,\ldots..,0,1,+,\cdot)]_d = f(A,B,C,\ldots.,1,0,\cdot,+)$

Relationsbetweencomplementanddual

$f_c(A,B,C,\ldots..) = f(\overline{A},\overline{B},\overline{C},\ldots.) = f_d(A,B,C,\ldots)$

$f_d(A,B,C,\ldots..) = f(A,B,\overline{C},\ldots) = \overline{f_c(A,B,C,\ldots..)}$

Thefirstrelationstatesthatthecomplementofafunctionf(A,B,C,…)canbeobtainedbycomplementingall the variables in the dualfunctionf_d(A, B, C, …..).

The secondrelationstates that the dualcan be obtained bycomplementing all the literals in f (A, B, C, ….).

DUALS:-

	Givenexpression	***Dual***
1.	$0=\overline{1}$	$\overline{1}=0$
2.	$0 \cdot 1 = 0$	$1+0=1$
3.	$0 \cdot 0 = 0$	$1+1=1$
4.	$1 \cdot 1 = 1$	$0+0=0$
5.	$A \cdot 0 = 0$	$A+1=1$
6.	$A \cdot 1 = A$	$A+0=A$
7.	$A \cdot A = A$	$A+A=A$
8.	$A \cdot \overline{A} = 0$	$A+\overline{A}=1$
9.	$A \cdot B = B \cdot A$	$A+B=B+A$
10.	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A+(B+C)=(A+B)+C$
11.	$A \cdot (B+C) = AB + AC$	$A+BC=(A+B)(A+C)$
12.	$A(A+B)=A$	$A+AB=A$
13.	$A \cdot (\overline{A} \cdot B) = A \cdot B$	$A+\overline{A}+B=A+B$
14.	$AB = \overline{A} + B$	$A+B=\overline{AB}$
15.	$(A+B)(\overline{A}+C)(B+C) = (A+B)(\overline{A}+C)$	$\overline{AB} + \overline{A}C + BC = AB + AC$
16.	$A + BC = (A+B)(A+C)$	$A(B+C) = AB + AC$
17.	$(A+C)(\overline{A}+B) = AB + \overline{A}C$	$AC + \overline{A}B = (A+B)(\overline{A}+C)$
18.	$(A+B)(C+D) = AC + AD + BC + BD$	$(AB+CD) = (A+C)(A+D)(B+C)(B+D)$
19.	$A + B = AB + \overline{A}\overline{B}$	$AB = (A+B)(\overline{A}+\overline{B})$
20.	$\overline{A}\overline{B} + \overline{A} + AB = 0$	$\overline{A} + B \cdot \overline{A} \cdot (A+B) = 1$

SUM-OF-PRODUCTS FORM:-

- This is also called disjunctive Canonical Form (DCF) or Expanded Sum of Products Form or Canonical Sum of Products Form.
- In thisform, thefunction is the sum of a number of products terms where each product term contains all variables of the function either in complemented or uncomplemented form.
- Thiscanalsobederivedfromthetruthtablebyfindingthesumofallthetermsthatcorrespondsto those combinations for which 'f ' assumes the value 1.
  Forexample
  $f(A,B,C)=A\bar{B}+B\bar{C}$
  $=A\bar{B}(C+\bar{C})+B\bar{C}(A+\bar{A})$
  $=A\bar{B}C+A\bar{B}\bar{C}+\bar{A}B\bar{C}+\bar{A}\bar{B}\bar{C}$
- Theproducttermwhichcontainsallthevariablesofthefunctionseitherincomplementedor uncomplemented form is called a minterm.
- Themintermisdenotedasmo,m1,m2....
- An'n'variablefunctioncanhave2nminterms.
- Another way of representing the function in canonical SOP form is the showing the sum of minterms for which the function equals to 1.
  Forexample
  $f(A,B,C)=m_1+m_2+m_3+m_5$
  or
  $f(A,B,C)=\sum m(1,2,3,5)$
  where$\sum$mrepresentsthesumofallthemintermswhosedecimalcodesaregiventhe parenthesis.

PRODUCT-OF-SUMSFORM:-

- ThisformisalsocalledasConjunctiveCanonicalForm(CCF)orExpandedProduct-of–SumsForm orCanonicalProductOfSumsForm.
- Thisisbyconsideringthecombinationsforwhichf=0
- Each termis asumof allthe variables.
- Thefunctionf(A,B,C)=$(A+B+C\cdot C)+(\bar{A}+\bar{B}+C\cdot C)$
  $=(A+B+C)(A+B+C)(\bar{A}+\bar{B}+C)(\bar{A}+B+C)$
- The sum term which contains each of the 'n'variables ineither complemented or uncomplemented form is called a maxterm.
- MaxtermisrepresentedasM0,M1,M2,.....
  Thus CCF of 'f' may be written
  asf(A,B,C)=$M_0\cdot M_4\cdot M_6\cdot M_7$
  or
  f(A,B,C)=(0,4, 6,7)
  Whererepresentedtheproductofallmaxterms.

CONVERSIONBETWEENCANONICALFORM:-

The complement of afunctionexpressed asthe sum of minterms equalsthe sum of minterms missing from the original function.

Example:-

$f(A,B,C)=\sum m(0,2,4,6,7)$

Thishas acomplement that canbe expressedas

$\overline{f(A,B,C)}=\sum m(1,3,5)=m_1+m_3+m_5$

If we complement f by De- Morgan'stheoremwe obtain 'f' in aform. f

$=(\overline{m_1+m_3+m_5})=\overline{m_1}.\overline{m_3}.\overline{m_5}$

$= M_1 M_3 M_5 = \prod M(1,3,5)$

Example:-

Expand $A(A+B)(A+\bar{B}+C)$ to maxterms and minterms.

Solution:-

In POS form

$A(A+B)(A+\bar{B}+C)$ $A = A + BB + C\bar{C}$

$= (A+B)(A+\bar{B}) + C \cdot C$

$= (A+B+C\bar{C})(A+B+C\bar{C})$

$= (A + B + C)(A + \bar{B} + C)(\bar{A} + B + \bar{C})(A + B + C)$ $\bar{A} + B = A + \bar{B} + C \cdot C$

$= (\bar{A}+B+C)(A + \bar{B} + C)$

Therefore

$A(\bar{A}+B)(A+\bar{B}+C)$

$= (A+B+C)(A+B+C)(A+\bar{B}+C)(\bar{A}+B+C)(A+\bar{B}+C)(\bar{A}+\bar{B}+C)$

$= (000)(001)(010)(011)(100)(101)$

$= M_0 \cdot M_1 \cdot M_2 \cdot M_3 \cdot M_4 \cdot M_5$

$= \prod M(0,1,2,3,4,5)$

The maxterms M_6 and M_7 are missing in the POS form. So, the SOP form will contain the minterms 6 and 7

KARNAUGH MAP OR K-MAP:-

- The K- map is a chart or a graph, composed of an arrangement of adjacent cells, each representing a particular combination of variables in sum or product form.
- The K-map is systematic method of simplifying the Boolean expression.

TWO VARIABLE K-MAP:-

A two variable expression can have $2^2 = 4$ possible combinations of the input variables A and B.

Mapping of SOP Expression:-

- The 2 variable K-map has $2^2 = 4$ squares. These squares are called cells.
- A '1' is placed in any square indicates that corresponding minterm is included in the output expression, and a 0 or no entry in any square indicates that the corresponding minterm does not appear in the expression for output.

A \ B	0	1
0	$\bar{A}\bar{B}$	$\bar{A}B$
1	$A\bar{B}$	AB

Example:-

Map expression $f = A\bar{B} + AB$

Solution:-

The expression minterms is F

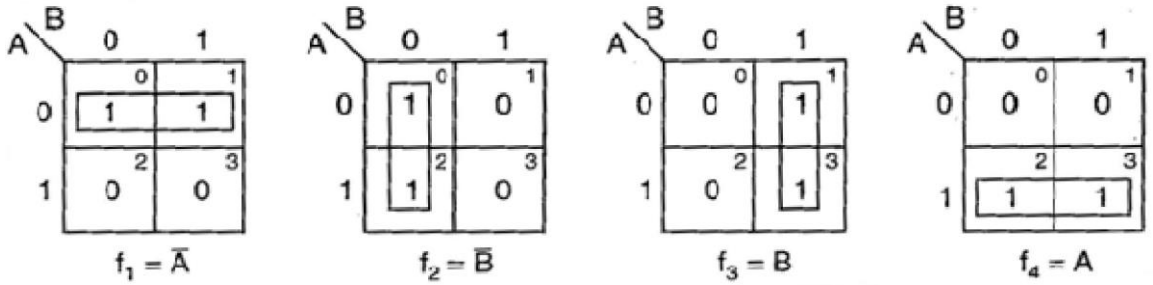
$= m_1 + m_2 = m(1, 2)$

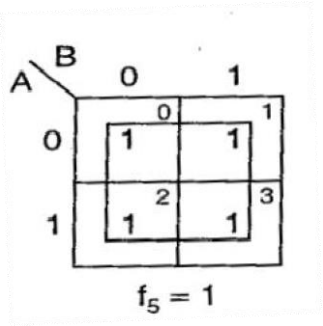
A \ B	0	1
0	0 (0)	1 (1)
1	1 (2)	0 (3)

MinimizationofSOPExpression:-

TominimizeaBooleanexpressiongivenintheSOPformbyusingK-map,theadjacentsquareshaving1s, that is minterms adjacent to each other are combined to form larger squares to eliminate some variables.

ThepossiblemintermgroupinginatwovariableK-mapareshownbelow





- Two minterms, which are adjacent to each other, can be combined to form a bigger square called 2 – square or a pair. This eliminates one variable that is not common to both the minterms.
- Two 2-squares adjacent to each other can be combined to form a 4- square. A 4- square eliminates 2 variables. A 4-square is called a quad.
- Consider only those variables which remain constant throughout the square, and ignore the variables which are varying. The non-complemented variable is the variable remaining constant as 1.The complemented variable is the variable remaining constant as a 0 and the variables are written as a product term.

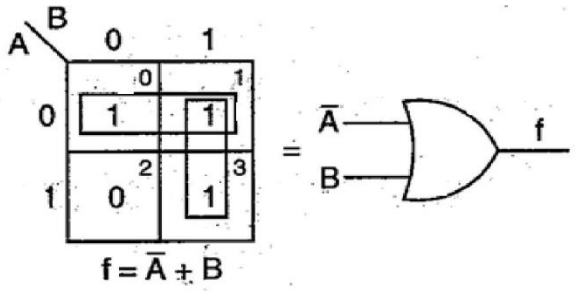
Example:-

Reduce theexpressionf= $A\bar{B}+A\bar{B}+AB$usingmapping.

Solution:-

Expressedintermsofminterms,thegivenexpressionis f =

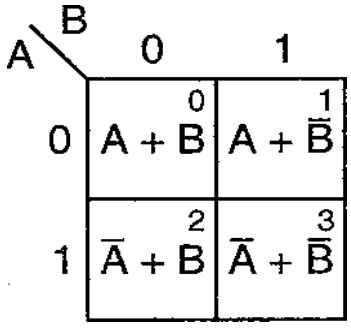
$$m_0 + m_1 + m_3 = \sum m\ (0, 1, 3)$$



F = $\bar{A}$+ B

Mappingof POSExpression:-

Each sum term in the standard POS expression is called a Maxterm. A function in two variables (A,B) has 4 possible maxterms, A + B, A + $\overline{B}$, $\overline{A}$ + B and $\overline{A}$ + $\overline{B}$. They are represented as M_0, M_1, M_2 and M_3 respectively.



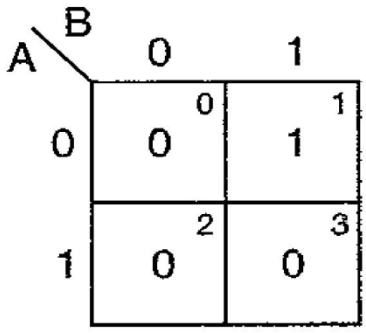
ThemaxtermofatwovariableK-map

Example:-

Plottheexpressionf=(A+B)(A+$\overline{B}$)(A+B)

Solution:-

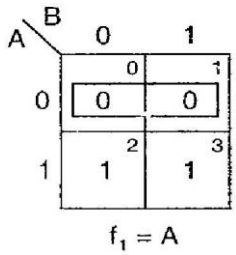
Expressionintermsofmaxtermsis f=ΠM (0,2,3)



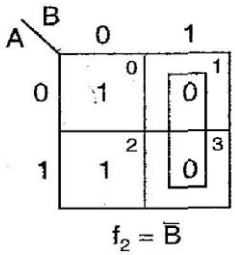
MinimizationofPOSExpressions:-

In POS form the adjacent 0s are combined into large square as possible. If the squares having complemented variable then the value remain constant as a 1 and the non-complemented variable if its value remainsconstant as a 0 along the entire square and then their sum term is written.

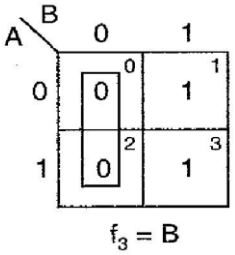
ThepossiblemaxtermsgroupinginatwovariableK-mapareshownbelow



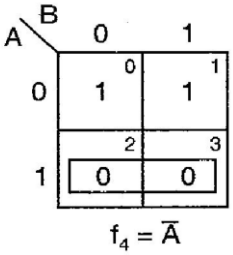
$f_1 = A$



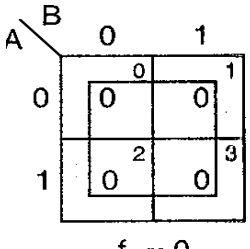
$f_2 = \overline{B}$



$f_3 = B$



$f_4 = \overline{A}$



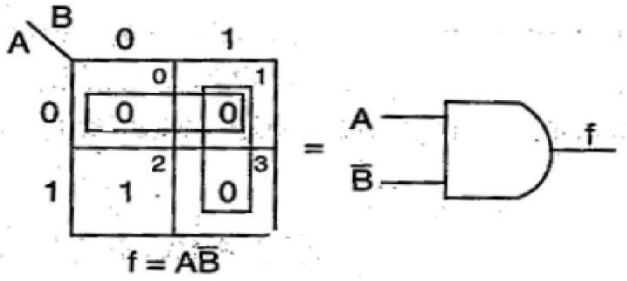
$f_5 = 0$

Example:-

Reduce the expressionf = (A+ B)(A+ B)(A+B) usingmapping

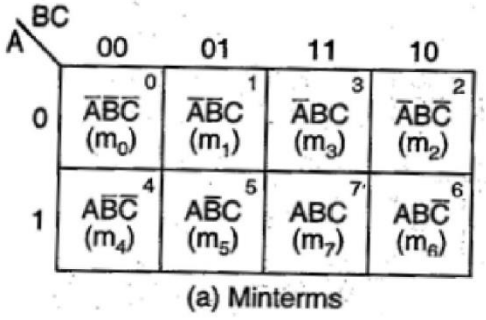
Solution:-

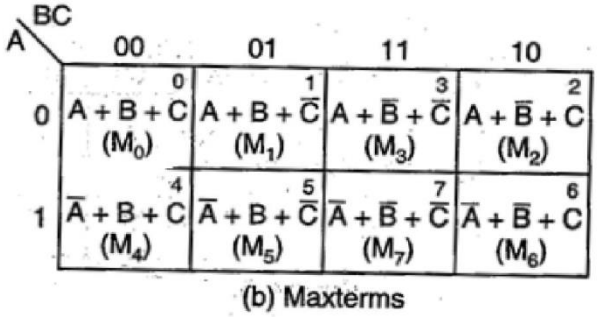
Thegivenexpressionintermsofmaxtermsis f=ΠM (0,1,3)



THREEVARIABLEK-MAP:-

Afunctioninthreevariables(A,B, C)canbeexpressedinSOPandPOSformhavingeightpossible combination.AthreevariableK-maphave8squaresorcellsandeachsquare minterm or onthemaprepresentsa maxterm is shown in the figure below.



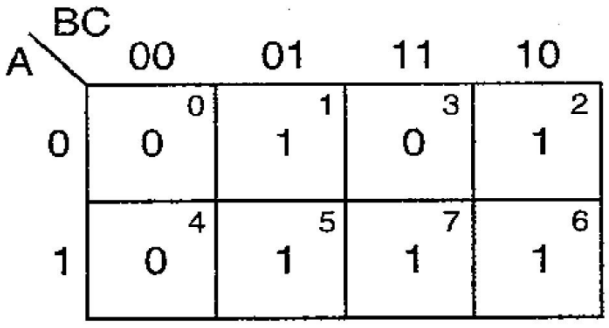


Example:-

Maptheexpressionf=ABC+ABC+ ABC+ ABC+ABC

Solution:-

So intheSOPformtheexpressionisf =∑m(1,5,2, 6,7)

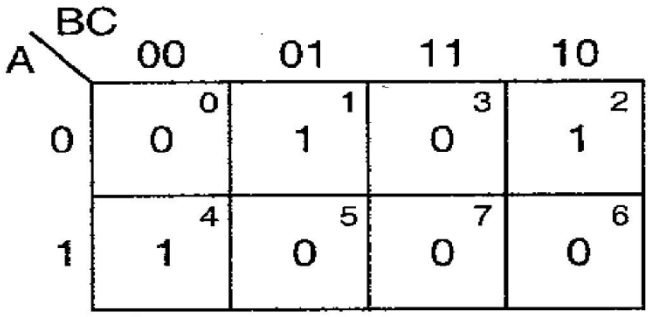


Example:-

Mapthe expressionf =(A+ B + C)(A+ B+C)(A+ B + C)(A+ B +C)(A+ B + C)

Solution:-

Soin thePOSformtheexpressionisf=ΠM(0,5,7,3,6)

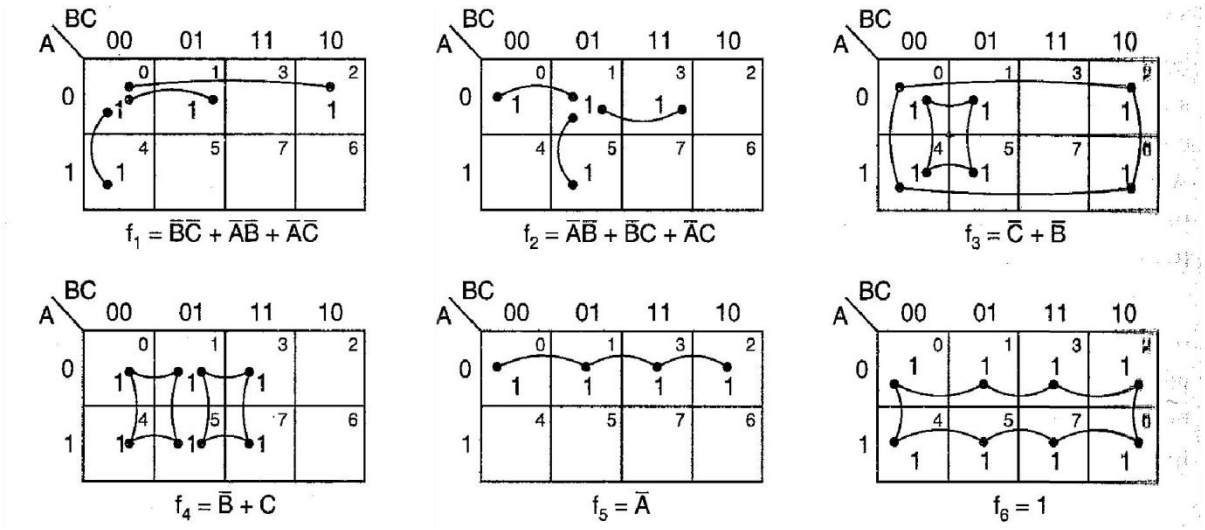


MinimizationofSOPandPOSExpressions:-

ForreducingtheBooleanexpressionsinSOP(POS)formthefollowingstepsaregiven below

- DrawtheK-mapandplace 1s(0s)correspondingtotheminterms(maxterms)oftheSOP(POS) expression.
- In the map 1s(0s) which are not adjacent to anyother 1(0) are the isolated minterms(maxterms). They are to be read as they are because they cannot be combined even into a 2-square.
- Forthose1s(0s)whichare adjacentto onlyoneother1(0)make thempairs (2squares).
- Forquads(4-squares)andoctet(8squares)ofadjacent1s(0s)eveniftheycontainsome1s(0s) which have already been combined. They must geometrically form a square or a rectangle.
- Forany1s(0s) thathavenotbeencombinedyetthencombinethemintobiggersquaresifpossible.
- Formtheminimalexpressionbysumming(multiplying)theproduct(sum)termsofallthegroups.

SomeofthepossiblecombinationsofmintermsinSOPform

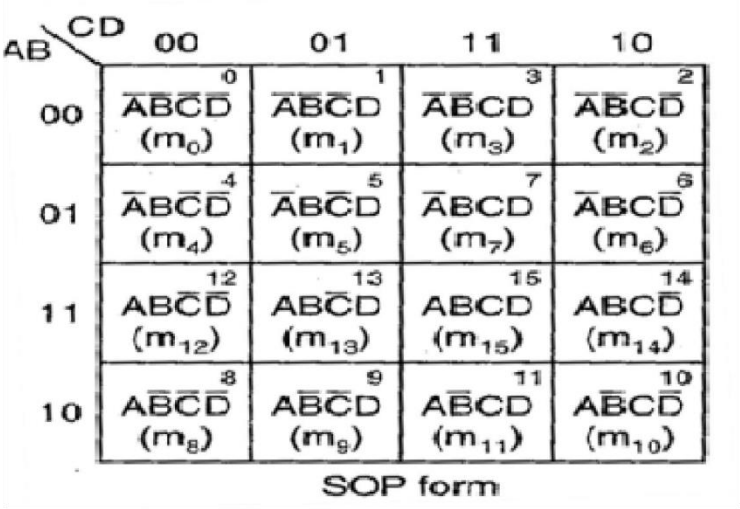


ThesepossiblecombinationsarealsoforPOSbut1sarereplacedby 0s.

FOURVARIABLEK-MAP:-

A four variable (A, B, C, D) expression can have 2^4 = 16 possible combinations of input variables. A four variable K-map has 2^4 = 16 squares or cells and each square on the map represents either a minterm or a maxterm as shown in the figure below. The binarynumber designations of the rows and columns are in the gray code. The binary numbers along the top of the map indicate the conditions of C and D along any column and binary numbers along left side indicate the conditions of A and B along any row. The numbers in the top right corners of the squares indicate the minterm or maxterm desginations.

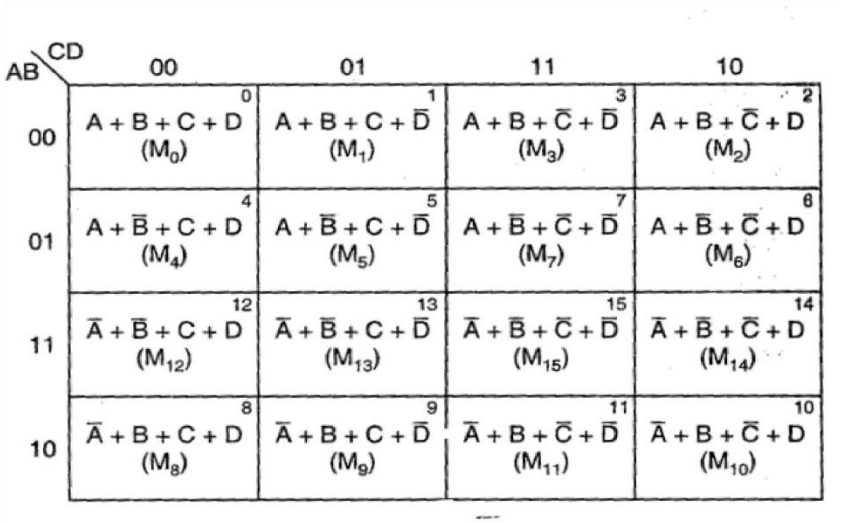
SOP FORM



AB \ CD	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$ (m_0) 0	$\bar{A}\bar{B}\bar{C}D$ (m_1) 1	$\bar{A}\bar{B}CD$ (m_3) 3	$\bar{A}\bar{B}C\bar{D}$ (m_2) 2
01	$\bar{A}B\bar{C}\bar{D}$ (m_4) 4	$\bar{A}B\bar{C}D$ (m_5) 5	$\bar{A}BCD$ (m_7) 7	$\bar{A}BC\bar{D}$ (m_6) 6
11	$AB\bar{C}\bar{D}$ (m_{12}) 12	$AB\bar{C}D$ (m_{13}) 13	$ABCD$ (m_{15}) 15	$ABC\bar{D}$ (m_{14}) 14
10	$A\bar{B}\bar{C}\bar{D}$ (m_8) 8	$A\bar{B}\bar{C}D$ (m_9) 9	$A\bar{B}CD$ (m_{11}) 11	$A\bar{B}C\bar{D}$ (m_{10}) 10

SOP form

POS FORM



AB \ CD	00	01	11	10
00	$A+B+C+D$ (M_0) 0	$A+B+C+\bar{D}$ (M_1) 1	$A+B+\bar{C}+\bar{D}$ (M_3) 3	$A+B+\bar{C}+D$ (M_2) 2
01	$A+\bar{B}+C+D$ (M_4) 4	$A+\bar{B}+C+\bar{D}$ (M_5) 5	$A+\bar{B}+\bar{C}+\bar{D}$ (M_7) 7	$A+\bar{B}+\bar{C}+D$ (M_6) 6
11	$\bar{A}+\bar{B}+C+D$ (M_{12}) 12	$\bar{A}+\bar{B}+C+\bar{D}$ (M_{13}) 13	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$ (M_{15}) 15	$\bar{A}+\bar{B}+\bar{C}+D$ (M_{14}) 14
10	$\bar{A}+B+C+D$ (M_8) 8	$\bar{A}+B+C+\bar{D}$ (M_9) 9	$\bar{A}+B+\bar{C}+\bar{D}$ (M_{11}) 11	$\bar{A}+B+\bar{C}+D$ (M_{10}) 10

MinimizationofSOPandPOSExpressions:-

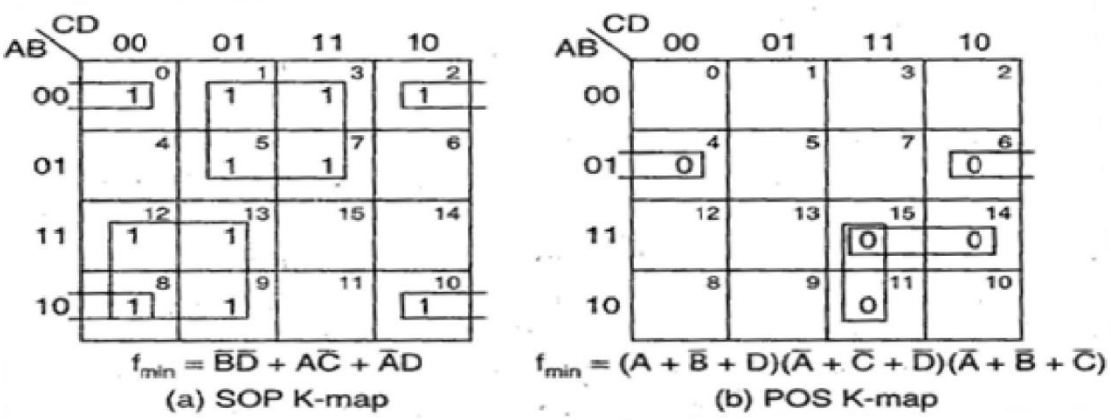
ForreducingtheBooleanexpressionsinSOP(POS)formthefollowingstepsaregivenbelow

- DrawtheK-mapandplace1s(0s)correspondingtotheminterms(maxterms)oftheSOP(POS) expression.
- In the map 1s(0s) which are not adjacent to anyother 1(0) are the isolated minterms (maxterms). They are to be read as they are because they cannot be combined even into a 2-square.
- Forthose 1s(0s)which areadjacent to onlyoneother1(0)make them pairs (2squares).
- Forquads(4-squares)andoctet(8squares)ofadjacent1s(0s)eveniftheycontainsome1s(0s) which have already been combined. They must geometrically form a square or a rectangle.
- Forany1s(0s)thathavenotbeencombinedyetthencombinethemintobiggersquaresifpossible.
- Formtheminimalexpressionbysumming(multiplying)theproduct(sum)termsofallthegroups.

Example:-

Reduceusingmappingtheexpressionf=∑m(0,1,2,3,5,7,8,9,10,12,13) Solution:-

ThegivenexpressioninPOSformisf=**Π**M(4,6,11,14,15)andinSOPformf=∑m(0,1,2,3,5,7,8,9, 10,12,13)



TheminimalSOP expression is$f_{min}=B\overline{D}+\overline{A}C+\overline{A}\overline{D}$

TheminimalPOSexpressionis$f_{min}=(A+B+D)(\overline{A}+C+D)(\overline{A}+\overline{B}+C)$

DON'TCARECOMBINATIONS:-

The combinationsfor which the values of the expression are not specifiedare called don't care combinations or optional combinations and such expression stand incompletely specified. The output is a don't care for these invalid combinations. The don't care terms are denoted by d or X. During the process of designing using SOP maps, each don't care is treated as 1 to reduce the map otherwise it is treated as 0 and left alone. During the process of designing using POS maps, each don't care is treated as 0 to reducethe map otherwise it is treated as 1 and left alone.

A standard SOP expression with don't cares can be converted into standard POS form by keeping the don't cares as they are, and the missing minterms of the SOP form are written as the maxterms of the POS form. Similarly, to convert a standard POS expression with don't cares can be converted into standard SOP form by keeping the don't cares as they are, and the missing maxterms of the POS form are written as the minterms of the SOP form.

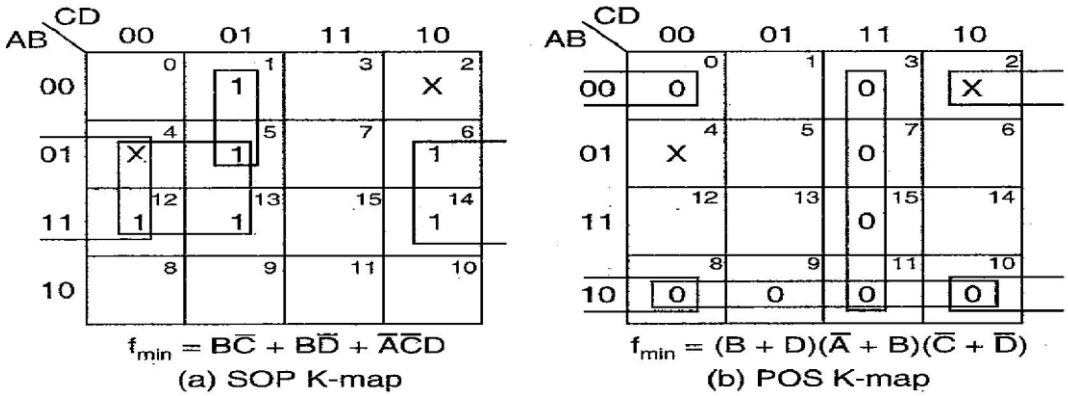
Example:-

Reducetheexpressionf=∑m(1,5,6,12,13,14)+d(2,4)usingK-map.

Solution:-

Thegiven expressioninSOPformisf=∑m(1,5,6,12, 13,14)+d(2,4)

Thegivenexpressionin POSformisf= ΠM(0,3,7,8,9,10,11,15)+ d(2,4)



Theminimalof SOPexpressionis$f_{min}=BC+B\overline{D}+A\overline{C}\overline{D}$

The minimalof POS expression isf_{min}= (B+ D)(A +$\overline{B}$) (C+ $\overline{D}$)

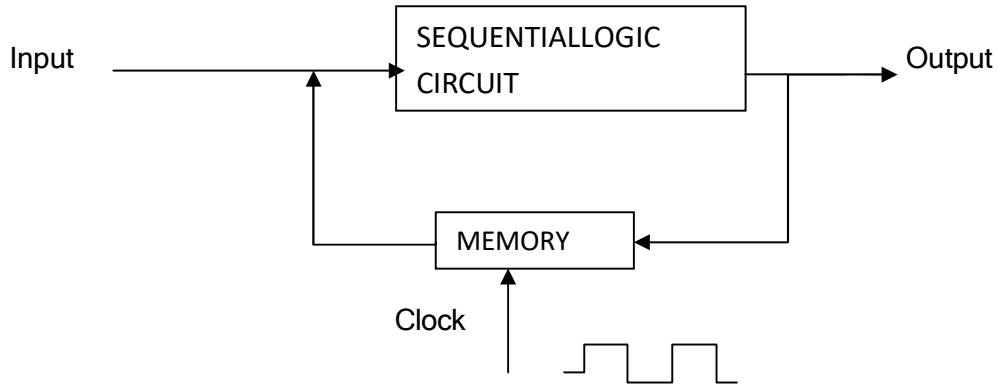
SEQUENTIALLOGIC CIRCUIT

SEQUENTIALCIRCUIT:-

- It is a circuit whose output depends upon the present input, previous output and the sequence in which the inputs are applied.

HOWTHESEQUENTIALCIRCUITISDIFFERENTFROMCOMBINATIONALCIRCUIT?:-

- In combinational circuit output depends upon present input at any instant of time and do not use memory. Hence previous input does not have any effect on the circuit. But sequential circuit hasmemory and depends upon present input and previous output.
- Sequential circuits are slower than combinational circuits and these sequential circuits are harder to design.



[BlockdiagramofSequentialLogic Circuit]

- The data stored by the memory element at any given instant of time is called the present state of sequential circuit.

TYPES:-

Sequentiallogiccircuits(SLC)areclassifiedas

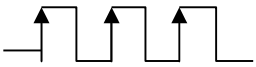
(i) SynchronousSLC
(ii) AsynchronousSLC

- The SLC that are controlled by clock are called synchronous SLC and those which are not controlled by a clock are asynchronous SLC.
- Clock:-Arecurringpulseiscalleda clock.

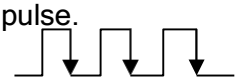
FLIP-FLOPAND LATCH:-

- Aflip-floporlatchisacircuitthathastwostablestatesandcanbeusedtostoreinformation.
- Aflip-flopisabinarystoragedevicecapableofstoringonebitofinformation.Inastablestate,the output of a flip-flop is either 0 or 1.
- Latchisanon-clockedflip-flopanditisthebuildingblockfortheflip-flop.
- Astorageelementindigitalcircuitcanmaintainabinarystateindefinitelyuntildirectedbyaninput signal to switch state.
- Storage element that operate with signal level are called latches andthose operate with clock transition are called as flip-flops.

- The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs.
- Aflip-flopiscalledsobecauseitsoutputeitherflipsorflopsmeaningtoswitchbackand forth.
- Aflip-flopisalsocalledabi-stablemulti-vibratorasithastwostablestates.Theinputsignalswhich command the flip-flop to change state are called excitations.
- Flip-flopsarestoragedevicesandcanstore1or0.
- Flip-flops using the clock signal are called clocked flip-flops. Control signals are effectiveonly if theyare applied in synchronization with the clock signal.
- Clock-signalsmaybepositive-edgetriggeredornegative-edgetriggered.
- Positive-edge triggered flip-flops are those in which state transitions take place only at positive-goingedge of the clock pulse.



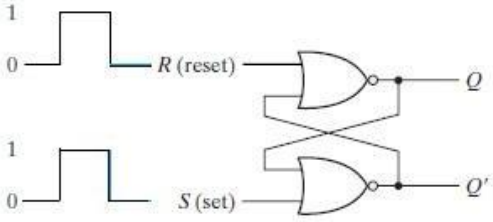
- Negative-edge triggered flip-flops are those in which state transition take place only at negative- going edge of the clock pulse.



- Somecommontypeofflip-flopsinclude
  a) SR(set-reset)F-F
  b) D(data ordelay)F-F
  c) T (toggle) F-F and
  d) JKF-F

SRlatch:-

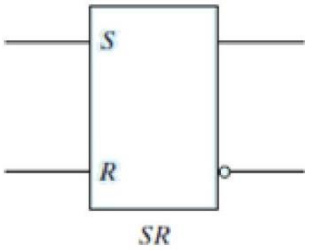
- TheSRlatchisacircuitwithtwocross-coupledNORgatesortwocross-coupledNANDgates.
- IthastwooutputslabeledQandQ'.Two inputsaretherelabeledSforsetandRfoereset.
- The latch has two useful states. When Q=0 and Q'=1 the condition is called reset state and when Q=1 and Q'=0 the condition is called set state.
- NormallyQandQ' arecomplement ofeach other.
- The figure represents a SR latch with two cross-coupled NOR gates. The circuit has NOR gates and as we know if any one of the input for a NOR gate is HIGH then its output will be LOWand if both the inputs are LOW then only the output will be HIGH.



- Under normal conditions, both inputs of the latch remain at 0 unless the state has to be changed. The application of a momentary 1 to the S input causes the latch to go to the set state. The S input must go back to 0 before any other changes take place, in order to avoid the occurrence of an undefined next state that results from the forbidden input condition.
- The firstcondition (S=1,R =0)istheaction thatmustbetakenbyinput Stobringthecircuit tothe set state. Removingthe active input from S leaves the circuit in the same state. After both inputs return to0, it is then possible to shift to the reset state by momentary applying a 1 to the R input. The 1 can then be removed from R, whereupon the circuit remains in the reset state. When both inputs S and R are equal to 0, the latch can be in either the set or the reset state, depending on which input was most recently a 1.

- If a 1 is applied to both the S and R inputs of the latch, both outputs go to 0. This action produces an undefined next state, because the state that results from the input transitions depends on the order in which they return to 0. It also violates the requirement that outputs be the complement of each other. In normal operation, this condition is avoided by making sure that 1's are not applied to both inputs simultaneously.
- TruthtableforSRlatchdesignedwithNORgatesisshownbelow.

Input		Output				Comment
S	R	Q	Q'	Q_{Next}	Q'_{Next}	
0	0	0	1	0	1	Nochange
0	0	1	0	1	0	
0	1	0	1	0	1	Reset
0	1	1	0	0	1	
1	0	0	1	1	0	Set
1	0	1	0	1	0	
1	1	0	1	X	X	Prohibited state
1	1	1	0	X	X	



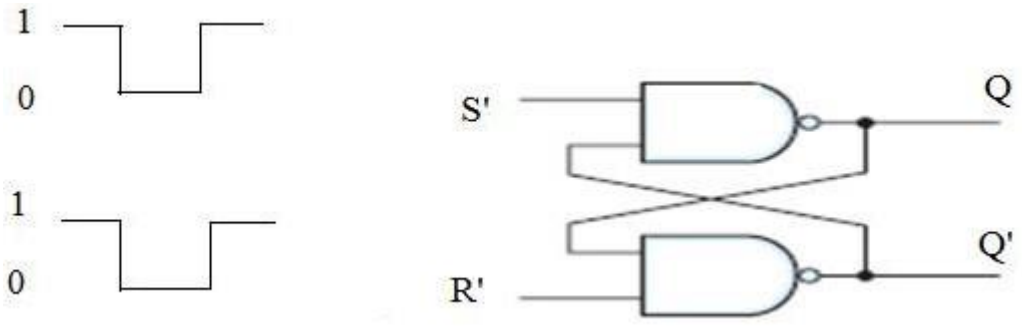
Symbolfor SRNOR Latch

Racing Condition:-

IncaseofaSRlatchwhenS=R=1inputisgivenboththeoutputwilltrytobecome0.Thisiscalled Racing condition.

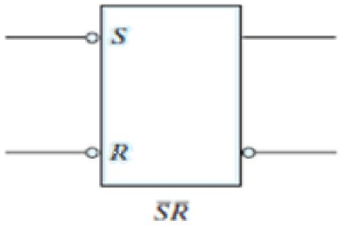
SRlatchusingNANDgate:-

- The below figure represents a SR latch with two cross-coupled NAND gates. The circuit has NAND gates and as we know ifany one oftheinput for a NAND gate is LOWthen its outputwill beHIGH andif both the inputs are HIGH then only the output will be LOW.
- It operates with both inputs normally at 1, unless the state of the latch has to be changed. The application of 0 to the S input causes output Q to go to 1, putting the latch in the set state. When the S input goes back to 1, the circuit remains in the set state. After both inputs go back to 1, we are allowedto change the state of the latch by placing a 0 in the R input. This action causes the circuit to go to the reset state and stay there even after both inputs return to 1.



- The condition that is forbidden for the NAND latch is both inputs being equal to 0 at the same time, an input combination that should be avoided.

Incomparingthe NAND withtheNOR latch,note that theinput signals for theNAND require the complement of those values used for the NOR latch. Because the NAND latch requires a 0 signal to change its state, it is sometimes referred toasan S'R' latch. The primes (or,sometimes, barsovertheletters) designate the fact that the inputs must be in their complement form to activate the circuit.



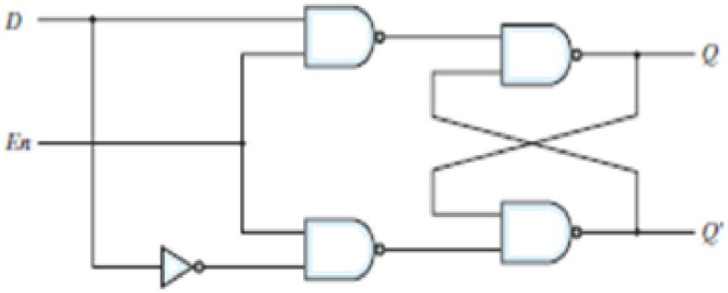
SR

The above representsthe symbolfor invertedSR latch or SRlatchusing NAND gate. Truth table for SR latch using NAND gate or Inverted SR latch

S	R	Q_{next}	Q'_{next}
0	0	Race	Race
0	1	0	1(Reset)
1	0	1	0(Set)
1	1	Q(No change)	Q'(No change)

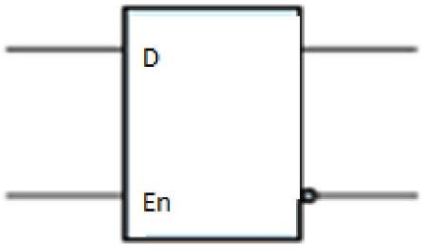
DLATCH:-

- One way to eliminate the undesirable condition ofthe indeterminate state in the SR latch is to ensurethat inputs S and R are never equal to 1 at the same time.



(a) Logic diagram

- This isdoneintheD latch.This latch hasonlytwoinputs: D(data) andEn(enable).
- TheD input goesdirectlyto theS input,and itscomplement is appliedto theR input.



D

(SymbolforD-Latch)

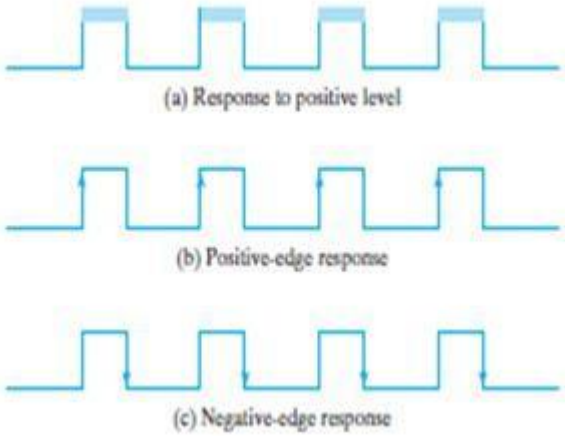
- Aslongastheenableinputisat0,thecross-coupledSRlatchhasbothinputsatthe1levelandthe circuit can't change state regardless of the value of D.
- ThebelowrepresentsthetruthtablefortheD-latch.

En	D	NextStateofQ
0	X	No change
1	0	Q=0;ResetState
1	1	Q=1;Set State

- The D input is sampled when En = 1. If D = 1, the Q output goes to 1, placing the circuit in the set state. If D = 0, output Q goes to 0, placing the circuit in the reset state. This situation provides a path frominput D to the output, and for this reason, the circuit is often called a TRANSPARENT latch.

TRIGGERINGMETHODS:-

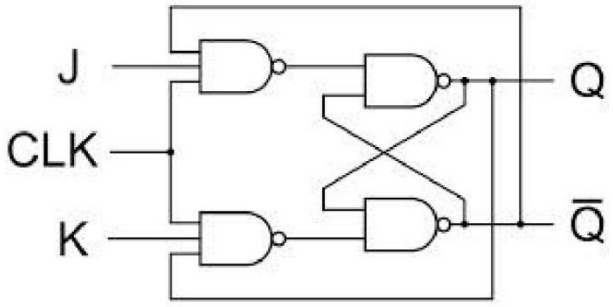
- The state of a latch or flip-flop is switched by a change in the control input. This momentary change iscalled a trigger, and the transition it causes is said to trigger the flip-flop.
- Flip-flop circuits are constructed in such a way as to make them operate properly when they are part ofa sequential circuit that employs a common clock.
- Theproblemwiththelatchisthatitrespondstoachangeinthelevelofaclockpulse.Forproper operation of a flip-flop it should be triggered only during a signal transition.
- Thiscanbeaccomplishedbyeliminatingthefeedbackpaththatisinherentintheoperationofthe sequentialcircuitusinglatches.Aclockpulsegoesthroughtwotransitions:from0to1andthereturn from1to 0.
- A ways that a latch can be modified to form a flip-flop is to produce a flip-flop that triggers only during a signal transition(from 0to 1 or from 1 to 0) of thesynchronizing signal (clock) and is disabled during the rest of the clock pulse.



JKFLIP-FLOP:-

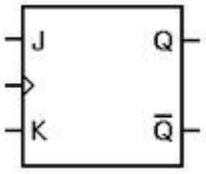
- The JK flip-flop can be constructed by using basic SR latch and a clock. In this case the outputs Q and Q' are returned back and connected to the inputs of NAND gates.
- ThissimpleJKflipFlopisthemostwidelyusedofalltheflip-flopdesignsandisconsideredtobea universal flip-flop circuit.
- The sequential operation of the JK flip flop is exactly the same as for the previous SR flip-flop with the same "Set" and "Reset" inputs.
- The difference this time is that the "JK flip flop" has no invalid or forbidden input states of the SR Latch even when S and R are both at logic "1".

(ThebelowdiagramshowsthecircuitdiagramofaJKflip-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".
- Due to this additional clocked input, a JK flip-flop has four possible input combinations, "logic 1", "logic 0", "no change" and "toggle".

- Thesymbolfor aJKflipflopissimilartothatofanSRbistable latchexceptttheclockinput.



(TheabovediagramshowsthesymbolofaJKflip-flop.)

- Both the S and the R inputs of the SR bi-stable have now been replaced by two inputs called the J andK inputs, respectively after its inventor Jack and Kilby. Then this equates to: J = S and K = R.
- The two 2-input NAND gates of the gated SR bi-stable have now been replaced by two 3-input NANDgates with the third input of each gate connected to the outputs at Q and Q'.
- This cross coupling of the SR flip-flop allows the previously invalid condition of S = "1" and R = "1" state to be used to produce a "toggle action" as the two inputs are now interlocked.
- If the circuit is now "SET" the J input is inhibited by the "0" status of Q' through the lower NAND gate. If the circuit is "RESET" the K input is inhibited by the "0" status of Q through the upper NAND gate. As Q andQ'arealwaysdifferentwecanusethemtocontroltheinput.

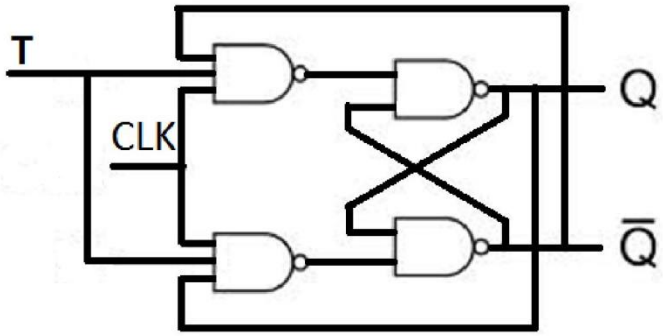
(TruthtableforJKflip-flop)

Input		Output		Comment
J	K	Q	Q_{next}	
0	0	0	0	Nochange
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	1	Toggle
1	1	1	0	

- Whenboth inputs J and Kareequaltologic "1", theJK flipfloptoggles.

TFLIP-FLOP:-

- Toggleflip-floporcommonlyknownasTflip-flop.
- Thisflip-flophasthesimilaroperationas thatoftheJK flip-flop withboth theinputsJandKareshorted i.e.botharegiventhecommoninput.



- HenceitstruthtableissameasthatofJKflip-flopwhenJ=K=0andJ=K=1.Soitstruthtableisas follows.

T	Q	Q_{next}	Comment
0	0	0	Nochange
	1	1	
1	0	1	Toggles
	1	0	

CHARACTERISTICTABLE:-

- Acharacteristictabledefinesthelogicalpropertiesofaflip-flopbydescribingitsoperationintabular form.
- Thenextstateisdefinedasafunctionof theinputsandthepresentstate.
- Q (t)refersto thepresentstateand Q(t+1) isthenext.
- Thus, Q (t) denotesthe state of theflip-flop immediatelybefore theclock edge, and Q(t + 1) denotes the state that results from the clock transition.
- ThecharacteristictablefortheJKflip-flopshowsthatthenextstateisequaltothepresentstatewhen inputs J and K are both equal to 0. This condition can be expressed as Q (t + 1) = Q (t), indicating that the clock produces no change of state.

CharacteristicTableOfJKFlip-Flop

J	K	Q(t+1)
0	0	Q(t) No change
0	1	0 Reset
1	0	1 Set
1	1	Q'(t)Complement

- When K = 1 and J = 0, the clock resets the flip-flop and Q(t + 1) = 0. With J = 1 and K = 0, the flip-flop sets and Q(t + 1) = 1. When both J and K are equal to 1, the next state changes to the complement of the present state, a transition that can be expressed as Q(t + 1) = Q'(t).
- ThecharacteristicequationforJKflip-flopisrepresentedas

$$Q(t+1)=JQ'+K'Q$$

CharacteristicTableofDFlip-Flop

D	Q(t+1)
0	0
1	1

- ThenextstateofaDflip-flopisdependentonlyontheDinputand isindependentofthepresent state.
- This can be expressed as Q (t + 1) = D. It means that the next-state value is equal to the value of D. Note that the D flip-flop does not have a "no-change" condition and its characteristic equation is written as Q(t+1)=D.

CharacteristicTableofTFlip-Flop

T	Q(t+1)
0	Q(t) Nochange
1	Q'(t) Complement

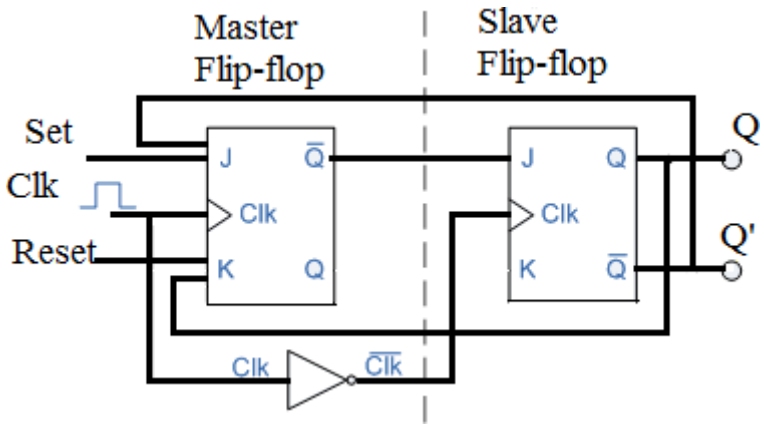
- The characteristic table of T flip-flop has only two conditions: When T = 0, the clock edge does not change the state; when T = 1, the clock edge complements the state of the flip-flop and the characteristic equation is

$$Q(t+1) = T \oplus Q = T'Q + TQ'$$

MASTER-SLAVEJKFLIP-FLOP:-

- TheMaster-SlaveFlip-Flopisbasicallytwogated SRflip-flopsconnectedtogetherinaseries configuration with the slave having an inverted clock pulse.
- The outputs from Q and Q' from the "Slave" flip-flop are fed back to the inputs of the "Master"with theoutputs of the "Master" flip flop being connected to the two inputs of the "Slave" flip flop.
- This feedback configuration from the slave's output to the master's input gives the characteristic toggle of the JK flip flop as shown below.

TheMaster-SlaveJKFlipFlop



- TheinputsignalsJandKareconnectedtothegated"master"SRflipflopwhich"locks"theinput condition while the clock (Clk) input is "HIGH" at logic level "1".
- Astheclockinputofthe"slave"flipflopistheinverse(complement)ofthe"master"clockinput,the "slave" SR flip flop does not toggle.
- The outputs from the "master" flipflop are only"seen" bythe gated "slave"flip flop when the clock input goes "LOW" to logic level "0".
- Whentheclockis"LOW",theoutputsfromthe"master"flipfloparelatchedandanyadditionalchanges to its inputs are ignored.
- Thegated"slave"flipflop nowrespondsto thestateof its inputspassedover bythe "master" section.
- Then on the "Low-to-High" transition of the clock pulse the inputs of the "master" flip flop are fedthrough to the gated inputs of the "slave" flip flop and on the "High-to-Low" transition the same inputs are reflected on the output of the "slave" making this type of flip flop edge or pulse-triggered.
- Then, the circuit accepts input data when the clock signal is "HIGH", and passes the data to the output on the falling-edge of the clock signal.
- In other words, the Master-Slave JK Flip flop is a "Synchronous" device as it only passes data with the timing of the clock signal.

FLIP-FLOPCONVERSIONS:-

SRFlipFloptoJKFlipFlop

For this J and K will be given as external inputs to S and R. As shown in the logic diagram below, S and R will be the outputs of the combinational circuit.

The truth tables for the flip flop conversion are given below. The present state is represented by Qp and Qp+1is the next state to be obtained when the J and K inputs are applied.

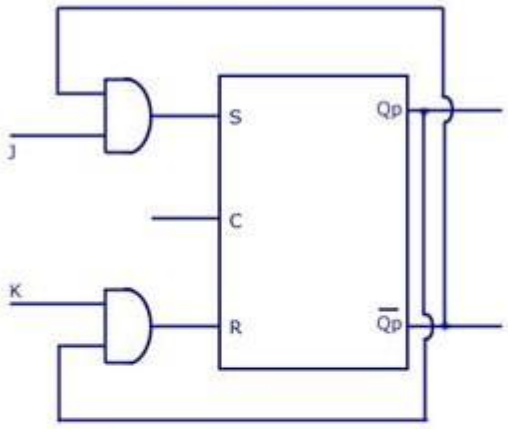
For two inputs J and K, there will be eight possible combinations. For each combination of J, K and Qp, the corresponding Qp+1 states are found.Qp+1 simply suggests the future values to be obtained bythe JK flipflop after the value of Qp. The table is then completed by writing the values of S and R required to get each Qp+1 from the corresponding Qp. That is, the values of S and R that are required to change the state of the flip flop from Qp to Qp+1 are written.

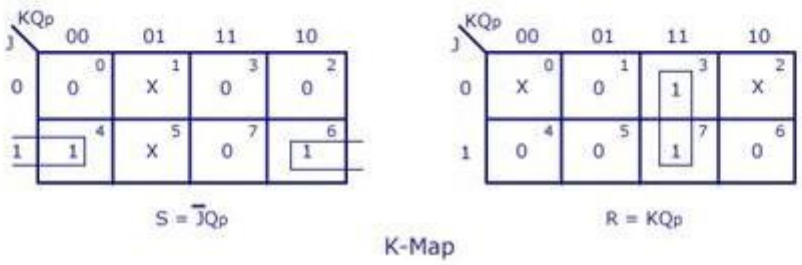
S-R Flip Flop to J-K Flip Flop

Conversion Table

J-K Inputs		Outputs		S-R Inputs	
J	K	Qp	Qp+1	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Logic Diagram





K-Map

JKFlipFloptoSRFlipFlop

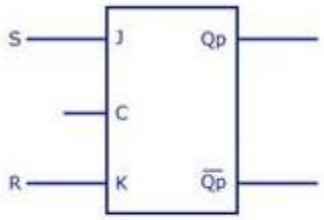
- This will be the reverse process of the above explained conversion. S and R will be the external inputsto J and K. J and K will be the outputs of the combinational circuit. Thus, the values of J and K have to be obtained in terms of S, R and Qp.
- Aconversiontable isto bewritten usingS, R,Qp,Qp+1, JandK.
- For two inputs, S and R, eight combinations are made. For each combination, the corresponding Qp+1 outputs are found out.
- The outputs for the combinations of S=1 and R=1 are not permitted for an SR flip flop. Thus the outputs are considered invalid and the J and K values are taken as “don’t cares”.

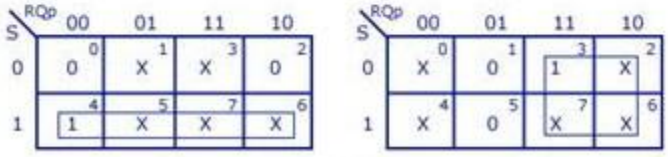
J-K Flip Flop to S-R Flip Flop

Conversion Table

S-R Inputs S	R	Outputs Qp	Qp+1	J-K Inputs J	K
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	X	1
1	0	0	1	1	X
1	0	1	1	X	0
1	1	Invalid		Dont care	
1	1	Invalid		Dont care	

Logic Diagram





SRFlipFloptoDFlipFlop

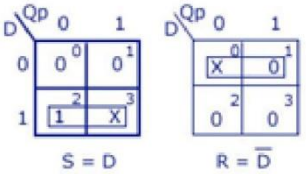
- SandRaretheactualinputsoftheflipflopandDistheexternalinputoftheflip flop.
- Thefourcombinations,thelogicdiagram,conversiontable,andtheK-mapforSandRintermsofD and Qp are shown below.

S-R Flip Flop to D Flip Flop

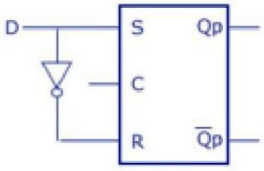
Conversion Table

D Input	Outputs Qp	Qp+1	S-R Inputs S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

K-maps



Logic Diagram



DFlipFlopto SRFlipFlop

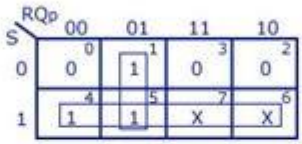
- D is the actual input of the flipflop and S and R are the external inputs. Eight possible combinations are achieved from the external inputs S, R and Qp.
- But,sincethecombinationofS=1andR=1areinvalid,thevaluesofQp+1andDareconsideredas “don’t cares”.
- The logic diagram showing the conversion from D to SR, and the K-map for D in terms of S, R and Qp are shown below.

D Flip Flop to S-R Flip Flop

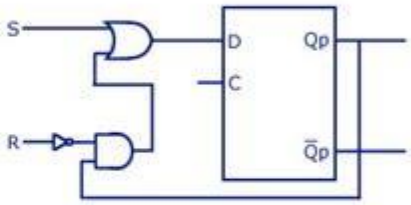
Conversion Table

S-R Inputs S	R	Outputs Qp	Qp+1	D Input
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	Invalid		Don't care
1	1	Invalid		Don't care

K-map



Logic Diagram



JKFlipFloptoTFlipFlop:-

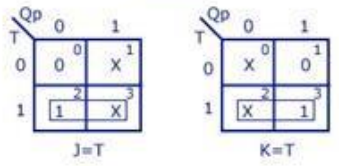
- JandKaretheactualinputsof theflipflopandTistakenastheexternalinputfor conversion
- FourcombinationsareproducedwithTandQp.JandKareexpressedintermsofTand Qp.
- Theconversiontable,K-maps,andthelogicdiagramaregiven below.

J-K Flip Flop to T Flip Flop

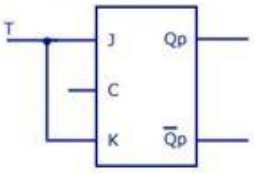
Conversion Table

T Input	Outputs Qp	Qp+1	J-K Inputs J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

K-maps



Logic Diagram



DFlipFloptoJKFlip Flop:-

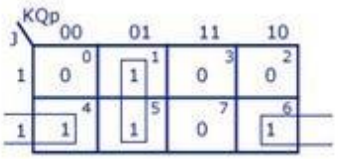
- Inthisconversion,Distheactualinputtotheflipflopand JandKaretheexternalinputs.
- J,KandQpmakeeightpossiblecombinations,asshownintheconversiontablebelow.Dis expressed in terms of J, K and Qp.
- Theconversiontable,theK-mapforDintermsofJ,KandQpandthelogicdiagramshowingthe conversion from D to JK are given in the figure below.

D Flip Flop to J-K Flip Flop

Conversion Table

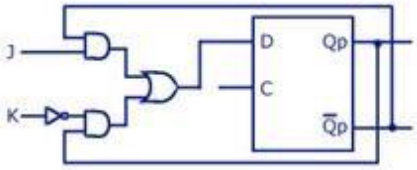
J-K Input J	K	Outputs Qp	Qp+1	D Input
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

K-map



$D = J\overline{Q}p + \overline{K}Qp$

Logic Diagram



JKFlipFloptoDFlip Flop:-

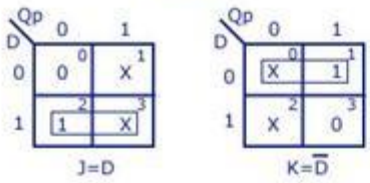
- DistheexternalinputandJandKaretheactualinputsoftheflipflop.DandQpmakefour combinations. J and K are expressed in terms of D and Qp.
- Thefourcombinationconversiontable,theK-mapsforJandKintermsofD andQp.

J-K Flip Flop to D Flip Flop

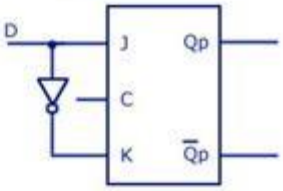
Conversion Table

D Input	Outputs Qp	Qp+1	J-K Inputs J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	0	X	0

K-maps

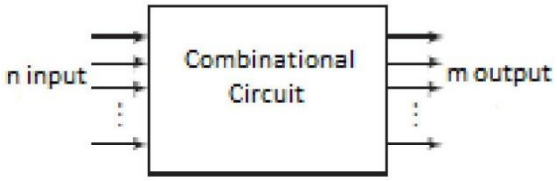


Logic Diagram



COMBINATIONALLOGIC CIRCUIT

- A combinational circuit consists of logic gates whose outputs at anytime are determinedfrom onlythe present combination of inputs.
- Acombinationalcircuitperformsanoperationthatcanbespecifiedlogicallybyasetof Boolean functions.
- Itconsistsofaninterconnectionoflogicgates.Combinationallogicgatesreacttothevaluesofthe signals at their inputs and produce the value of the output signal, transforming binary information from the given input data to a required output data.
- Ablockdiagramofacombinationalcircuitisshowninthebelowfigure.
- The n input binary variables come from an external source; the m output variables are produced by the internal combinational logic circuit and go to an external destination.
- Each input and output variable exists physically as an analog signal whose values are interpreted to be a binary signal that represents logic 1and logic 0.



BINARYADDER-SUBTRACTOR**:-**

- Digital computers perform a variety of information-processing tasks. Among the functions encountered are the various arithmetic operations.
- The most basic arithmetic operation is the addition of two binary digits. This simple addition consists of four possible elementary operations: 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, and 1 + 1 = 10.
- The first three operations produce a sum of one digit, but when both augend and addend bits are equal to 1; the binary sum consists of two digits. The higher significant bit of this result is called a carry.
- Whentheaugendandaddendnumberscontainmoresignificantdigits,thecarryobtainedfromthe addition of two bits is added to the next higher order pair of significant bits.
- Acombinationalcircuitthatperformstheadditionof twobitsiscalledahalfadder.
- One that performsthe addition of threebits (two significant bits and aprevious carry) is afulladder.The names of thecircuits stem from the fact that two half adders canbe employed to implement afulladder.

HALF ADDER:-

- Thiscircuitneedstwobinaryinputs andtwobinary outputs.
- The input variables designate the augend and addend bits; the output variables produce the sum andcarry. Symbols x and y are assigned to the two inputs and S (for sum) and C (for carry) to the outputs.
- Thetruthtableforthehalfadderislistedinthebelowtable.
- TheCoutputis1only whenbothinputsare1.TheSoutputrepresentstheleastsignificantbitofthe sum.
- ThesimplifiedBooleanfunctionsforthetwooutputscanbeobtaineddirectlyfromthetruth table.

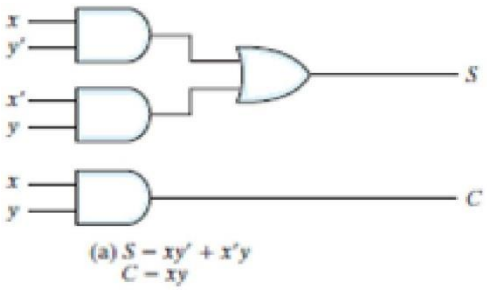
x	y	D	B
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table

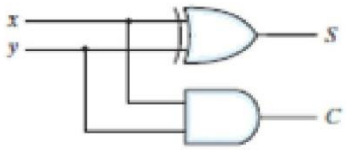
- Thesimplifiedsum-of-productsexpressionsare

$$S=x'y+xy' \quad C = xy$$

- The logic diagram of the half adder implemented in sum of products is shown in the below figure. It can be also implemented with an exclusive-OR and an AND gate.



(a) $S = xy' + x'y$
$C = xy$



(b) $S = x \oplus y$
$C = xy$

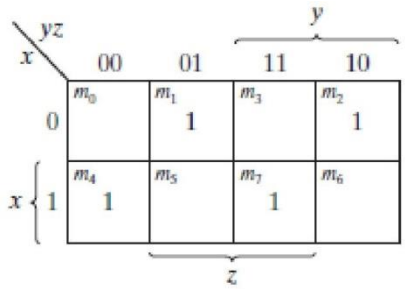
FULLADDER:-

- Afulladderisacombinationalcircuitthatformsthearithmeticsumofthree bits.
- It consists of three inputs and two outputs. Two of the input variables, denoted by x and y , represent the two significant bits to be added. The third input, z , represents the carry from the previous lower significant position.

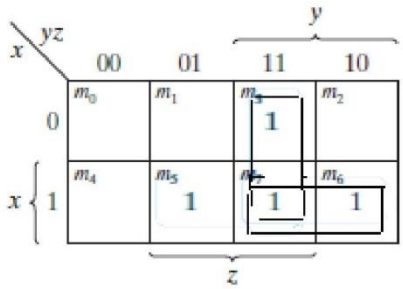
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Truth Table

- Two outputs are necessary because the arithmetic sum of three binary digits ranges in value from 0 to3, and binary representation of 2 or 3 needs two bits. The two outputs are designated by the symbols S for sum and C for carry.



(a) $S = x'y'z + x'yz' + xy'z' + xyz$



(b) $C = xy + xz + yz$

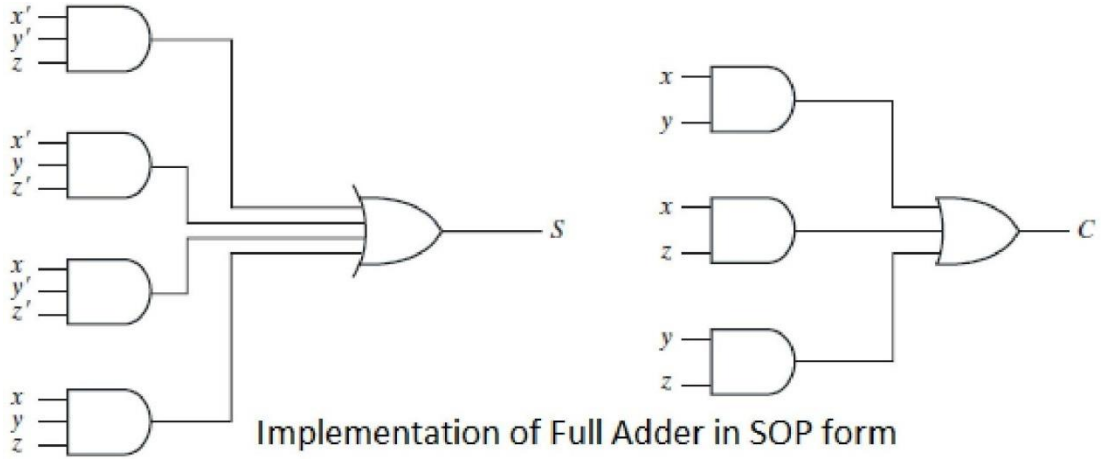
K-Map for full adder

- The binary variable S gives the value of the least significant bit of the sum. The binary variable C gives the output carry formed by adding the input carry and the bits of the words.
- Theeightrowsundertheinput variablesdesignateallpossible combinationsof thethreevariables.The output variables are determined from the arithmetic sum of the input bits. When all input bits are 0, the output is 0.
- The S output is equal to 1 when only one input is equal to 1 or when all three inputs are equal to 1. The C output has a carry of 1 if two or three inputs are equal to 1.
- Thesimplifiedexpressionsare

S= x'y'z + x'yz'+xy'z' +xyz

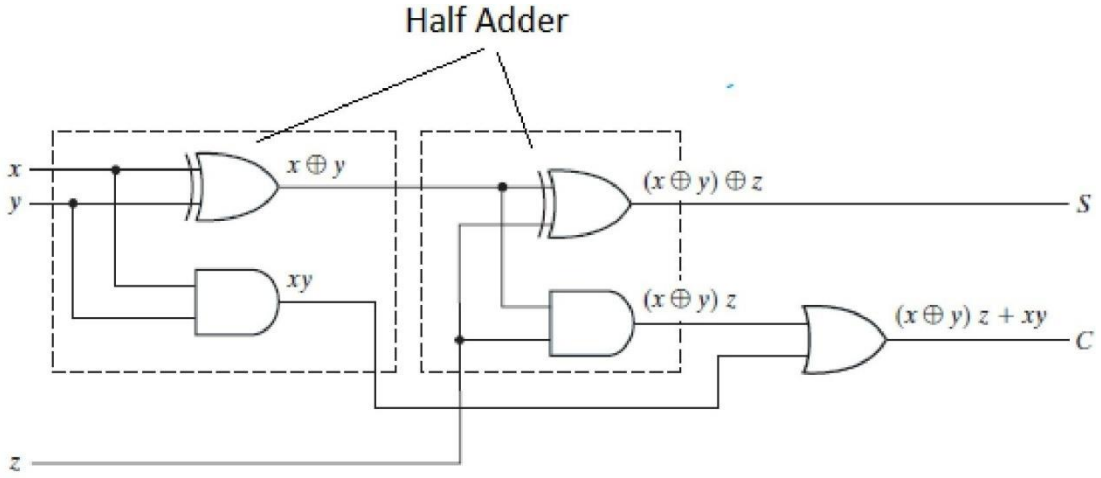
C=xy+xz+yz

- Thelogicdiagramforthefulladderimplementedinsum-of-productsformisshowninfigure.



Implementation of Full Adder in SOP form

- ItcanalsobeimplementedwithtwohalfaddersandoneORgateasshowninthefigure.



Implementation of Full Adder using Two Half Adders and an OR gate

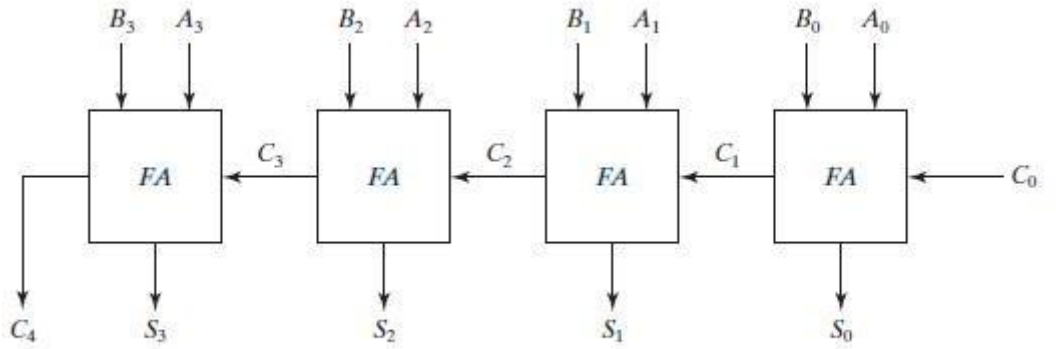
- Afulladderisacombinationalcircuitthatformsthearithmeticsumofthree bits.

BINARYADDER:-

- Abinaryadderis adigitalcircuitthatproducesthearithmeticsumof twobinarynumbers.
- It can be constructed with full adders connected in cascade, with the output carry from each full adderconnected to the input carry of the next full adder in the chain.
- Additionofn-bitnumbersrequiresachainofnfulladdersorachainofone-halfadderandn-1full adders. In the former case, the input carry to the least significant position is fixed at 0.
- Theinterconnectionoffourfull-adder(FA)circuitstoprovideafour-bitbinaryripplecarryadderis shown in the figure.
- TheaugendbitsofA and theaddend bitsofB aredesignatedby subscriptnumbers fromrightto left, with subscript 0 denoting the least significant bit.
- The carries are connected in a chain through the full adders. The input carry to the adder is C0, and it ripples through the full adders to the output carry C4. The S outputs generate the required sum bits.
- Ann-bitadderrequiresnfulladders,witheachoutputcarryconnectedtotheinputcarryofthenext higher order full adder.
- Consider the two binary numbers A = 1011 and B = 0011. Their sum S = 1110 is formed with the four-bit adder as follows:

Subscript *i*:	3	2	1	0	
Input carry	0	1	1	0	C_i
Augend	1	0	1	1	A_i
Addend	0	0	1	1	B_i
Sum	1	1	1	0	S_i
Output carry	0	0	1	1	C_{i+1}

- The bits are added with full adders, starting from the least significant position (subscript 0), to form the sum bit and carry bit. The input carry C_0 in the least significant position must be 0.
- The value of C_{i+1} in a given significant position is the output carry of the full adder. This value is transferred into the input carry of the full adder that adds the bits one higher significant position to the left.
- The sum bits are thus generated starting from the rightmost position and are available as soon as the corresponding previous carry bit is generated. All the carries must be generated for the correct sum bits to appear at the outputs.



Four Bit Binary Adder

HALF SUBTRACTOR:-

- This circuit needs two binary inputs and two binary outputs.
- Symbols x and y are assigned to the two inputs and D (for difference) and B (for borrow) to the outputs.
- The truth table for the half subtractor is listed in the below table.

x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

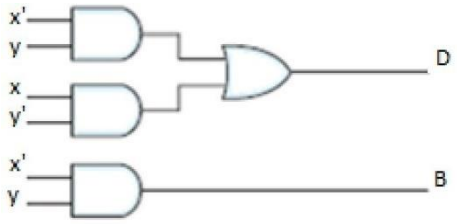
Truth Table

- The B output is 1 only when the inputs are 0 and 1. The D output represents the least significant bit of the subtraction.
- The subtraction operation is done by using the following rules as

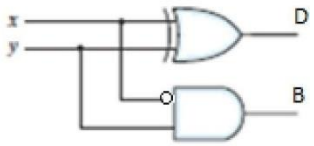
0-0=0;
0-1=1 with borrow 1;
1-0=1;
1-1=0.

- The simplified Boolean functions for the two outputs can be obtained directly from the truth table. The simplified sum-of-products expressions are

D=x'y+xy' and B=x'y



D = x'y+xy'
B = x'y



D = x ⊕ y
B=x'y

- The logic diagram of the half adder implemented in sum of products is shown in the figure. It can bealso implemented with an exclusive-OR and an AND gate with one inverted input.

FULLSUBTRACTOR:-

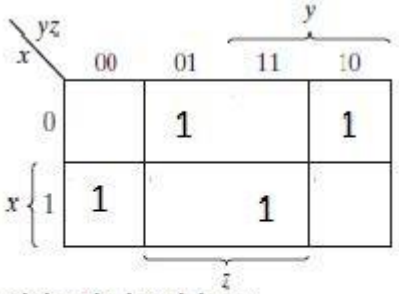
- Afullsubtractorisacombinationalcircuitthatformsthearithmeticsubtractionoperationofthreebits.
- It consists of three inputs and two outputs. Two of the input variables, denoted by x and y , represent the two significant bits to be subtracted. The third input, z , is subtracted from the result 0f the first subtraction.

x	y	z	D	B
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

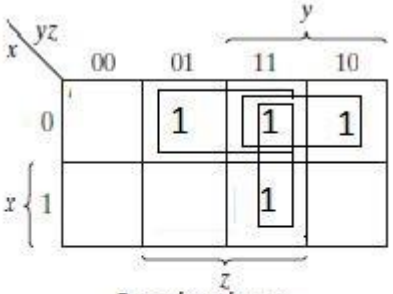
Truth Table

- Two outputs are necessary because the arithmetic subtraction of three binary digits ranges in valuefrom 0 to 3, and binary representation of 2 or 3 needs two bits. The two outputs are designated by the symbols D for difference and B for borrow.

- The binary variable D gives the value of the least significant bit of the difference. The binary variable B gives the output borrow formed during the subtraction process.



D = x'y'z+x'yz'+xy'z'+xyz



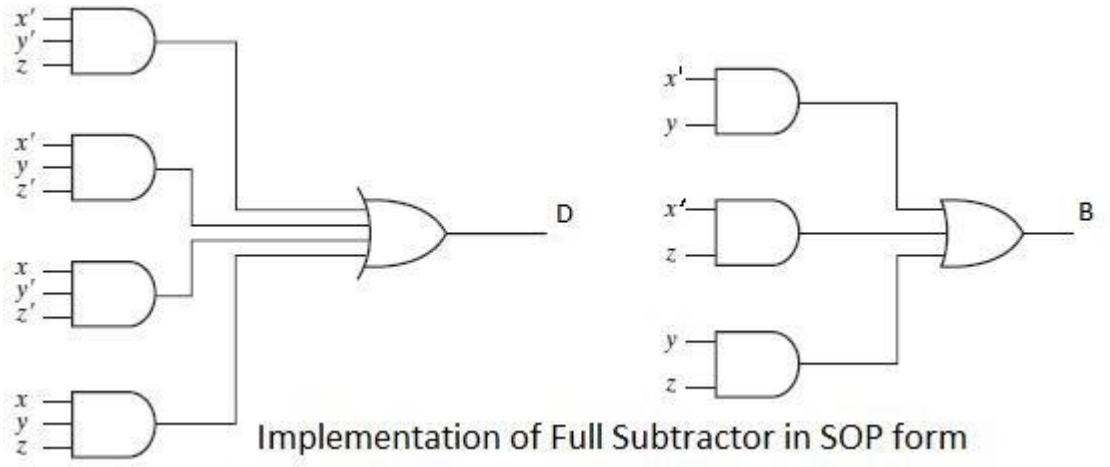
B = x'z+x'y+yz

K-Map for full Subtractor

- The eight rows under the input variables designate all possible combinations of the three variables. The output variables are determined from the arithmetic subtraction of the input bits.
- ThedifferenceDbecomes1whenanyoneoftheinputis1orallthreeinputsareequalto1andthe borrow B is 1 when the input combination is (0 0 1) or (0 1 0) or (0 1 1) or (1 1 1).
- Thesimplifiedexpressionsare

D=x'y'z+x'yz'+ xy'z'+xyz B =
x'z + x'y + yz

- Thelogicdiagramforthefulladderimplementedinsum-of-productsformisshowninfigure.



Implementation of Full Subtractor in SOP form

MAGNITUDECOMPARATOR:-

- A magnitude comparator is a combinational circuit that compares two numbers A and B and determines their relative magnitudes.
- Thefollowingdescriptionisabouta2-bitmagnitudecomparatorcircuit.
- The outcome of the comparison is specified by three binary variables that indicate whether A < B, A =B, or A > B.
- Considertwonumbers,AandB,withtwodigitseach.Nowwritingthecoefficientsofthenumbersin descending order of significance:

$$A=A_1A_0$$
$$B=B_1B_0$$

- The two numbers are equal if all pairs of significant digits are equal i.e. if and only if A1 = B1, and A0 = B0.
- Whenthenumbersare binary,thedigitsareeither1or0,and theequality ofeachpairofbitscanbe expressed logically with an exclusive-NOR function as

$$x1=A_1B_1+A_1'B_1'$$

$$\text{And } x0=A_0B_0+A_0'B_0'$$

- TheequalityofthetwonumbersAandBisdisplayedinacombinationalcircuitbyanoutputbinary variable that we designate by the symbol (A = B).
- Thisbinary variableis equalto 1if theinput numbers, A andB, are equal, andis equalto0 otherwise.
- For equality to exist, all xi variables must be equal to 1, a condition that dictates an AND operation of all variables:

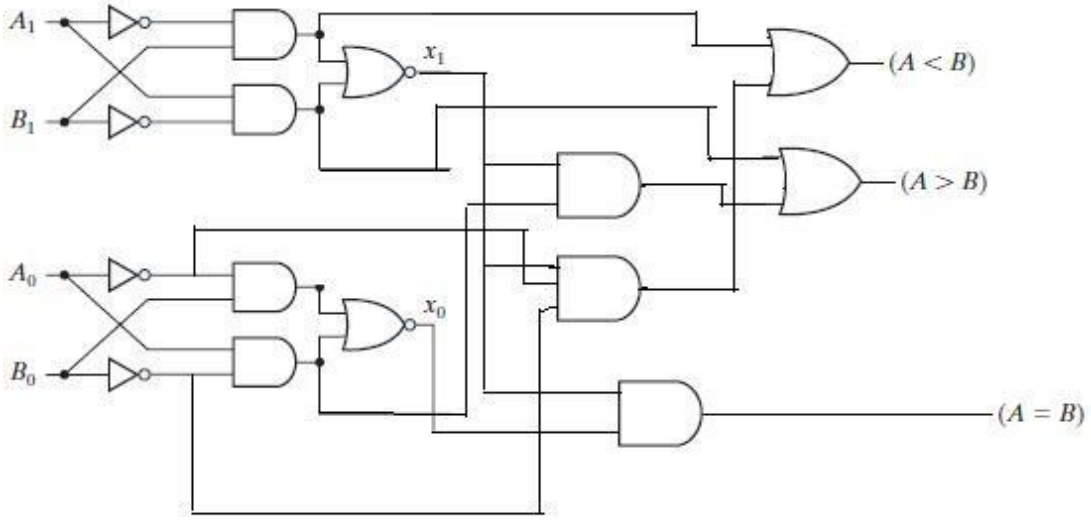
$$(A=B)= x_1x_0$$

- Thebinaryvariable(A =B)is equalto1 onlyif all pairsof digitsof thetwo numbersareequal.
- To determine whether A is greater or less than B, we inspect the relative magnitudes of pairs of significant digits, starting from the most significant position. If the two digits of a pair are equal, we compare the next lower significant pair of digits. If the corresponding digit of A is 1 and that of B is 0, we conclude that A >B. If the correspondingdigit ofA is 0 and that of B is 1, we have A <B. The sequential comparison can be expressed logically by the two Boolean functions

$$(A > B) = A_1B_1'+x_1A_0B'_0$$
$$(A<B)=A_1'B_1+x_1A_0'B_0'$$

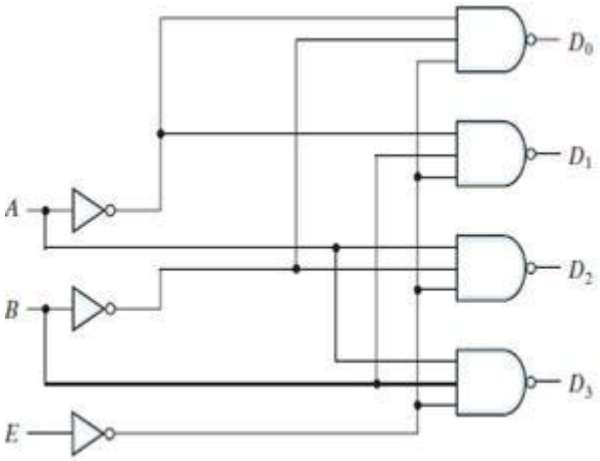
A_1	A_0	B_1	B_0	A>B	A<B	A=B
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

Truth Table



Logic Diagram of 2-bit Magnitude Comparator

DECODER:-



E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
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	1

- A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.
- Ifthen-bitcodedinformationhasunusedcombinations,thedecodermayhavefewerthan2noutputs.
- Thedecoderspresentedherearecalledn-to-m-linedecoders,wherem…2n.
- Theirpurposeisto generatethe 2n(or fewer) mintermsof ninput variables.
- Each combination of inputs will assert a unique output. The name decoder is also used in conjunctionwith other code converters, such as a BCD-to-seven-segment decoder.
- Considerthethree-to-eight-linedecodercircuitofthreeinputsaredecodedintoeightoutputs,each representing one of the minterms of the three input variables.
- Thethreeinvertersprovidethecomplementoftheinputs,andeachoneoftheeightANDgatesgenerates one of the minterms.
- The input variables represent a binarynumber, and the outputs represent theeight digits of a number in the octal number system.
- However,athree-to-eight-linedecodercanbeusedfordecodinganythree-bitcodetoprovideeight outputs, one for each element of the code.
- Atwo-to-four-linedecoderwithanenableinputconstructedwithNANDgatesisshownin Fig.
- The circuit operates with complemented outputs and a complement enable input. The decoder is enabled when E is equal to 0 (i.e., active-low enable). As indicated by the truth table, only one output can be equal to 0 at any given time; all other outputs are equal to 1.
- Theoutputwhosevalueisequalto0representsthemintermselectedbyinputsAandB.
- ThecircuitisdisabledwhenEisequalto1,regardlessofthevaluesoftheothertwoinputs.
- Whenthecircuitisdisabled,noneoftheoutputsareequalto0andnone ofthemintermsareselected.
- Ingeneral,adecodermayoperatewithcomplementedorun-complementedoutputs.
- Theenableinputmaybeactivatedwitha0orwith a1signal.
- Somedecodershavetwoormoreenableinputsthatmustsatisfyagivenlogicconditioninorderto enable the circuit.
- A decoder with enable input can function as a demultiplexer– a circuit that receives information from a single line and directs it to one of 2n possible output lines.
- Theselectionofaspecificoutputiscontrolledbythebitcombinationofn selection lines.
- The decoder of Fig. can function as a one-to-four-line demultiplexer when E is taken as a data input line and A and B are taken as the selection inputs.
- ThesingleinputvariableEhasapathtoallfouroutputs,buttheinputinformationisdirectedtoonly one of the output lines, as specified by the binary combination of the two selection lines A and B .
- Thisfeaturecanbeverifiedfromthetruthtableofthe circuit.
- Forexample,iftheselectionlinesAB=10,outputD_2willbethesameastheinputvalueE,whileall other outputs are maintained at 1.
- Sincedecoderanddemultiplexeroperationsareobtainedfromthesamecircuit,adecoderwithan enableinputisreferredtoasadecoder–demultiplexer.
- Aapplicationofthisdecoderisbinary-to-octalconversion.

ENCODER:-

- Anencoderisadigitalcircuitthatperformstheinverseoperationofadecoder.
- Anencoderhas 2n (or fewer)input lines andn output lines.
- Theoutputlines,asanaggregate,generatethebinarycodecorrespondingtotheinput value.

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

- Theabove Encoder has eightinputs (one foreach oftheoctal digits)and threeoutputs that generate the corresponding binary number.
- Itis assumedthat only oneinput hasavalue of1 atanygiven time.
- TheencodercanbeimplementedwithORgateswhoseinputsaredetermineddirectlyfromthetruth table.
- Outputzis equalto1when theinput octaldigit is1, 3,5, or7.
- Output yis 1for octaldigits 2,3, 6,or 7,andoutput xis 1for digits4, 5,6,or7.
- TheseconditionscanbeexpressedbythefollowingBooleanoutputfunctions:

$$z=D_1+D_3+D_5+D_7$$
$$y=D_2+D_3+D_6+D_7 \quad x$$
$$=D_4+D_5+D_6+D_7$$

- TheencodercanbeimplementedwiththreeORgates.
- Theencoderdefinedabovehasthelimitationthatonlyoneinputcanbeactiveatanygiventime.
- Iftwoinputsareactivesimultaneously,theoutputproducesanundefined combination.
- To resolve this ambiguity, encoder circuits must establish an input priority to ensure that only one input is encoded which is done in the Priority Encoder .

PRIORITYENCODER:-

- Aprioritыencoderisanencodercircuitthatincludestheprioritыfunction.
- The operation of the priority encoder is such that if two or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence.

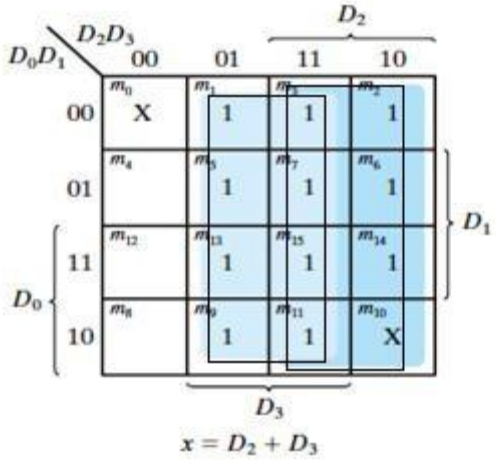
Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

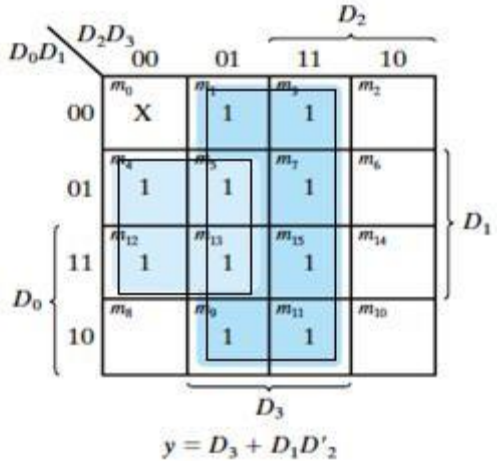
- In addition to the two outputs x and y , the circuit has a third output designated by V ; this is a valid bitindicator that is set to 1 when one or moreinputsareequalto1.

- If all inputs are 0, there is no valid input and V is equal to 0.
- The other two outputs are not inspected when V equals 0 and are specified as don't-care conditions.
- Here X 's in output columns represent don't-care conditions, the X 's in the input columns are useful for representing a truth table in condensed form.

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

- Higher the subscript number, the higher the priority of the input.
- Input D3 has the highest priority, so, regardless of the values of the other inputs, when this input is 1, the output for xy is 11 (binary 3).
- If D2 = 1, provided that D3 = 0, regardless of the values of the other two lower priority inputs the output is 10.
- The output for D1 is generated only if higher priority inputs are 0, and so on down the priority levels.



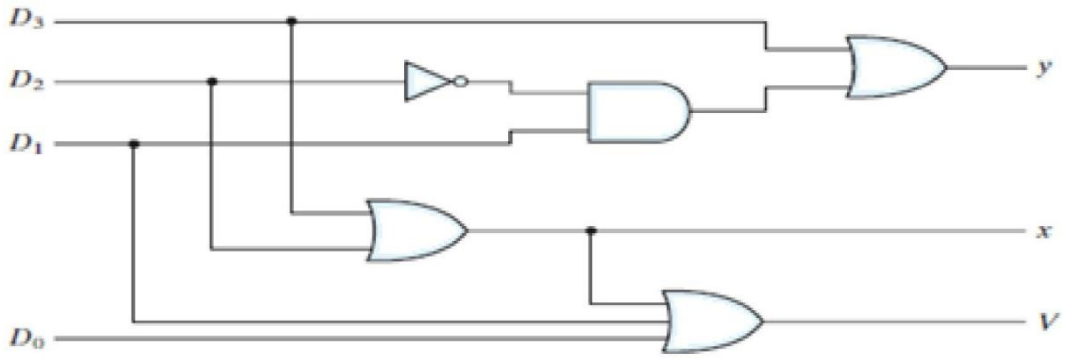


- The maps for simplifying outputs x and y are shown in above Fig.
- The minterms for the two functions are derived from its truth table.
- Although the table has only five rows, when each X in a row is replaced first by 0 and then by 1, we obtain all 16 possible input combinations.
- For example, the fourth row in the table, with inputs XX10, represents the four minterms 0010, 0110, 1010, and 1110. The simplified Boolean expressions for the priority encoder are obtained from the maps.
- The condition for output V is an OR function of all the input variables.
- The priority encoder is implemented according to the following Boolean functions: x =

  $D_2 + D_3$

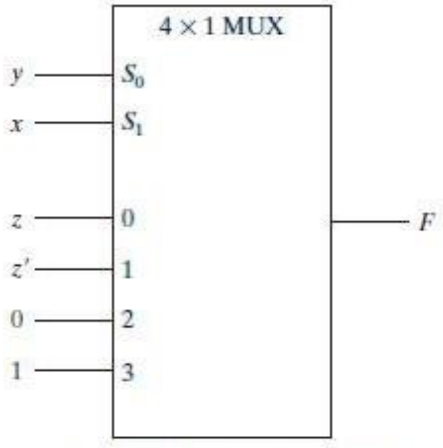
  $y = D_3 + D_1D'_2$

  $V = D_0 + D_1 + D_2 + D_3$

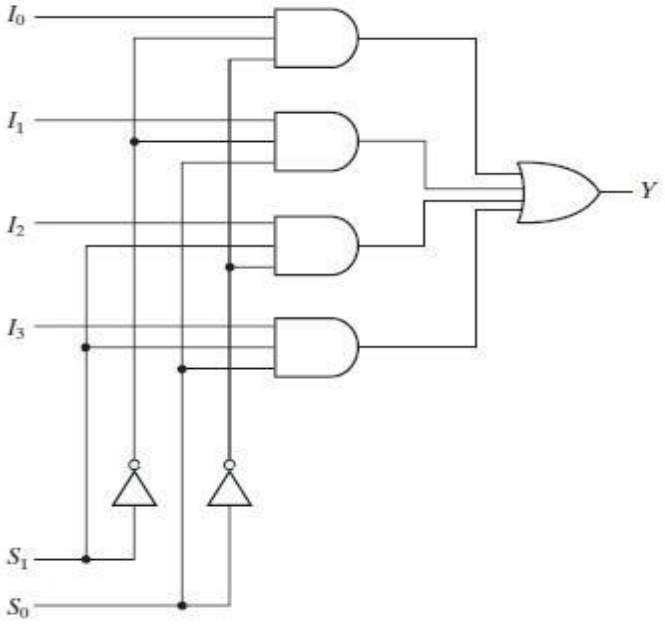


MULTIPLEXER:-

- A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line.
- Theselectionofaparticularinputlineiscontrolledbyasetofselectionlines.
- Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.
- Afour-to-one-linemultiplexerisshowninthebelowfigure.Eachofthefourinputs,I_0throughI_3,is applied to one input of an AND gate.
- SelectionlinesS_1andS_0aredecodedtoselectaparticularAND gate.TheoutputsoftheAND gates are applied to a single OR gate that provides the one-line output.
- Thefunction table lists theinputthat ispassedto the outputfor eachcombinationof thebinaryselection values.
- Todemonstratetheoperationofthecircuit,considerthecasewhen
  S_1S_0= 10.
- The AND gate associated with input I_2 has two of its inputs equal to 1 and the third input connected toI_2.
- TheotherthreeANDgateshaveatleastoneinputequalto0,whichmakestheiroutputsequalto0. TheoutputoftheORgateisnowequaltothevalueofI_2,providingapathfromtheselectedinputto the output.
- Amultiplexerisalsocalledadataselector,sinceitselectsoneofmanyinputsandsteersthebinary information to the output line.



(b) Multiplexer implementation



Logic diagram

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

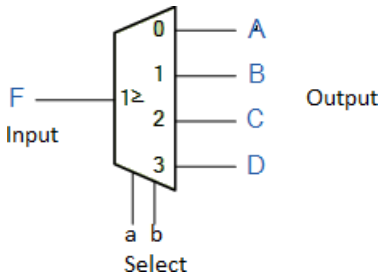
Truth table

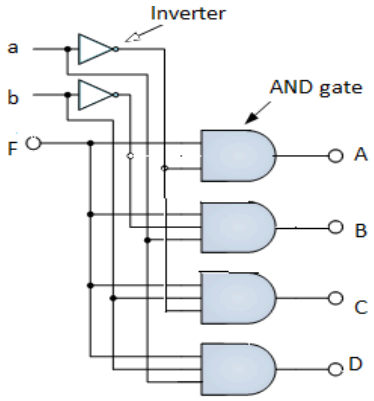
DEMULTIPLEXER:-

- Thedatadistributor,knownmorecommonlyasaDemultiplexeror“Demux”forshort,istheexact opposite of the Multiplexer.
- Thedemultiplexertakesonesingleinputdatalineandthenswitchesittoanyoneofanumberof individualoutputlinesoneatatime.Thedemultiplexerconvertsaserialdatasignalattheinputtoa parallel data at its output lines as shown below.
- The Boolean expression for this 1-to-4 demultiplexer above with outputs A to D and data select lines a,b is given as:

$$F=(ab)'A+a'bB+ab'C+abD$$

- The function ofthe demultiplexer is to switch one common data input line to any one of the 4 outputdata lines A to D in our example above. As with the multiplexer the individual solid state switches are selected by the binary input address code on the output select pins “a” and “b” as shown.





Logic Diagram

- Unlike multiplexers which convert datafrom asingle data line tomultiple lines anddemultiplexerswhich convert multiple lines to a single data line, there are devices available which convert data to and from multiple lines and in the next tutorial about combinational logic devices.
- Standard demultiplexer IC packages available are the TTL 74LS138 1 to 8-output demultiplexer, theTTL 74LS139 Dual 1-to-4 output demultiplexer or the CMOS CD4514 1-to-16 output demultiplexer.

Output Select		Data output Selected
b	a	
0	0	A
0	1	B
1	0	C
1	1	D

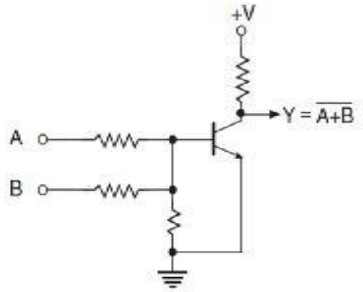
Truth Table

LOGIC FAMILIES

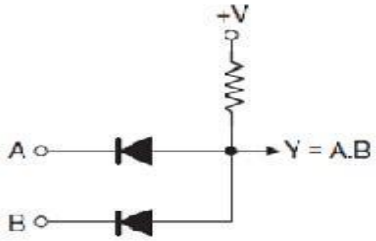
- A circuit configuration or approach used to produce a type of digital integrated circuit is called Logic Family.
- By using logic families we can generate different logic functions, when fabricated in the form of an IC with the same approach, or in other words belonging to the same logic family, will have identical electrical characteristics.
- ThesetofdigitalICsbelongingtothesamelogicfamilyareelectricallycompatiblewitheachother.
- Some common Characteristics of the Same Logic Family include Supply voltage range, speed of response, power dissipation, input and output logic levels, current sourcing and sinking capability, fan-out, noise margin, etc.
- Choosing digital ICs from the same logic family guarantees that these ICs are compatible with respectto each other and that the system as a whole performs the intended logic function.

TYPESOF LOGICFAMILY:-

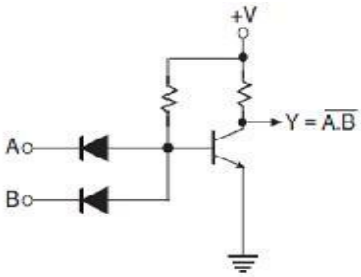
- TheentirerangeofdigitalICsisfabricatedusingeitherbipolardevicesorMOSdevicesora combination of the two.
- Bipolarfamiliesinclude:-
  - Diodelogic(DL)
  - Resistor-Transistor logic (RTL)
  - Diode-transistor logic (DTL)
  - Transistor-Transistorlogic(TTL)
  - Emitter Coupled Logic (ECL),
    - (alsoknownasCurrentModeLogic(CML))
  - Integrated Injection logic (I2L)
- TheBi-MOSlogicfamilyusesbothbipolarandMOSdevices.



Resistor -Transistor logic (RTL)



Diode Logic (DL)



Diode-Transistor logic (DTL)

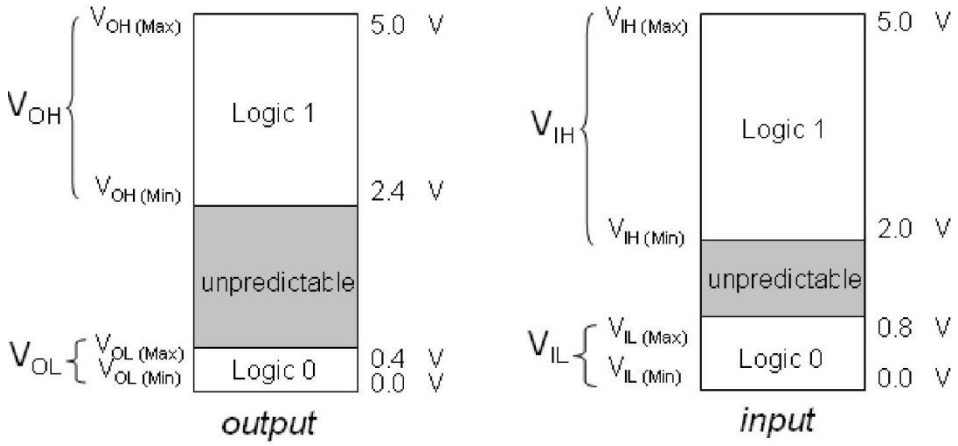
- Abovearesomeexampleof DL,RTLandDTL.
- MOSfamiliesinclude:-
  - ThePMOS family (using P-channel MOSFETs)
  - TheNMOSfamily(usingN-channelMOSFETs)
  - TheCMOSfamily(usingbothN- andP-channeldevices) SOME

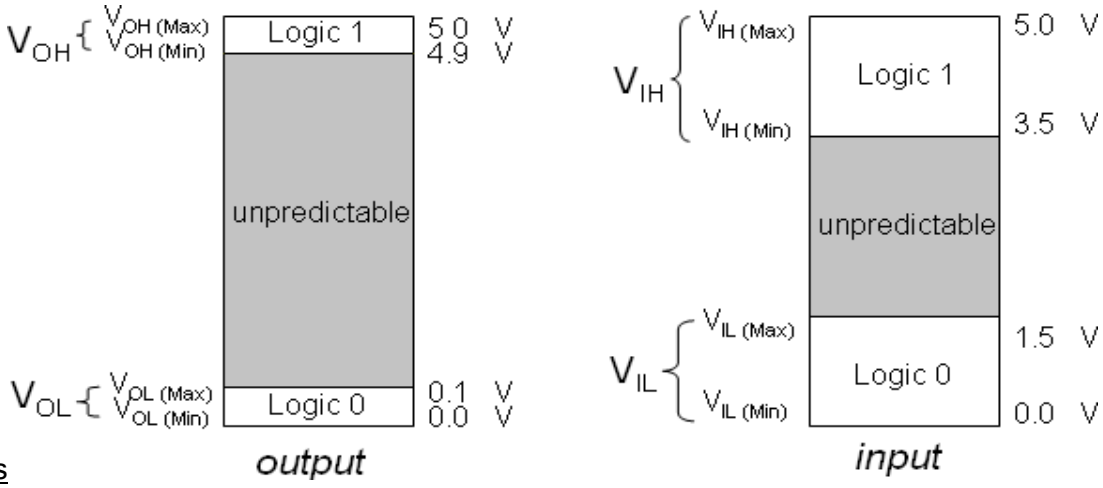
OPERATIONAL PROPERTIES OF LOGIC FAMILY:-

DCSupplyVoltage:-

- ThenominalvalueofthedcsupplyvoltageforTTL(transisitor-transistor logic) and CMOS (complementary metal-oxide semiconductor) devices is +5V.Althoughommittedfromlogicdiagramsforsimplicity,thisvoltage is connected to Vcc or VDD pin of an IC package and ground is connected to the GND pin.

TTLLogic Levels





CMOSLogicLevels

NoiseImmunity:-

- Noise is the unwanted voltage that is induced in electrical circuits and can present a threat to the poor operation of the circuit. In order not to be adverselyeffected by noise, a logic circuit must have a certain amount of 'noise immunity'.
- This is the ability to tolerate a certain amount of unwanted voltage fluctuation on its inputs without changing its output state is called Noise Immunity.

NoiseMargin:-

- Ameasureofacircuit'snoiseimmunityiscalled'noisemargin'whichisexpressedinvolts.
- There are two values of noise margin specified for a given logic circuit: the HIGH (V_{NH}) and LOW (V_{NL}) noise margins.

Thesearedefinedbyfollowing equations:
$V_{NH}=V_{OH}(Min)-V_{IH}(Min)V_{NL}=V_{IL}(Max)- V_{OL}(Max)$

PowerDissipation:-

- A logic gate draws ICCHcurrent from the supply when the gate is in the HIGH output state, draws ICCL current from the supply in the LOW output state.
- Averagepower is
  PD= VCCICC where ICC= (ICCH +ICCL) /2

PropagationDelaytime:-

- When a signal passes (propagates) through a logic circuit, it always experiences a time delay asshown below. A change in the output level always occurs a short time, called 'propagation delay time' , later than the change in the input level that caused it.

Fan Out ofGates:-

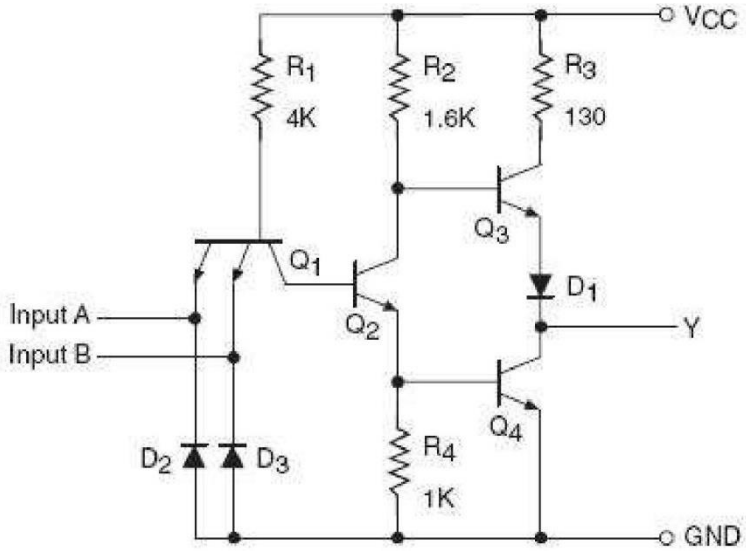
- When the output of a logic gate is connected to one or more inputs of other gates, a load on the driving gate is created. There is a limit to the number of load gates that a given gate can drive. This limit is called the 'Fan-Out' of the gate.

TRANSISTOR-TRANSISTORLOGIC:-

- InTransistor-TransistorlogicorjustTTL,logicgatesarebuiltonlyaroundtransistors.
- TTL was developed in 1965. Through the yearsbasicTTL has been improved to meet performance requirements. There are many versions or families of TTL.
- Forexample
    - Standard TTL
    - HighSpeedTTL(twiceasfast,twiceasmuch power)
    - LowPowerTTL(1/10thespeed,1/10thepowerof“standard"TTL)
    - SchhottkyTTL etc.(forhigh-frequency uses)
- AllTTLlogicfamilieshavethreeconfigurationsforoutputs
    1. Totempole output
    2. Opencollectoroutput
    3. Tristateoutput

Totempole output:-

- Additionofanactivepullupcircuitintheoutputofagateiscalledtotem pole.
- Toincreasetheswitchingspeedofthegatewhichislimitedduetotheparasiticcapacitanceatthe output totem pole is used.
- Thecircuitofatotem-poleNANDgateisshownbelow,whichhasgotthree stages
    1. Input Stage
    2. PhaseSplitterStage
    3. OutputStage



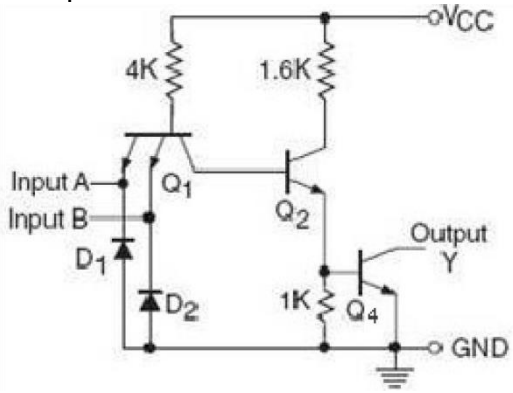
- Transistor Q1 is a two-emitter NPN transistor, which is equivalent two NPN transistors with their baseand emitter terminals tied together.
- Thetwo emitters arethetwo inputs of the NANDgate
  In TTL technology multiple emitter transistors are used for the input devices DiodesD2andD3areprotectiondiodesusedtolimitnegativeinputvoltages.
- Whenthereislargenegativevoltageatinput,thediodeconductsandshortingittothegroundQ2 provides complementary voltages for the output transistors Q3 and Q4.
- ThecombinationofQ3andQ4formstheoutputcircuitoftenreferredtoasatotempolearrangement (Q4is stackedon topofQ3).Insuch anarrangement,eitherQ3orQ4 conductsata timedepending upon the logic status of the inputs
  DiodeD1ensuresthatQ4 willturn off whenQ2ison(HIGHinput) The output Y is taken from the top of Q3

AdvantagesofTotemPoleOutput:-

- Thefeaturesofthisarrangement are
    1. Lowpower consumption
    2. Fast switching
    3. Lowoutputimpedance

OPEN COLLECTOR OUTPUT:-

- FigurebelowshowsthecircuitofatypicalTTLgatewithopen-collectoroutputObserveherethatthe circuit elements associated with Q3 in the totem-pole circuit are missing and the collector of Q4 is left open-circuited, hence the name open-collector.



- An open-collector output can present a logic LOW output. Since there is no internal path from theoutput Y to the supply voltage V_{CC} , the circuit cannot present a logic HIGH on its own.

AdvantagesofOpenCollectorOutputs:-

- Open-collectoroutputscanbetieddirectlytogetherwhichresultsinthelogicalANDingoftheoutputs. ThustheequivalentofanANDgatecanbeformedbysimplyconnectingthe outputs.
- Increased current levels - Standard TTL gates with totem-pole outputs can only provide a HIGH current output of 0.4 mA and a LOW current of 1.6 mA. Many open-collector gates have increased current ratings.
- DifferentvoltagelevelsAwidevarietyofoutputHIGHvoltagescanbeachievedusingopen-collector gates. This is useful in interfacing different logic families that have different voltage and current level requirements.

Disadvantageofopen-collectorgates:-

- They have slow switching speed. This is because the value ofpull-up resistor is in kW, which results ina relatively long time Constants

ComparisonofTotemPoleandOpenCollectorOutput:-

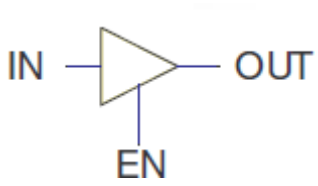
- The major advantage of using a totem-pole connection is that it offers low-output impedancein both the HIGH and LOW output states

Totem Pole	Open Collector
Output stage consists of pull-up transistor (Q3), diode resistor and pull-down transistor (Q4)	Output stage consists of only pull-down transistor
External pull-up resistor is not required	External pull-up resistor is required for proper operation of gate
Output of two gates cannot be tied together	Output of two gates can be tied together using wired AND technique
Operating speed is high	Operating speed is low

TRISTATE(THREE-STATE)LOGICOUPUT:-

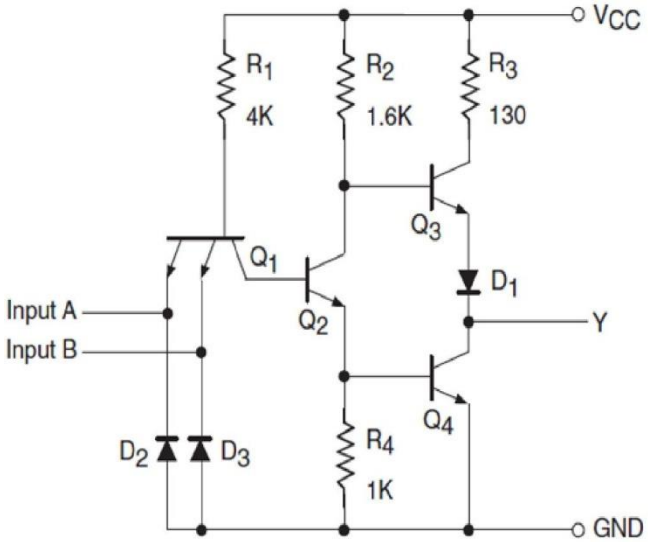
- Tristateoutputcombinestheadvantagesofthetotem-poleandopencollectorcircuits.
- ThreeoutputstatesareHIGH,LOW,andhighimpedance(Hi-Z).

EN	IN	OUT
0	X	HI-Z
1	0	0
1	1	1



- For the symbol and truth table, IN is the data input, and EN, the additional enable input for control. For EN = 0, regardless of the value on IN(denoted by X), the output value is Hi-Z. For EN = 1, the output value follows the input value.
- Data input, IN,can be inverted. Control input, EN, can be inverted by addition of"bubbles" to signals IN OUT EN.
- Thisrequirestwoinputs:inputandenableENistomakeoutputHi-Zorfollowinput.

STANDARDTTLNANDGATE:



A	B	Q1	Q2	Q3	Q4	Y
L	L	sat	off	off	Off	H
L	H	sat	off	off	Off	H
H	L	sat	off	off	Off	H
H	H	ia m	sat	sat	On	L

CMOSTECHNOLOGY:-

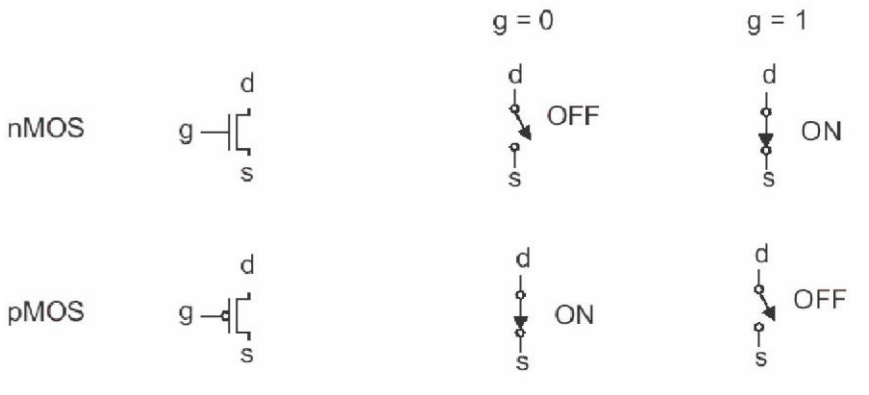
- MOSstandsforMetalOxideSemiconductorandthistechnologyusesFETs.
- MOScanbeclassifiedintothreesub-families:
  PMOS (P-channel)
  NMOS(N-channel)
  CMOS(ComplementaryMOS,most common)
- The following simplified symbols are used to represent MOSFET transistors in most CMOS. The gateof a MOS transistor controls the flow of the current between the drain and the source. The MOS transistor can be viewed as a simple ON/OFF switch.

AdvantagesofMOSDigitalICs:-

- TheyaresimpleandinexpensivetofabricateCanbeusedforHigherintegrationandconsumelittlepower.

DisadvantagesofMOSDigitalICs:-

- ThereispossibilityforStatic-electricitydamage.
- Theyareslower thanTTL.

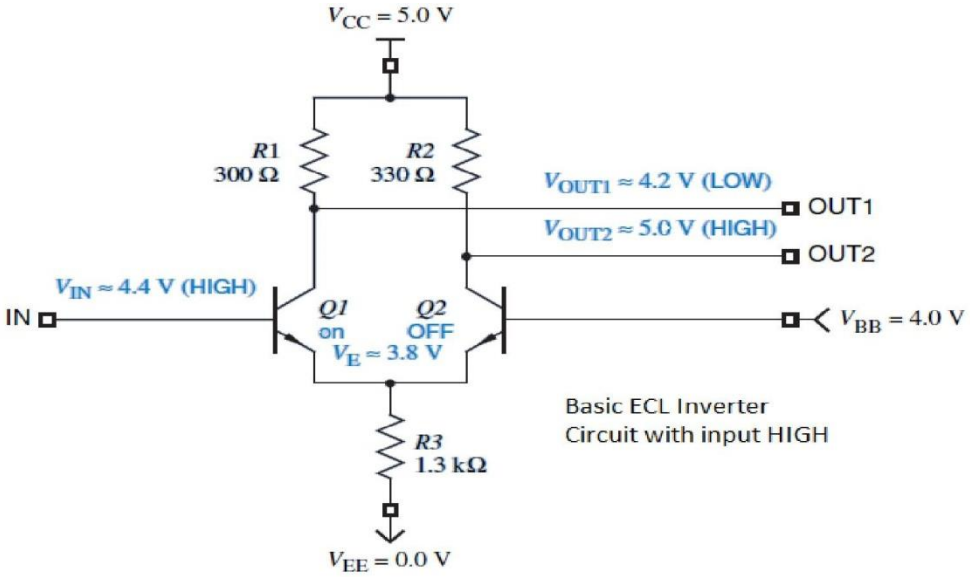


ECL:EMITTER-COUPLEDLOGIC:-

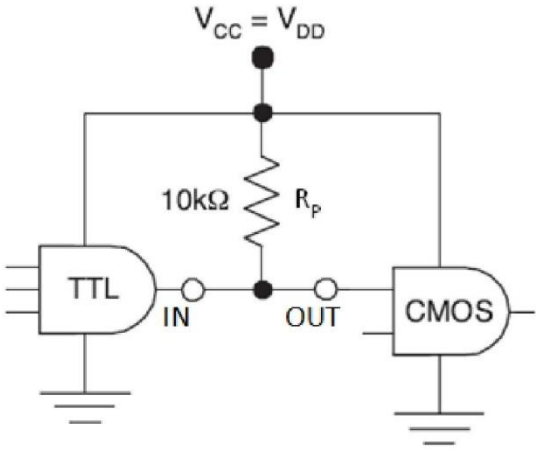
- The key to reduce propagation delay in a bipolar logic family is to prevent a gate's transistors from saturating. Itis possible to prevent saturation by using a radically different circuit structure, called current-mode logic (CML) or emitter-coupled logic (ECL).
- Unliketheotherlogicfamiliesinthischapter,ECLdoesnotproducealargevoltageswingbetweenthe LOWand HIGH levels but it has a small voltage swing, less than a volt, and it internally switchescurrent between two possible paths, depending on the output state.

BasicECL Circuit

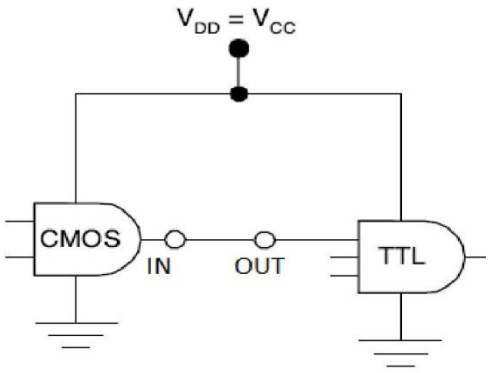
- The basic idea of current-mode logic is illustrated by the inverter/buffer circuit in the figure. This circuit has both an inverting output (OUT1) and a non-inverting output (OUT2).
- Twotransistorsareconnectedasadifferentialamplifierwithacommonemitterresistor.
- Thesupply voltages forthisexample areVCC = 5.0,VBB =4.0,andVEE=0V,and theinput LOW and HIGH levels are defined to be 3.6 and 4.4 V. This circuit actually produces output LOW and HIGH levels that are 0.6 V higher (4.2 and 5.0 V).



INTERFACINGOFTTLTOCMOS



INTERFACINGOFCMOSTOTTL



TTLvs. CMOS:-

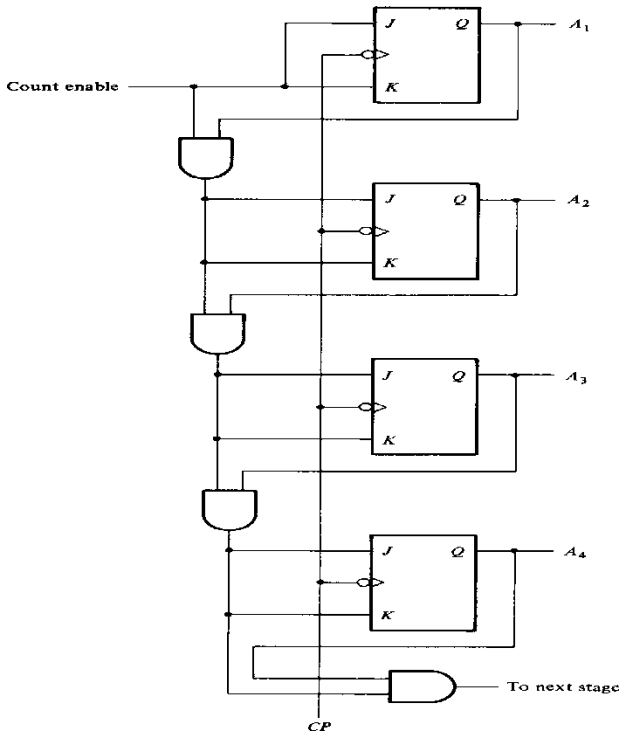
- TTLhaslesspropagationdelaythanCMOSi.e.TTLisgoodwherehighspeedis needed.
- AndCMOS4000isgoodforBatteryequipmentandwherespeedisnotso important.
- CMOSrequireslesspowerthanTTLi.e.powerdissipationandhencepowerconsumptionislessfor CMOS.

COUNTER

- A counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred. In electronics, counters can be implemented quite easily using register-type circuits.
- Therearedifferenttypesofcounters,viz.
    - Asynchronous(ripple)counter
    - Synchronous counter
    - Decadecounter
    - Up/downcounter
    - Ringcounter
    - Johnsoncounter
    - Cascadedcounter
    - Modulus counter.

Synchronouscounter

- A4-bitsynchronouscounterusingJKflip-flopsisshowninthe figure.
- In synchronouscounters, the clock inputsof alltheflip-flopsare connected together and are triggered by the input pulses. Thus, all the flip-flops change state simultaneously (in parallel).

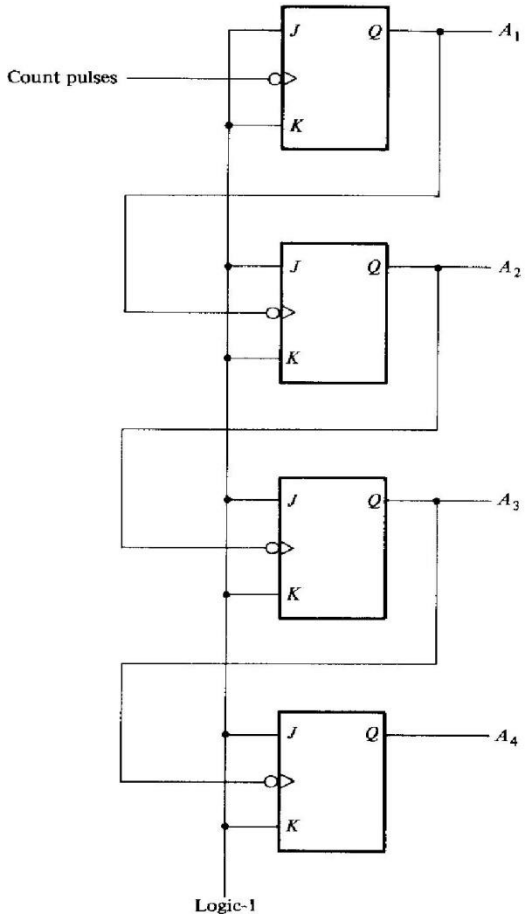


- Thecircuitbelowisa4-bitsynchronous counter.
- The J andK inputs of FF0 areconnectedto HIGH. FF1 has its Jand K inputs connectedto the output of FF0, and the J and K inputs of FF2 are connected to the output of an AND gate that is fed by the outputs of FF0 and FF1.
- Asimplewayofimplementingthelogicforeachbitofanascendingcounter(whichiswhatisdepicted in the image to the right) is for each bit to toggle when all ofthe less significantbits are ata logic highstate.
- For example, bit 1 toggles when bit 0 is logic high; bit 2 toggles when both bit 1 and bit 0 are logic high; bit 3 toggles when bit 2, bit 1 and bit 0 are all high; and so on.

- Synchronous counters can also be implemented with hardware finite state machines, which are morecomplex but allow for smoother, more stable transitions.

AsynchronousCounter

- An asynchronous (ripple) counter is a single d-type flip-flop, with its J (data) input fed from its own inverted output.
- This circuit can store onebit, and hence can count from zero to one before it overflows (starts over from 0).



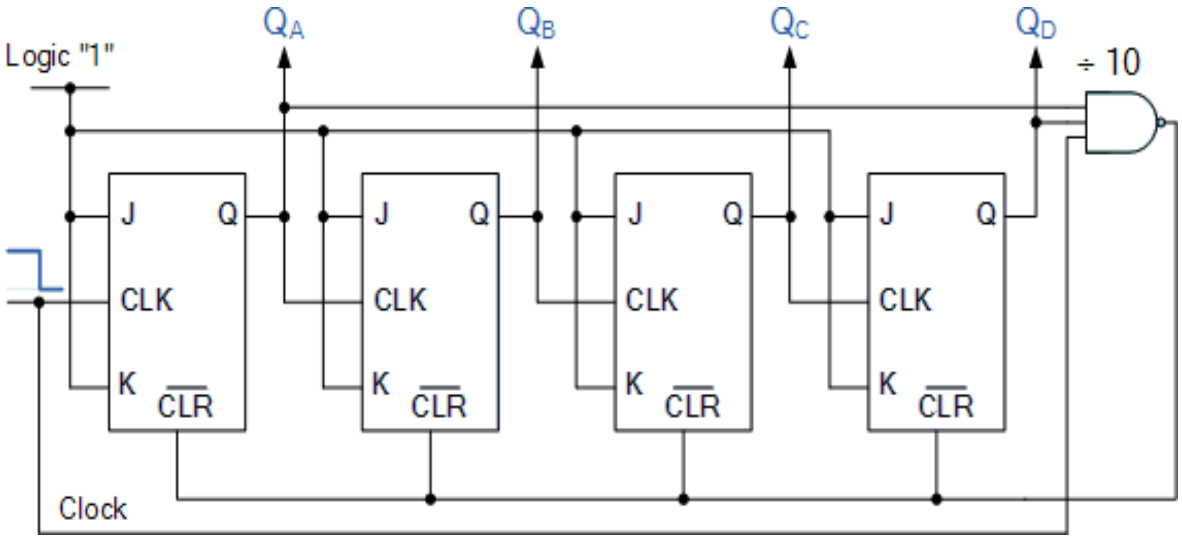
- This counter will increment once for every clock cycle and takes two clock cycles to overflow, so every cycle it will alternate between a transition from 0 to 1 and a transition from 1 to 0.
- Thiscreates anewclock with a50% duty cycleat exactlyhalfthefrequency ofthe input clock.
- If this output is then used as the clock signal for a similarly arranged D flip-flop, remembering to invert the output to the input, one will get another 1 bit counter that counts half as fast. These together yield a two-bit counter.
- Additional flip-flops can be added, by always inverting the output to its own input, and using the output from the previous flip-flop as the clock signal. The result is called a ripplecounter, which can countto $2^n - 1$, where n is the number of bits (flip-flop stages) in the counter.
- Ripple counters suffer from unstable outputs as the overflows "ripple" from stage to stage, but they find application as dividers for clock signals.

ModulusCounter

- Amoduluscounteristhatwhichproducesanoutputpulseafteracertainnumberofinputpulsesis applied.
- Inmoduluscounterthetotalcountpossibleisbasedonthenumberof stages,i.e.,digitpositions.

- Moduluscountersareusedindigitalcomputers.
- A binary modulo-8 counter with three flip-flops, i.e., three stages, will produce an output pulse, i.e., display an output one-digit, after eight input pulses have been counted, i.e., entered or applied. This assumes that the counter started in the zero-condition.

AsynchronousDecadeCounter



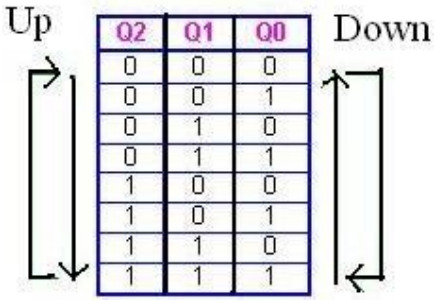
- Adecadecounter cancountfromBCD“0” toBCD “9”.
- A decade counter requires resetting tozero when the output countreaches thedecimal value of 10, ie. when DCBA = 1010 and this condition is fed back to the reset input.
- A counter with a count sequence from binary “0000” (BCD = “0”) through to “1001” (BCD = “9”) is generally referredto as aBCDbinary-coded-decimal counter becauseits tenstate sequence is that of a BCD code but binary decade counters are more common.
- Thistypeofasynchronouscountercountsupwardsoneachleadingedgeoftheinputclock signal starting from 0000 until it reaches an output 1001 (decimal 9).
- Both outputs Q_A and $Q_D$are now equal to logic “1” and the output from the NAND gate changes state from logic “1” to a logic “0” level and whose output is also connectedto theCLEAR(CLR) inputs of all the J-K Flip-flops.
- This signalcauses allof theQ outputsto be reset back to binary0000 on the count of 10. OnceQA and QD are both equal to logic “0” the output of the NAND gate returns back to a logic level “1” and the counter restarts again from 0000. We now have a decade or Modulo-10 counter.

DecadeCounterTruthTable

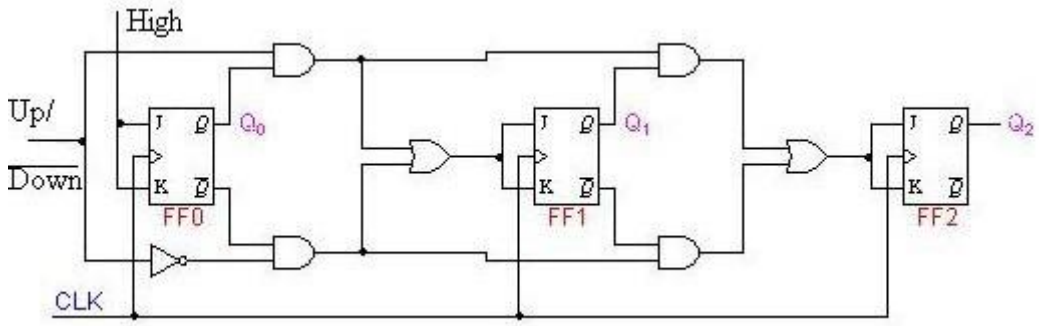
Clock Count	Output bit Pattern				Decimal Value
	QD	QC	QB	QA	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	Counter Resets its Outputs back to Zero				

Up/DownCounter

- In a synchronous up-down binary counter the flip-flop in the lowest-order position is complemented with every pulse.
- A flip-flop in any other position is complemented with a pulse, provided all the lower-order pulse equal to 0.
- Up/Downcounterisusedtocontrolthedirectionofthecounterthroughacertain sequence.



- Fromthesequencetablewecanobservethat:
  - For boththe UPandDOWNsequences,Q_0toggles on eachclock pulse.
  - FortheUPsequence,Q_1changesstateonthenextclockpulsewhenQ_0=1.
  - FortheDOWNsequence,Q_1changesstateonthenextclockpulsewhen Q_0=0.
  - FortheUPsequence,Q_2changesstateonthenextclockpulsewhenQ_0=Q_1=1.
  - FortheDOWNsequence,Q_2changesstateonthenextclockpulsewhen Q_0=Q_1=0.



- These characteristics are implemented with the AND, OR & NOT logic connected as shown in the logic diagram above.

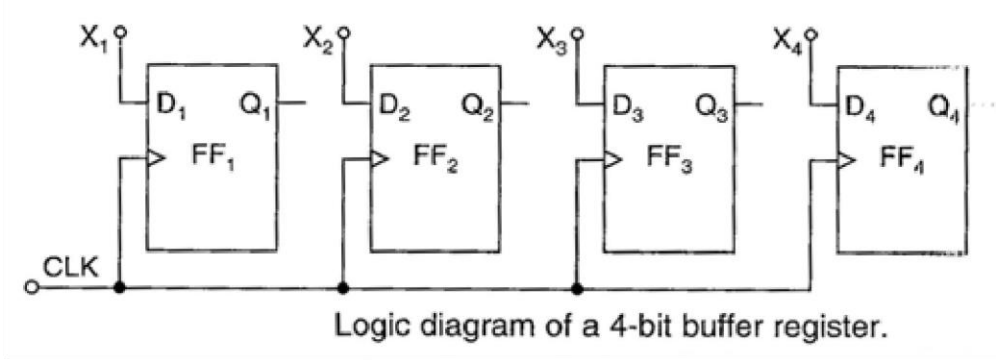
REGISTERS

INTRODUCTION:-

- Thesequentialcircuitsknownasregisterareveryimportantlogicalblockinmostofthedigitalsystems.
- Registersareusedforstorage andtransferofbinaryinformationinadigitalsystem.
- Aregisterismostlyusedforthepurposeofstoringandshiftingbinarydataenteredintoitfroman external source and has no characteristics internal sequence of states.
- Thestoragecapacityofaregisterisdefinedasthenumberofbitsofdigitaldata,itcanstoreorretain.
- Theseregistersarenormallyusedfortemporarystorage of data.

BUFFERREGISTER:-

- Thesearethesimplestregistersandareusedforsimplystoringabinaryword.
- ThesemaybecontrolledbyControlledBufferRegister.
- Dflip-flopsareused forconstructingabufferregisterorotherflip- flopcanbeused.
- Thefigureshownbelowisa4-bitbufferregister.



Logic diagram of a 4-bit buffer register.

- Thebinarywordtobestoredisappliedtothedataterminals.
- Whentheclockpulseisapplied,theoutputwordbecomesthesameasthewordappliedattheinput terminals, i.e. the input word is loaded into the register by the application of clock pulse.
- Whenthepositiveclockedgearrives,thestoredword becomes:

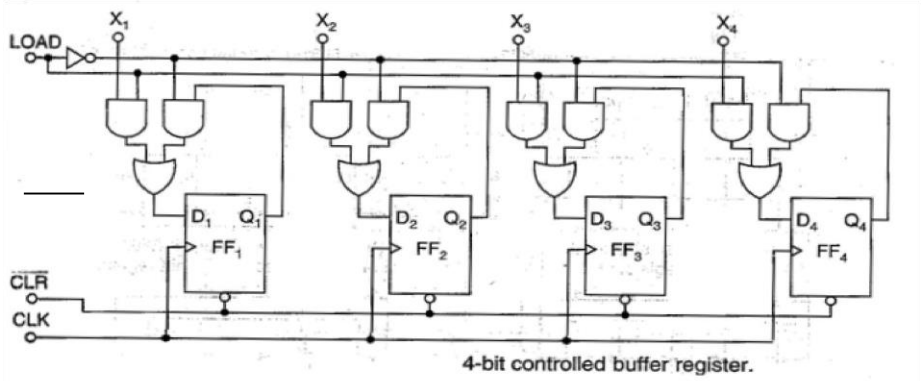
  Q4Q3Q2Q1=X4X3 X2X1

  or Q =X.

  Thiscircuitistooprimitivetobeofanyuse.

CONTROLLEDBUFFERREGISTER:-

- Thefigureshowsacontrolledbuffer register.



4-bit controlled buffer register.

- IfCLRgoesLOW,alltheflip-flopsare RESETand the outputbecomes, Q= 0000.
- WhenCLRisHIGH,theregisterisreadyforaction

- LOADiscontrolinput.
- WhenLOADis HIGH, thedata bitsX can reachthe D inputsofFFs.
- Atthepositivegoingedgeofthenextclockpulse,theregisterisloaded,i.e.

  Q4 Q3Q2Q1=X4X3X2X1

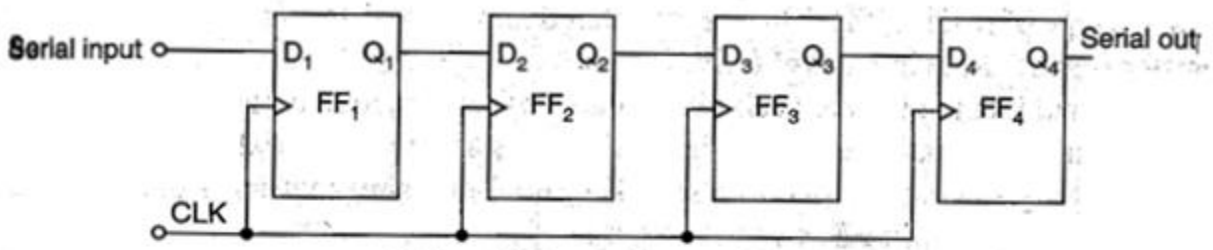
  or Q =X.
- WhenLOADisLOW,theXbitscannotreachtheFFs.Atthesametimetheinvertedsignal LOAD is HIGH. This forces each flip-flop output to feedback to its data input.
- Thereforedataiscirculatedorretainedaseachclockpulsearrives.
- Inotherwordsthecontentregisterremainsunchangedinspiteoftheclock pulses.
- LongerbufferregisterscanbuiltbyaddingmoreFFs.

CONTROLLEDBUFFERREGISTER:-

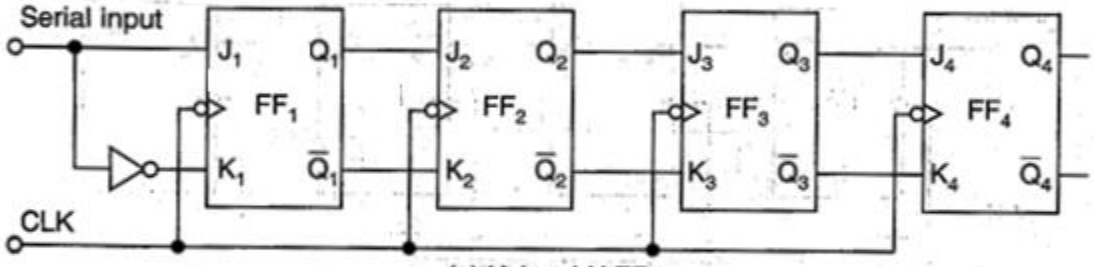
- A number of FFsconnected togethersuch that data maybe shifted into and shifted out of them is called a shift register.
- Datamaybeshiftedintooroutoftheregistereitherinserialformorinparallelform.
- Therearefourbasictypesofshift registers
  1. Serialin,serialout
  2. Serialin,parallelout
  3. Parallelin,serialout
  4. Parallelin,parallelout

SERIALIN,SERIALOUTSHIFT REGISTER:-

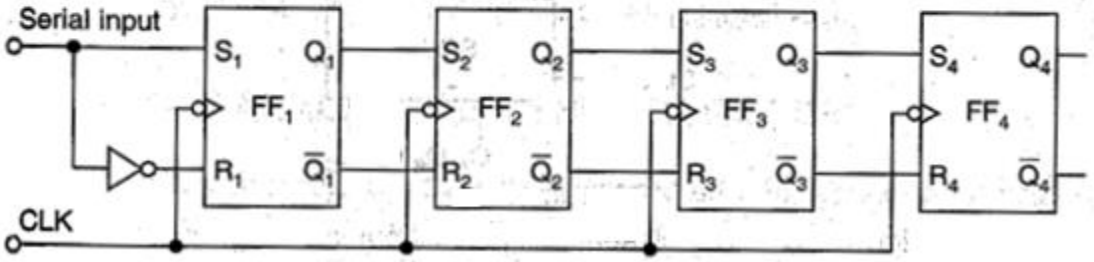
- Thistypeofshiftregisteracceptsdataserially,i.e.,onebitatatimeandalsooutputsdataserially.
- Thelogicdiagramofafourbitserialin,serialoutshiftregisterisshowninbelowfigure:
- In4stagesi.e.with4FFs,theregistercanstoreupto 4bitsofdata.
- Serial data is applied at the D input of the first FF. The Q output of the first FF is connected to the Dinput ofthe second FF,the output ofthe second FF is connected to the D input ofthe third FF and theQ output of the third FF is connected to the D input of the fourth FF. The data is outputted from the Q terminal of the last FF.
- Whenaserialdataistransferredtoaregister,eachnewbitisclockedintothefirstFFatthepositive goingedgeofeachclockpulse.
- ThebitthatispreviouslystoredbythefirstFFistransferredto thesecondFF.
- Thebitthatis storedbythe secondFFis transferredtothe thirdFF,andsoon.
- Thebitthatwas storedbythelast FFisshiftedout.
- Ashiftregister canalsobe constructedusing J-KFFsor S-RFFsasshownin thefigurebelow.



4-bit serial-in, serial-out, shift-right, shift register.



(a) Using J-K FFs

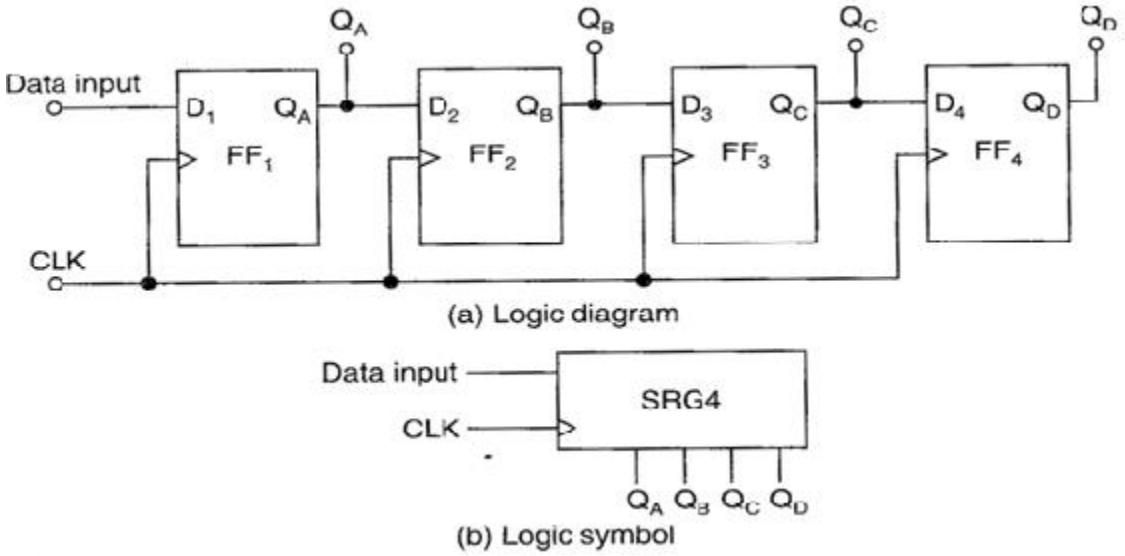


(b) Using S-R FFs

A 4-bit serial-in, serial-out, shift register.

SERIALIN,PARALLELOUTSHIFTREGISTER:-

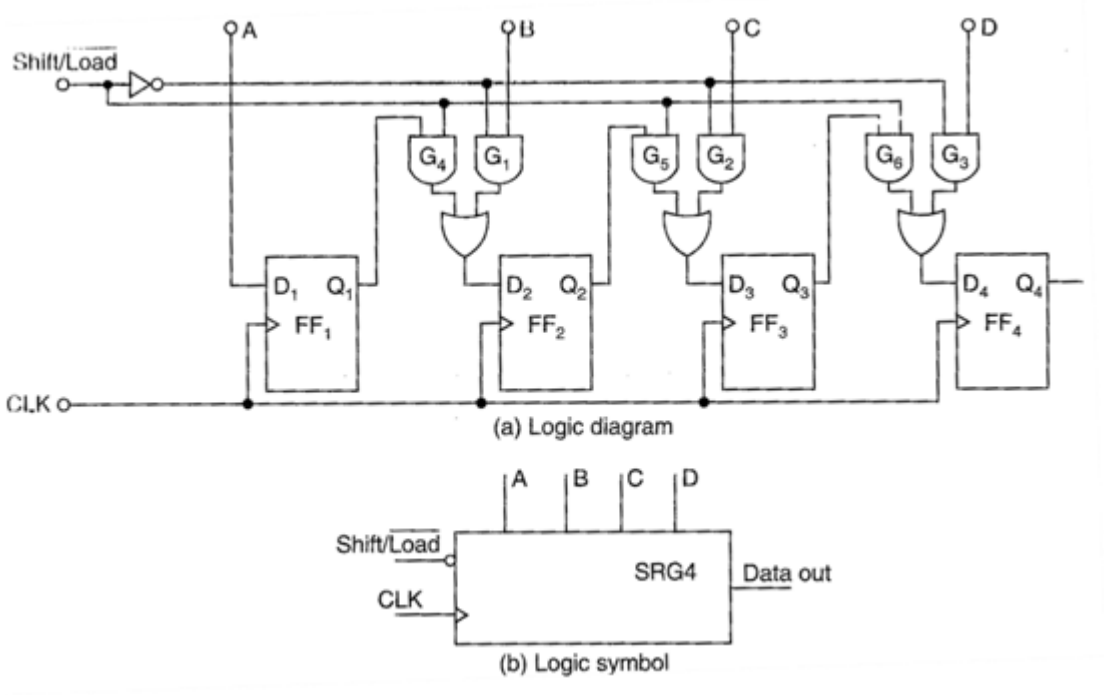
- Inthistypeofregister,thedatabitsareenteredintotheregisterserially,butthedatastoredinthe register serially, but the stored in the register is shifted out in the parallel form.
- When the data bits are stored once, each bits appears on its respective output line and all bitsare availablesimultaneously,ratherthanbit–by–bitbasisasintheserialoutput.
- The serial in, parallel out shift register can be used as a serial in, serial out shift register if the output is taken from the Q terminal of the last FF.
- Thelogicdiagramandlogicsymbolofa 4bitserialin,parralleloutshiftregister isgiven below.



A4- bitserialin,parralleloutshift register

PARALLELIN,SERIALOUTSHIFTREGISTER:-

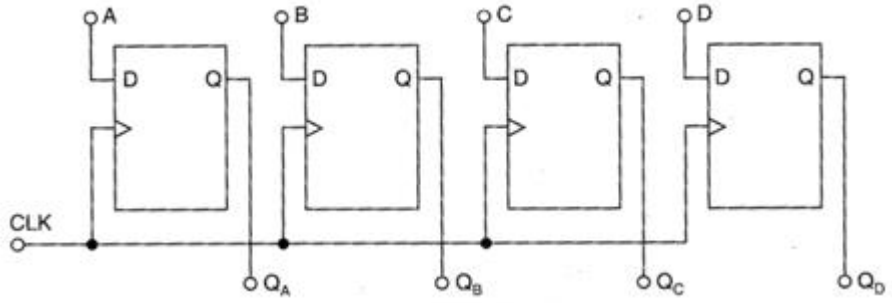
- For parallel in, serial out shift register the data bits are entered simultaneously into their respective stages on parallel lines, rather than on bit by bit basis on one lineas with serial data inputs, but the data bits are transferred out of the register serially, i.e., on a bit by bit basis over a single line.
- The logic diagram and logic symbol of 4 bit parallel in, serial out shift register using D FFs is shown below.
- There are four data lines A, B, C and D through which the data is entered into the register in parallel form.
- ThesignalShift/$\overline{\text{LOAD}}$allows
    1. Thedatatobeenteredinparallelformintotheregisterand
    2. ThedatatobeshiftedoutseriallyfromterminalQ_4.
- When Shift/$\overline{\text{LOAD}}$lineisHIGH,gatesG1,G2,andG3aredisabled,butgatesG4,G5andG6are enabled allowing the data bits to shift right from one stage to next.
- When Shift /LOAD line is LOW, gates G4, G5 and G6 are disabled, whereas gates G1, G2 and G3 are enabled allowing the data input to appear at the D inputs of the respective FFs.
- Whenclockpulseisapplied,thesedatabitsareshiftedtotheQoutputterminalsoftheFFsand therefore the data is inputted in one step.
- The OR gate allows either the normal shifting operation or the parallel data entry depending on whichAND gates are enabled by the level on the Shift /LOAD input.



A4- bitparallelin,serialoutshift register

PARALLELIN,PARALLELOUTSHIFTREGISTER:-

- In a parallel in, parallel out shift register, the data entered into the register in parallel form and also the data taken out of the register in parallel form. Immediately following the simultaneous entry of all data bits appear on the parallel outputs.
- Thefigureshownbelowisa4bitparallelinparalleloutshiftregisterusing DFFs.
- DataappliedtotheDinputterminalsoftheFFs.
- Whenaclockpulseisappliedatthepositiveedgeofthatpulse,theDinputsareshiftedintotheQ outputs of the FFs.
- Theregisternowstoresthedata.
- Thestoreddataisavailableinstantaneouslyforshiftingoutinparallelform.

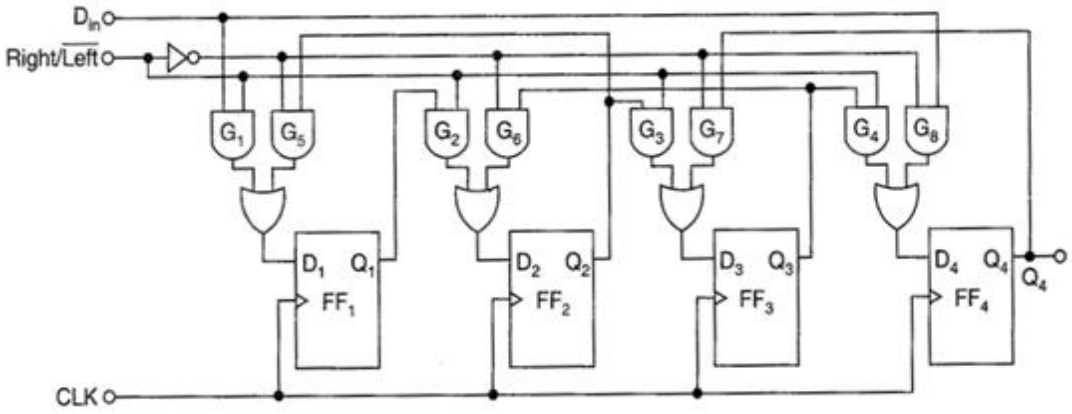


Logicdiagramofa4–bitparallelin,paralleloutshift register

BIDIRECTIONALSHIFT REGISTER:-

- In bidirectional shift register is one in which the data bits can be shifted from left to right or from right to left.
- Thefigureshownbelowthelogicdiagramofa4bitserialin,serialout,bidirectional(shift-left,shift- right) shift register.
- Right/$\overline{\text{Left}}$isthemodesignal.WhenRight/$\overline{\text{Left}}$isa1,thelogiccircuitworksasashiftrightshift register. When Right /Left is a 0, the logic circuit works as a shift right shift register.

- The bidirectional is achieved by using the mode signal and two AND gates and one OR gate for each stage.
- A HIGH on the Right/$\overline{\text{Left}}$ control input enables the AND gates G_1, G_2, G_3 and G_4 and disables the AND gates G_5, G_6, G_7 and G_8 and the state of Q output of each FF is passed through the gate to the D inputof the following FF. When clock pulse occurs, the data bits are effectively shifted one place to the right.
- A LOW Right/$\overline{\text{Left}}$ control input enables the AND gates G_5, G_6, G_7 and G_8 and disables the AND gates G_1, G_2, G_3 and G_4 and the Q outputofeach FF is passed to theD input ofthe preceding FF. When clock pulse occurs the data bits are then effectively shifted one place to the left.
- So,thecircuitworksasabidirectionalshiftregister.



Logicdiagramof4-bitbidirectionalshiftregister

UNIVERSALSHIFTREGISTERS:-

- The register which has both shifts and parallel load capabilities, it is referred as a universal shiftregister. So, universal shift register is a bidirectional register, whose input can be either in serial form or in parallel form and whose output also can be either in serial form or parallel form.
- Theuniversalshiftregistercanberealizedusingmultiplexers.
- The figure shows the logic diagram of a 4 bit universal shift register that has all the capabilities of a general shift register.

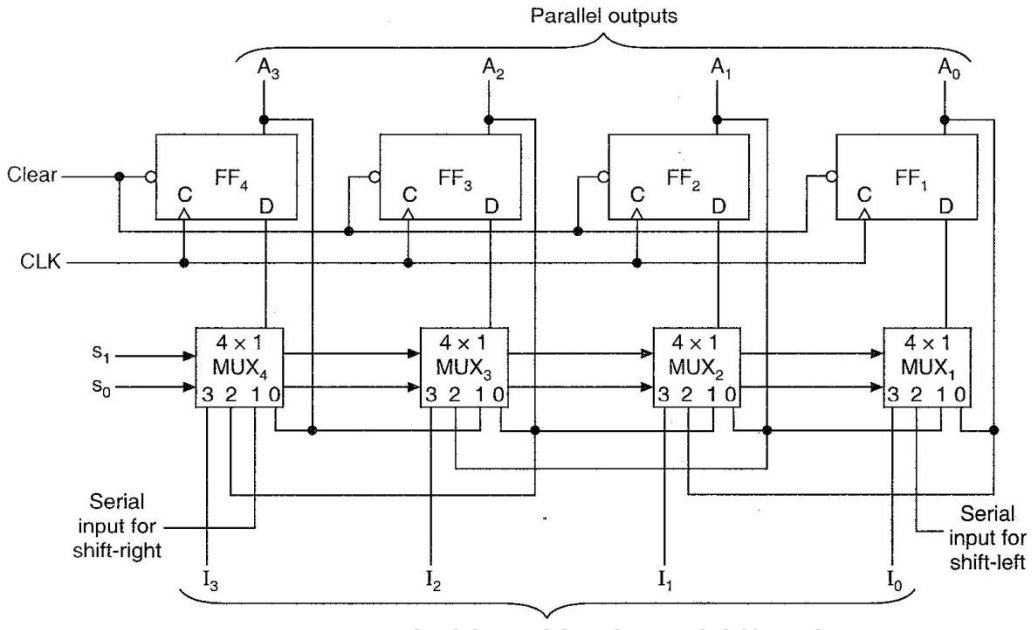


Fig-(a) 4bituniversalshiftregister

- Itconsists offourDflip-flops andfour multiplexers.
- ThefourmultiplexershavetwocommonselectioninputsS_1andS_0.
- Input 0 in each multiplexer is selected when S_1S_0 = 00, input 1 is selected when S_1S_0 = 01, and input 2 is selected when S_1S_0 = 10 and input 3 is selected when S_1S_0= 11.
- Theselectioninputscontrolthemodeofoperationoftheregisterisaccordingtothefunctionentries shown in the table.
- When S_1S_0 = 00 the present value of the register is applied to the D inputsof flip-flops. This conditionforms a path from the output of each FF into the input of the same FF.
- The next clock edge transfers into each FF the binary value it held previously, and no change of stateoccurs.
- When S_1S_0 = 01, terminal 1 ofthe multiplexer inputs have a path of the D inputs of the flip- flops. Thiscauses a shift right operation, with serial input transferred into FF_4.
- WhenS_1S_0=10ashift leftoperationresultswiththeotherserialinputgoingintotheFF_1.
- Finally when S_1S_0 = 11, the binary information on the parallel input lines is transferred into the register simultaneously during the next clock edge.

Functionaltablefortheregisteroffig– a:

Mode control		
S_1	S_0	Register operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

APPLICATIONSOFSHIFTREGISTERS:-

1. **Timedelays:**
   - Indigitalsystems,itisnecessarytodelaythetransferofdatauntiltheoperationoftheother datahavebeencompleted,ortosynchronizethearrivalofdataat processed asubsystemwhereitis with other data.
   - Ashiftregistercanbeusedtodelaythearrivalofserialdatabya specificnumberofclock pulses, since the number of stages corresponds to the number of clock pulses required to shift each bit completely through the register.
   - Thetotaltimedelaycanbecontrolledbyadjustingtheclockfrequencyandbythenumberof stagesinthe register.
   - In practice, the clock frequency is fixed and the total delay can be adjusted only by controllingthe number of stages through which the data is passed.
2. **Serial/Parralleldataconversion:**
   - Transferofdatainparallelformismuchfasterthanthatinserialform.
   - Similarlytheprocessingofdataismuchfasterwhenallthedatabitsareavailable simultaneously.Thusindigitalsystemsinwhichspeedisimportantsotooperateondata parallel form is used.
   - Whenlargedataistobetransmittedoverlongdistances,transmittingdataonparallellinesis costlyandimpracticable.
   - Itisconvenientandeconomicaltotransmitdatainserialform,sinceserialdatatransmission requires only one line.

- Shift registers are used for converting serial data to parallel form, so that a serial input can be processed by a parallel system and for converting parallel data to serial form, so that parallel data can be transmitted serially.
- A serial in, parallel out shift register can be used to perform serial-to parallel conversion, and a parallel in, serial out shift register can be used to perform parallel- to -serial conversion.
- Auniversal shift register can be used to perform both the serial- to–parallel and parallel-to-serial data conversion.
- Abidirectionalshiftregistercanbeusedtoreversetheorderof data.

RINGANDJOHNSONCOUNTER:-

- Ringcountersareconstructedbymodifyingtheserial-in,serial-out,shiftregister.
- Therearetwotypesofring counters
  i) Basic ringcounter
  ii) Johnsoncounter
- Thebasicringcountercanbeobtainedfromaserial-inserial-outshiftregisterbyconnectingtheQ output of the last FF to the D input of the first FF.
- TheJohnsoncountercanbeobtainedfromserial-in,serial-out,shiftregisterbyconnectingtheQ output of the last FF to the D input of the first FF.
- Ringcounteroutputscanbeusedasasequenceofsynchronizing pulses.
- Thering counterisadecimalcounter.

D/AandA/DConverter

WeightedRegisterNetwork

Themostsignificantbit(MSB)resistanceisone-eighthoftheleastsignificantbit(LSB)resistance.$R_L$ismuchlargerthan 8R. The voltages V_A, V_B, $V_C$and $V_D$can be either equal to V (for logic 1) or 0 (for logical 0). Thus there are 2^4= 16 input combinations from 0000 to 1111. The output voltage V_0, given by Millman's theorem is

V_0=

Wheninputis0001,V_A=V_B= V_C=0andV_D=VandoutputisV/15.Ifinputis0010,V_A=V_B=V_D=0andV_C=Vgivingan output of2V/15. If inputis 0011,V_A= V_B=0 andV_C=V_D= V givingan output of3v/15.Thus, the output voltage varies from 0 to V in steps of V/15.

BinaryLadder Network

The weighted resistor network requires a range of resistor values. The binary ladder network requires only two resistancevalues.Fromnode1,theresistancetothedigitalsourceis2Randresistancetogroundisalso2R.Fromnode 2, the resistance to digital source is 2R and resistance to ground =

R+(2R)(2R)/(2R+2R)=2R

Thus,fromeachofthenodes1,2,3,4,theresistancetosourceandgroundis2Reach.Adigitalinput0001meansthatDis connected to V and A, B, C are grounded. The output voltage $V_0$is V/16. Thus as input varies from 0000 to 1111, the output varies from V/16 to V in steps of V/16.

Acompletedigital-to-analogconvertercircuitconsistsofanumberofladdernetworks(todealwithmorebitsofdata), operational amplifier, gates etc.

PerformanceCharacteristicsofD/A converters

TheperformancecharacteristicsofD/Aconvertersareresolution,accuracy,linearerrors,monotonicity,settingtimeand temperature sensitivity.

(a) **Resolution:**Itisthereciprocalofthenumberofdiscretestepsintheoutput.Evidentlyresolutiondepends on the number of bits.The percentage resolution is [1/ $(2^N$-1)] * 100 where N is the number of bits. The percentage resolution for different values of N is given in table.

(b) **Accuracy:** It is a measure of the difference between actual output and expected output. It is expressed as a percentageofthemaximumoutputvoltage.Ifthemaximumoutputvoltage(orfullscaledeflection)is5Vand accuracyis±0.1%,thenthemaximumerroris $\frac{0.1}{100}$ *5=0.005Vor5mV.Ideallytheaccuracyshouldbebetter than±0.5ofLSB.Inan8bitconverter,LSBis1/256or0.39%offullscale.Theaccuracyshouldbebetterthan 0.2%.

(c) **Setting Time:**When the input signal changes, it is desirable that analogoutput signal should immediately show the new output value. However in actual practice, the D/A converter takes some time to settle at the new positionoftheoutputvoltage.Settingtimeisdefinedasthetimetakenbytheconvertertosettlewith±1/2 LSB of its final value when a change in input digital signal occurs. The final time taken to settle down to new value is due to the transients and oscillations in the output voltage. Figure shows the definition of setting time.

Table

3 bit Binary word	Analog voltage
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Fig 1

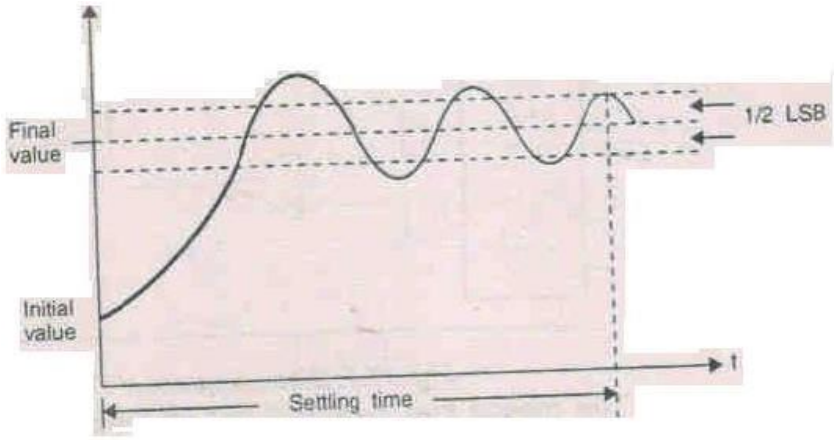
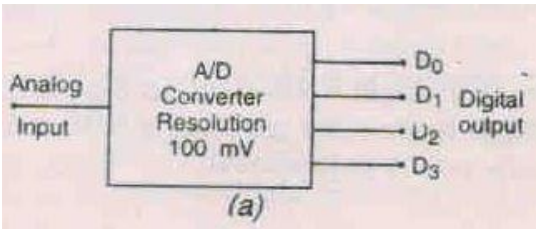


Fig. Settling time of D/A converter

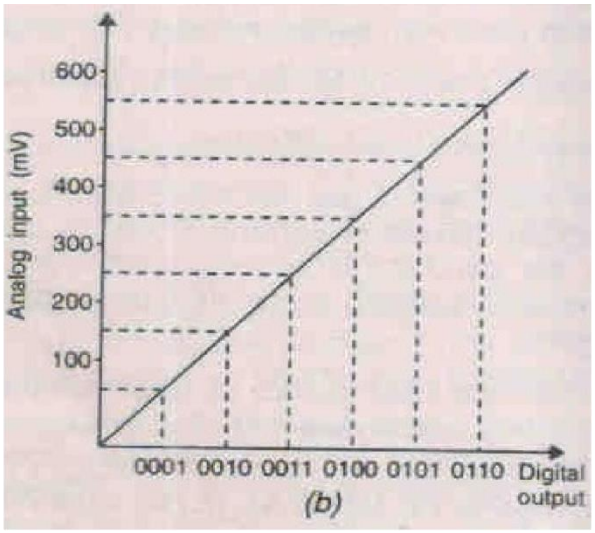
Quantization error:

An analog to digital converter changes analog signal into digital signal. It is important to note that in D/A converterthe number of input is fixed. In 4 bit D/a converter there are 16 possible inputs and in 6 bit D/A converterthereare64possibleinputs.However,inA/Dconvertertheanaloginputvoltagecanhaveanyvaluein the specified range but the digital output can have only 2^Ndiscrete levels (for N bit converter). This means that there is a certain range of input voltage which correspond to every discrete output level.

Consider a 4 bit A/D converter having a resolution of 1 count per 100 mV. Fig (b) shows the analog input and digital output. It is seen that for input voltage range of 50 mV to 150 mV, the output is same i.e. 0001, for input voltage range of 150 mV to 250 mV, the output is the same, i.e. 0010. Thus we have one digital output for each 100 mV input range. If the digital signal of 0010 is fed to a D/A converter, it will show an output of 200 V whereastheoriginalinputvoltagewasbetween150Vand250v.Thiserroriscalledquntisationerrorandinthis case this quntisation error can be ±50 mV and is equal to ±1/2 LSB.



Fig(a)A/DConverter



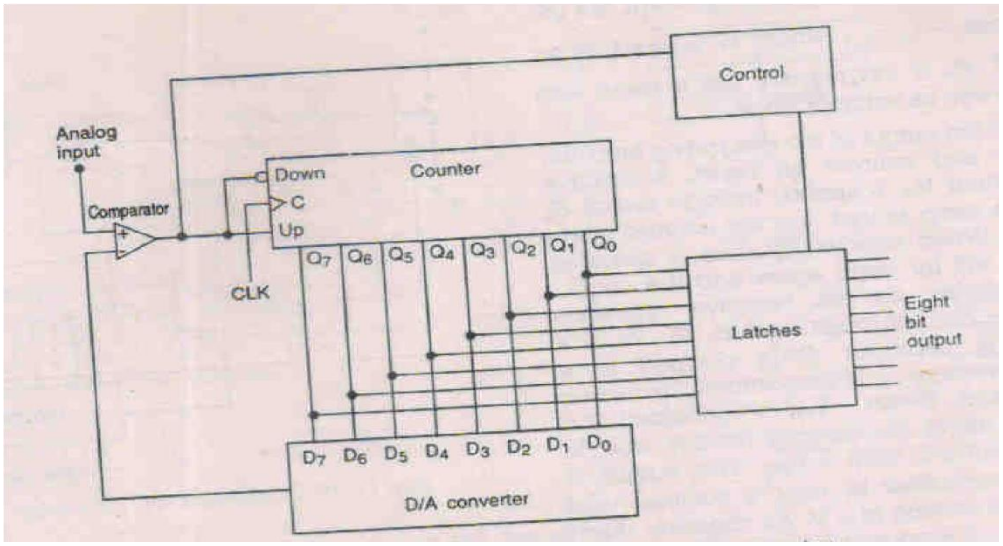
Fig(b)Quantisationerror

StairStepA/D Converter/RampA/Dconverter:

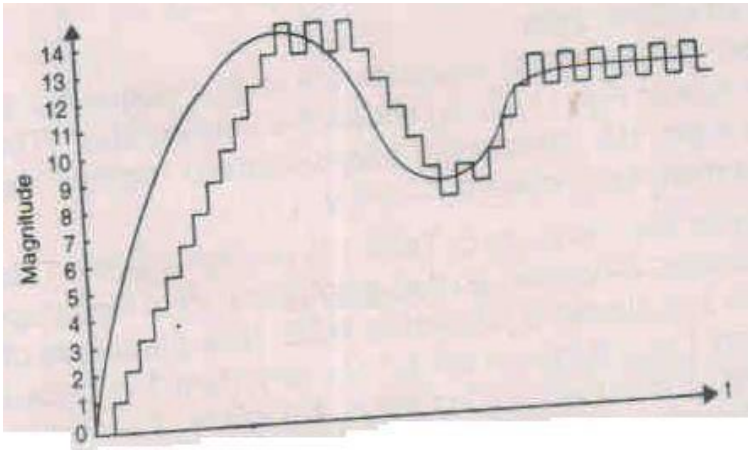
ThisconverterisalsocalleddigitalramporthecountertypeA/Dconverter.Figureshowstheconfigurationfor8 bit converter. As seen in figure it uses a D/A converter and a binary counter to produce the digital number corresponding to analog input. The main components are comparator, AND gate, D/A converter, divide by 256 counter and latches. The analog input is given to non-inverting terminal of comparator. The D/A converter provides stair step reference voltage.

Let he counter be in reset state and output of D/A converter be zero. An analog input is given to non-inverting terminal of comparator. Since the reference input is 0, the comparator gives High output and enables the AND gate. The clock pulses cause advancing of counter through its binary states and stair step reference voltage is produced from D/A converter. As the counter keeps advancing, successively higher stair step output voltage is produced. When this stair step voltage reaches the level ofanalog input voltage, the comparator output goes LowanddisablestheANDgate.Theclockpulsesarecutoffandcounterstops.Thestateofcounteratthispoint is equal to the number of steps in reference voltage at which comparison occurs. The binary number corresponding to this number of steps is the value of the analog input voltage. The control logic causes this binary number to be loaded into the latches and counter is reset.

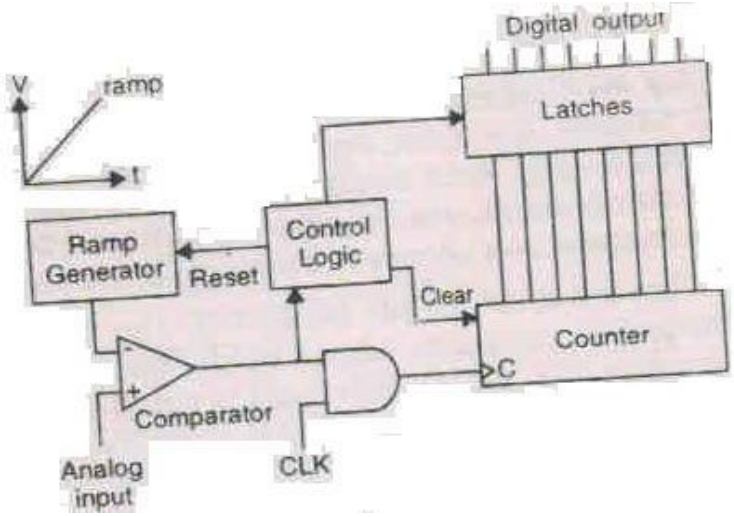
Thisconverterisratherslowinactionbecausethecounterhastopassthroughthemaximumnumberofstates before a conversion takes place. For 8 bit device this means 256 counter states.



Fig(a)8bitup-downcountertypeA/Dconverter



Fig(b)Trackingactionofupdowncountertype A/D Converter



Fig(c)SingleslopeA/Dconverter

DualslopeA/Dconverter:

ThesingleslopeA/Dconverterissuscetibletonoise.Thedualslopeconverterisfreefromthisproblem.Ituses anop-ampusedasintegretingamplifierforrampgenerator.Itisdualslopedevicebecauseitusesafixedslope ramp as well as variable slope ramp. Fig. Shows the configuration.

Itisseenthattheintegretingop-ampusesacapacitorinthefeedbackpath.

Outputvoltageofintegretingop-amp=-$\frac{1}{C}\int i dt$=-$\frac{1}{RC}\int V_{in} dt$

Thus the output voltage isintegral of analog input voltage. If$V_{in}$is constant,we getan output-$V_{in}\frac{t}{RC}$which is afixedsloperamp.IfV_{in}isvaringwegetarampwithfixedaswellasvariableslope.

Lettheoutputoftheintegretingamplifierbezeroandcounterbereset.ApositiveanaloginputV_{in}isapplied through switch S, we get a ramp output and the counter starts working. When counter reaches a specifiedcount,itwillberesetagainandthecontrollogicswitchesonthenegativereferencevoltage-V_{ref}(through switchS).AtthisinstantthecapacitorCischargedtoanegativevoltage-Vproportionaltoanaloginput voltage.When-V_{ref}isconnectedthecapacitorstartsdischarginglinearlyduetoconstantcurrentfrom-V_{ref}.

Theoutputofintegretingamplifierisnowapositivefixedsloperampstartingat– V.Ascapacitordischarges,the counter advances from the reset state. When the output of integretorbecomes zero, the comparator output

becomesLowand disables the clock signal to the AND gate. Thecounter istherefore stopped and the binary counterislatched.Thiscompletesoneconversioncycle.Thebinarycountisproportionaltoanaloginput V_{in}.

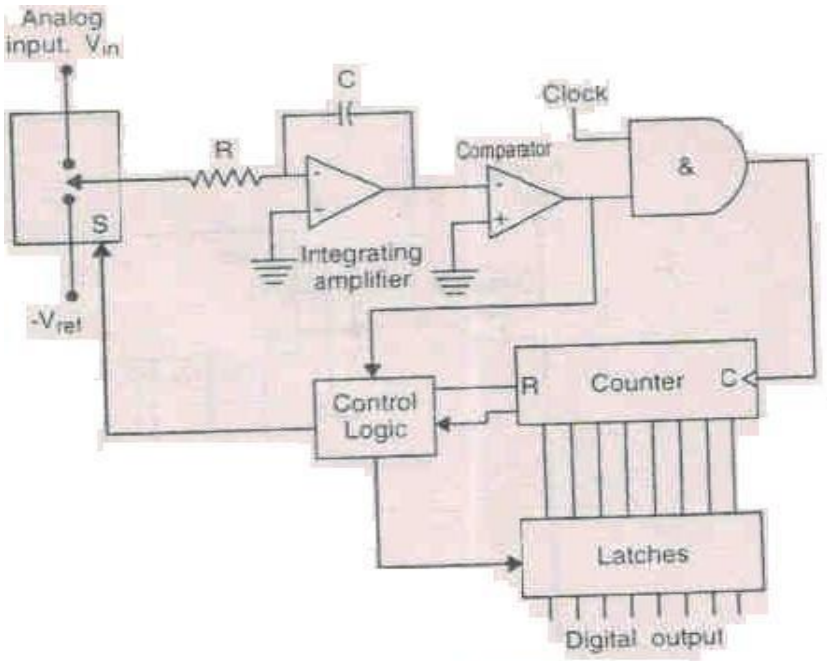


Fig. Dual slope A/D converter

SuccessiveApproximationA/DConverter:

ThisisthemostwidelyusedA/Dconverter.Asthenamesuggeststhedigitaloutputtendstowardsanalog input through successive approximations. Fig. Shows the configuration. The main components are op-amp comparator, control logic,SA (successive approximation) registerand D/A converter. As shown it is a six bit device using a maximum reference of 64 V.

Let the analog input be 26.1 v. The SA register is first set to zero. Then 1 is placed in MSB. This is fed to D/A converter whose output goes to comparator. Since the analog input (26.1 V) is less than D/A output (i.e. 32 V) the MSB is set to zero. Then 1 is placed in bit next to MSB. Now the output of D/A is 16 V. Since analog input is more than 16 V, this 1 is retained in this bit position. Next 1 is placed in third bit position. Now the D/A output is 24 V which is less than analog input. Therefore this 1 bit is retained and 1 is placed in the next bit. Now the D/A outputis 28 V, which is more than analog input. Therefore this 1 bit is set to zero and 1 is placed in 5^{th} bit position producing aD/Aoutputof 26V. Itisless thananaloginput.Thereforethis1bitisretained.Now1 isplaced inLSB producing a D/A output of 27 V which is more than analog input. Therefore LSB is set to zero and the converter gives an output of 26 V.

ThesuccessiveapproximationmethodofA/Dconverter isveryfastandtakesonlyabout250ns/ bit.

PerformanceCharacteristicsofA/Dconverters:

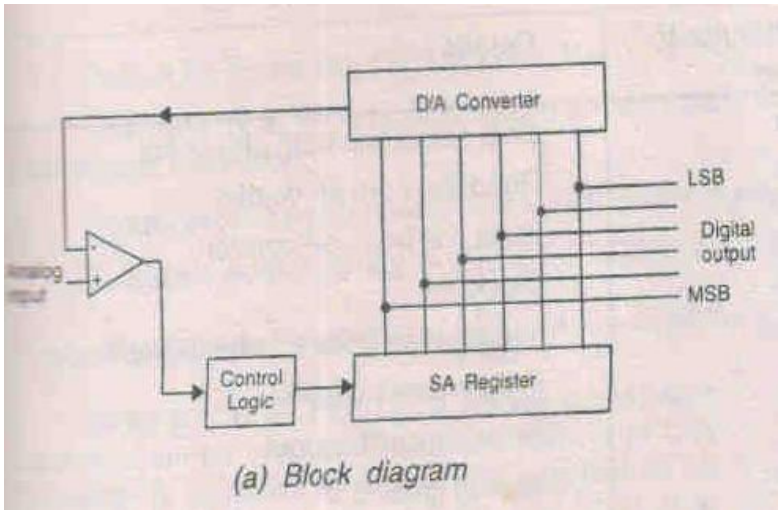
TheperformancecharacteristicsofA/Dconvertersareresolution,accuracy,A/DgainanddriftandA/Dspeed.

(a) Resolution:A/Drsolutionis thechange involtageinputnecessaryfor aonebitchangeinoutput.Itcanalso be expressed as percent.

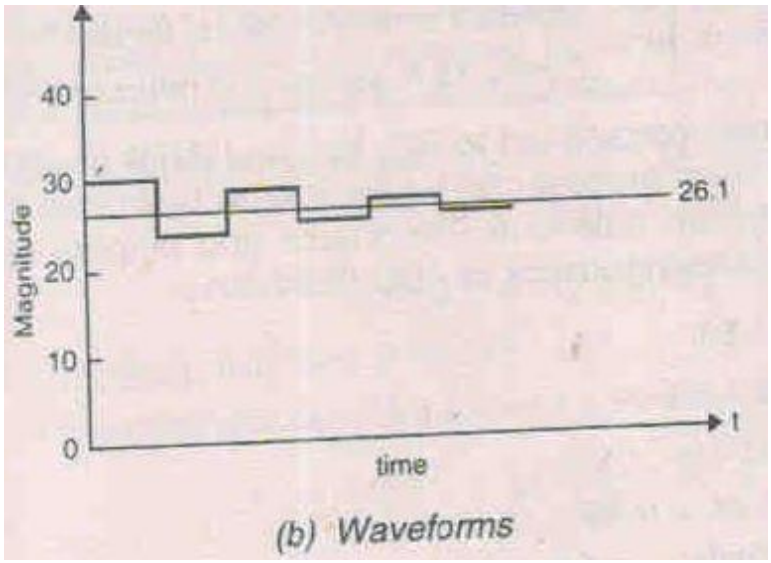
(b) A/DAccuracy:TheaccuracyofA/Dconversionislimitedbythe±1/2LSBduetoquantisationerrorandthe other errors of the system. It is defined as the maximum deviation of digital output from the ideal linear reference line. Ideally it aproaches ±1/2 LSB.

(c) A/DgainandDrift:A/Dgainisthevoltageoutput isdevidedbythe voltage inputatthelinearityreferenceline. It can usually be zeroed out.
Driftmeanschangeincircuitparameterswithtime.Drifterrorsof upto±1/2LSB willcauseamaximum errors of one LSB between the first and the last transition. Very low drift is quite difficult to achieve and increases cost of the device.

(d) A/Dspeed:Itcanbe definedintwo ways,i.e.eitherthetimenecessarytodooneconversionortheline between successive conversion at the highest rate possible. Speed depends on the settling time of components and the speed of the logic.



(a) Block diagram



(b) Waveforms