CHAPTER : 01

NUMBER SYSTEM AND CODES

Page: 1
Date: / /

✱ Number system

A number system defines a set of values used to represent quantity. the number system is used for representing the information. we use the number system in the digital system.

The digit value in the number system is calculated using

1. The digit
2. The index, where the digit is present in the number.
3. finally the base numbers, the total number of digits available in the number system.

Type of Number system

There are various types of number system used for representing information.

1. Binary Number system
2. Decimal Number system
3. Hexadecimal Number system
4. Octal Number system

संख्याओं को लिखने एवं उनके नामकरण के सुव्यवस्थित नियमों को

★ Characteristic of Number system

1. Radix / Base
2. Symbol

① Base :- Base is a maximum number of symbols that represents Number system.

1. the number of values that a digit can assume is equal to the base of the system.

For ex → for a decimal system, the base is "10" hence every digit can assume 10 values (0-9)

2. The largest value of a digit is always one less than base

③ The each digit position represents a different multiple of base is the numbers have positional importance.

Name of number system	Base
Binary	2
Decimal	10
Hexadecimal	16
Octal	8

★ Signals में तीन चीज होती है

(1) time period
(2) Frequency
(3) Magnitude

★ Number system का use digital electronic या device के साथ interaction करने के लिए किया जाता है और electrical signal के value / magnitude ज्ञात करने के लिए

(2) Symbol :- यह बताता है कि digital eletronic device कि value कहाँ से कहाँ तक assume कि जाती है

It's can assume is equal to the base of the system

Forexample ★ the octal system (the base is "10" hence symbol are (0-7)

Name of Number system	Base	Symbol
Binary	2	0 / 1
Decimal	10	0-9
Hexadecimal	16	0-9 A-F
Octal	8	0-7

1) Binary Number system

→ Generally, a binary system is used in the digital computers. It is a Computer friendly

→ A binary number system uses only two digits namely 0 and 1

→ It uses the base 2

Characteristics :

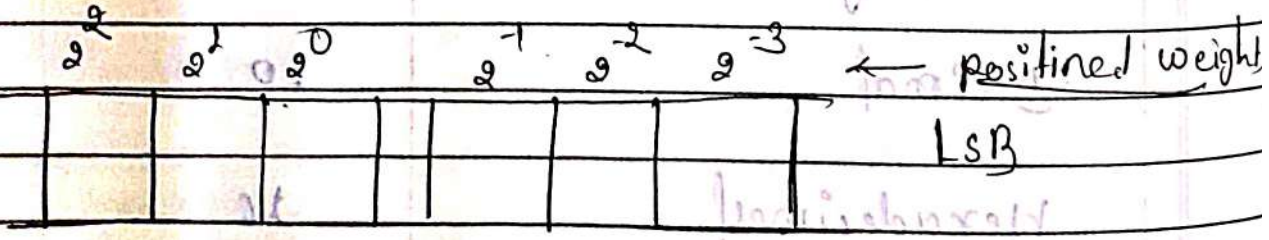
(I.) It holds only two values either 0 or 1

(II) It is also known as the base 2 number system.

(III) The position of a digit represents the 0 power of the base (2)

Example : $(101011)_2$, $(11000011)_2$, $(110)_2$ etc

★ The binary digits are also called as bits. Thus binary system is a two bit system



* MSB → Most significant Bit

* LSB → least significant Bit

MSB → The left most bit in a given binary number with the highest weight.

LSB → The right most bit in a given number with the least / lowest weight.

Decimal Number system

The decimal number are used in our day to day life. the decimal number system containing ten (10) digits from 0 to 9. It uses the base 10. its is user friendley

ex → $(108)_{10}$ $(689)_{10}$

Octal Number system

The octal Number system has base 8 (means it has only eight from 0 to 7).

Feature

1) Base the base used for octal number system is 8

2) the number of values assumed by each digit:
Each digit in the octal system will assume 8 different Values from 0 to 7.

→ The largest value of a digit :

The largest value of a digit in the octal system will be 7 means the octal no. higher than 7 will not be 8.

There are only eight possible digit value to represent a number with the help of only three bits.

Characteristics :

(1) An octal number system carries eight (8) digit starting from 0, 1, 2, 3, 4, 5, 6, 7

(2) It is also known as the base 8 number system

Octal को use शुरू में आए operating system को command देने के लिए बनाया गया था। three digit का command होता है।

(4) Hexadecimal Number system

→ Base :- The base of hexadecimal system is 16

→ Number of value assumed by each digit :-

The number of values assumed by each digit is 16. The value include digits 0 to 9 and A - F.
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

0 represents the least significant whereas F represent the most significant digit.

→ Characteristic :-

(1) It has ten digits from 0 to 9 and 6 letters from A to F

(2) The letter from A to F defines number from 10 to 15

Conversion between different number system :

(1) Decimal to Binary

→ (1) Decimal to other number system

→ (2) Other Number system to Decimal

Q. → $(105)_{10} = (?)_2$

2	105	1 ↑ LSB
2	52	0
2	26	0
2	13	1
2	6	0
2	3	1
	1	1 MSB

Binary Number = $(1101001)_2$

Thus = $(105)_{10} = (1101001)_2$ Ans

Page: 8
Date: / /

Question To check valid and invalid Number

$(11)_{10}$ → Valid Number

$(1A)_{10}$ → Invalid number

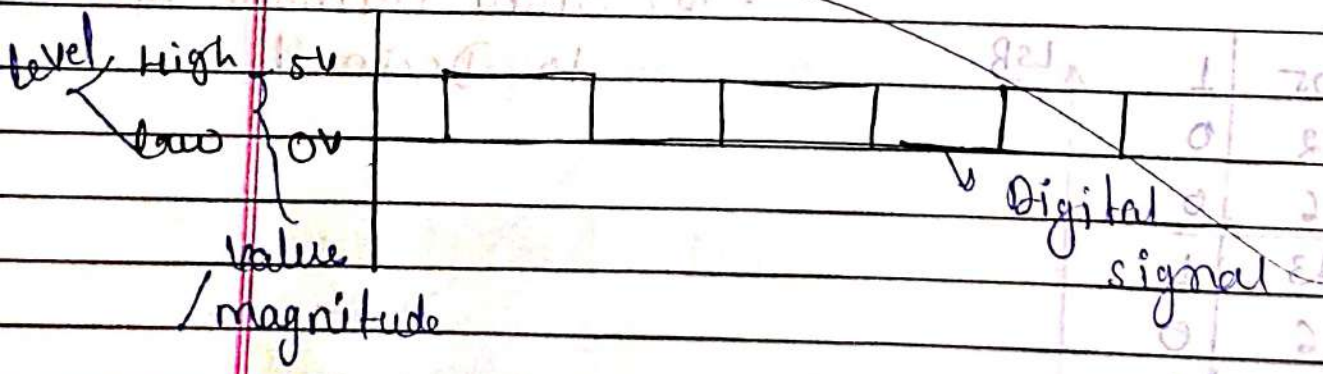
$(19)_8$ → Invalid Number

Q $(1C)_{12}$ → Invalid Number

→ Base of given Number = 12

Symbol of this Number = $(0-9)\ (A-B)$

Hence C can not be symbol of 12 Number system with base 12.



$(7N)_{16}$ → Invalid

क्योंकि यह Hexadecimal है उसका symbol $(0-9\ \ A-F)$ तक ही होता है इसलिए N, 16 base के साथ Number System में नहीं बनता।

Decimal to Binary

Q $(204)_{10} = (?)_2$

2	204	0
2	102	0
2	51	1
2	25	1
2	12	0
2	6	0
2	3	1
2	1	1
	0	

$(204)_{10} = (11001100)_2$

Decimal to ~~octal~~ Binary

$(204.75)_{10} = (?)_2$

2	204	0
2	102	0
2	51	1
2	25	1
2	12	0
2	6	0
2	3	1
	1	1

$0.75 \times 2 = 1.50$ → 1

$0.5 \times 2 = 1.0$ → 1

$(204.75)_{10} = (11001100.11)_2$

Decimal to Octal

Q $(512)_{10} = (\quad)_8$

8	512	0 ↑
8	64	0
8	8	0
	1	1

$(512)_{10} = (1000)_8$ Ans

Q $(204)_{10} = (\quad)_8$

8	204	4 ↑
8	25	1
	3	3

$(204)_{10} = (314)_8$ Ans

Q. 250

8	250	2 ↑
8	31	7
	3	3

$(250)_{10} = (372)_8$ Ans

Decimal to Hexadecimal

Q. $(259)_{10} = (\quad)_{16}$

16	259	3	↑
16	16	0	
	1	1	

$(259)_{10} = (103)_{16}$

Q. $(123)_{10} = (?)_{16}$

16	123	11	↑
	7	7	

$(123)_{10} = (7B)_{16}$

Q. $(341)_{10} = (?)_{16}$

16	341	5	↑
16	21	5	
	1	1	

$= (155)_{16}$

Convert the decimal number 35 into radix 5 equivalent

5	35	0	
5	7	2	
	1	1	

Type II decimal

Convert the decimal number to binary

Ques $(0.42)_{10} = (?)_2$

$0.42 \times 2 = 0.84$		0
$0.84 \times 2 = 1.68$		1
$0.68 \times 2 = 1.36$		1
$0.36 \times 2 = 0.72$		0
$0.72 \times 2 = 1.44$		1

$(0.42)_{10} = (0.01101)_2$

Ques → $(0.8)_{10} = (?)_2$

$0.8 \times 2 = 1.6$		1
$0.6 \times 2 = 1.2$		1
$0.2 \times 2 = 0.4$		0
$0.4 \times 2 = 0.8$		0
$0.8 \times 2 = 1.6$		1

$(0.8)_{10} = (0.11001)_2$

Quest $(0.6234)_{10} = (?)_8$

$0.6234 \times 8 = 4.9872$		4
$0.9872 \times 8 = 7.8976$		7
$0.8976 \times 8 = 7.1808$		7
$0.1808 \times 8 = 1.4464$		1

$(0.1774)_2$

~~Type III~~ Example

Conversion of mixed Decimal Number to any Other Number system

(a) $(85.63)_{10} = (?)_2$

2	85	1
2	42	0
2	21	1
2	10	0
2	5	1
2	2	0
	1	1

$0.63 \times 2 = 1.26$ → 1
$0.26 \times 2 = 0.52$ → 0
$0.52 \times 2 = 1.04$ → 1
$0.04 \times 2 = 0.08$ → 0
$0.8 \times 2 = 0.16$ → 0

$(85.63)_{10} = (1010101.10100)_2$

(b) $(3000.45)_{10}$ into its equivalent octal Number.

8	3000	0
8	375	7
8	46	6
	5	5

$0.45 \times 8 = 3.60$ → 3
$0.60 \times 8 = 4.8$ → 4
$0.8 \times 8 = 6.4$ → 6
$0.4 \times 8 = 3.2$ → 3

$(3000.45)_{10} = (5670.3463)_8$

Q Convert $(2003.31)_{10}$ into its equivalent Hex numb.

16	2003	3
16	125	13
	7	7

$0.31 \times 16 = 4.96$ → 4
$0.96 \times 16 = 15.36$ → 15
$0.36 \times 16 = 5.76$ → 5

$(7D3.4F5)_{16}$ Ans.

Q $(2005.31)_{10} = (?)_{16}$

16	2005	5
16	125	13
	7	7

$(2005)_{10} = (7D5)_2$

Decimal Fraction	× Base	= product	Recorded digit Decimal	Hex
0.31	× 16	= 4.96	4	4
0.96	× 16	= 15.36	15	F
0.36	× 16	= 5.76	5	5
0.76	× 16	= 12.16	12	C
0.16	× 16	= 2.56	2	2

$(2005.31)_{10} = (7D5.475C2)_2$

Q $(2005.31)_{10} = (?)_{16}$

16	2005	5
16	125	13
	7	7

$(2005)_{10} = (7D5)_2$

Decimal Fraction	× Base	= product	Recorded digit Decimal	Hex
0.31	× 16 =	4.96	4	4
0.96	× 16 =	15.36	15	F
0.36	× 16 =	5.76	5	5
0.76	× 16 =	12.16	12	C
0.16	× 16 =	2.56	2	2

$(2005.31)_{10} = (7D5.475C2)_2$

Type - II

To convert Other Number system to Decimal

(1) Binary to Decimal

Q. $(11101111)_2$

$= 1\times2^7 + 1\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0$

$= 128 + 64 + 32 + 16 + 0 + 4 + 2 + 1$

$= (247)_{10}$ Ans

$(11101111)_2 = (247)_{10}$

Q. $(10001111.11)_2$

$= 0\times2^7 + 0\times2^6 + 0\times2^5 + 1\times2^4 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2}$

$= 128 + 8 + 4 + 2 + 1 + 0.5 + 0.25$

$= (143.75)_{10}$ Ans

Q $(101)_2 = ()_{10}$

$= 2^2\times1 + 0\times2^1 + 2^0\times1$

$= 4 + 1$

$= (5)_{10}$

② Octal to Decimal

8 की multiple

Q. → $(756.225)_8 = (\quad)_{10}$

$= 7\times 8^2 + 5\times 8^1 + 6\times 8^0 + 2\times 8^{-1} + 2\times 8^{-2} + 5\times 8^{-3}$

$= 448 + 40 + 6 + 0.25 + 0.3125 + 0.00976$

$= 494 + 0.29101$

$= (494.29101)_{10}$

$(756.225)_8 = (494.29101)_{10}$

③ Hexadecimal to Decimal

$(7FB.16)_{16} = (\quad)_{10}$

$= 7\times 16^2 + F\times 16^1 + B\times 16^0 + 1\times 16^{-1} + 6\times 16^{-2}$

$= 7\times 64 + 15\times 16 + 11\times 1 + 0.0625 + 0.02343$

$= (2043.08593)_{10}$

$(7FB.16)_{16} = (2043.085937)_{10}$

8421

Type :- III

Binary to Octal

Decimal (base 10)	Binary (base 2)	Hexadecimal (base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Ques:- Convert the octal number $(364)_8$ into equivalent binary number

$(364)_8 = (?)_2$

Given octal number

3	6	4
011	110	100

Convert each digit to binary

$(364)_8 = (011110100)_2$

Binary to octal

Q. $(011110100)_2 = (?)_8$

Step 1 :- three pair

$= (011\ 110\ 100)_2$

$= (364)_8$ Ans

Q. $(5318)_8 = (?)_2$

5	3	1
101	011	001

$= (531)_8 = (101011001)_2$ Answer

III Hexa decimal to Binary conversion

Q $(AFB2)_{16} = (?)_2$

$(A\ F\ B\ 2)_{16}$

Given Hex no. A F B 2

Each digit converted to its four bit binary equivalent 1010 1111 1011 0010

Hence $= (AFB2)_{16} = (1010\ 1111\ 1011\ 0010)_2$

Q. $(53B.25)_{16} = (?)_2$

Given no. 5 3 B . 2 5

Convert each digit 0101 0011 1011 . 0010 0101

$(53B.25)_{16} = (0101\ 0011\ 1011 . 0010\ 0101)_2$

Q. $(1100\ 0111\ 0101 . 1100)_2 = (?)_{16}$

Ans $(C75.C)_2$

Unit 5 Octal to convert other system

IV Octal to Hex decimal Conversion

step1 ÷ Octal to convert binary

step2 ÷ binary to convert hexadecimal

Question ÷ $(436)_8 = (?)_{16}$

$(436)_8 = (?)_2$

Given no. 4 3 6

Binary equivalent 100 011 110

$(436)_8 = (100011110)_2$

Step2 $(100011110)_2 = (?)_{16}$

0001 0001 1110

$= 11E_{16}$

$(436)_8 = (11E)_{16}$ Ans

Homework

Convert the following binary numbers to decimal

(a) $(1011)_2$

$= 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 1 + 1 \times 2^0$

$= 8 + 0 + 2 + 1$

$= (11)_{10}$

$(1011)_2 = (11)_{10}$ Answer

(b) $(1101101)_2$

$= 1 \times 2^6 + 2^5 \times 1 + 2^4 \times 0 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1$

$= 64 + 32 + 8 + 4 + 1$

$= (109)_{10}$ Answer

$= (1101101)_2 = (109)_{10}$ Ans

(c) 1101.11

$= 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 + 2^{-1} \times 1 + 2^{-2} \times 1$

$= 8 + 4 + 1 + 0.5 + 0.25$

$= 13.75$ Ans

(d) $(1101110.011)_2$

$= 2^6 \times 1 + 2^5 \times 1 + 2^4 \times 0 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$

$= 64 + 32 + 8 + 4 + 2 + 0 + 0.25 + 0.125$

$= 110.375$ Ans

(3) Convert the following decimal number to binary

(a) 37

2	37	1
2	18	0
2	9	01
2	4	0
2	2	0
	1	1

$= (100101)_2$ Ans

(b) 28

2	28	0
2	14	0
2	7	1
2	3	1
	1	1

$= (11100)_2$ Ans

(c) 197.56

2	197	1
2	98	0
2	49	1
2	24	0
2	12	0
2	6	0
2	3	1
	1	1

$= (11000101)_2$ Ans

	base	product	Carry
$0.56 \times 2 =$		1.12	1
$0.12 \times 2 =$		0.24	0
$0.24 \times 2 =$		0.48	0
$0.48 \times 2 =$		0.96	0

$= (11000101.1000)_2$

(d) 205.05

2	205	1
2	102	0
2	51	1
2	25	1
2	12	0
2	6	0
2	3	1
	1	1

$= (11001101)_2$

	base	product	Carry
$0.05 \times 2 =$		0.10	0
$0.10 \times 2 =$		0.2	0
$0.2 \times 2 =$		0.4	0
$0.4 \times 2 =$		0.8	0
$0.8 \times 2 =$		1.6	1
$0.6 \times 2 =$		1.2	1

$= (11001101.000011)_2$ Ans

Hexadecimal to Octal

Q. $(3EC)_{16} = (?)_8$

$(3EC)_{16} = (\quad)_2$

3 E C

0011 1110 1100

$(3EC)_{16} = (001111101100)_2$

$= (001111101100)_2 = (?)_8$

$= (1754)_8$

Octal to Decimal

$= 1\times8^3 + 7\times8^2 + 5\times8^1 + 4\times8^0$

$= 512 + 448 + 40 + 4 = (604)_{10}$

8	604	4
8	75	5
	9	
8	1004	4
8	125	5
8	15	7
	1	1

1754

$1\times2^3 + 3\times2^2 + 5\times2 + 4$

$8 + 6 + 10 + 4 = 28$

8	28	4
8	3	3

Q. $(1AC)_{16} = (?)_8$

$(1AC)_{16} = (?)_8$

1 (0001) A(1010) C(1100)

$(AC)_{16} = (000110101100)_2$

$(000110101100)_2 = (?)_8$

$= (6574)_8$ Answer:

octal to Decimal

$6 \times 8^2 + 5 \times 8^1 + 8^0 \times 4$

$= (428)_{10}$

Decimal to octal

2	428	0
2	64	0
2	32	0
2	16	0
2	8	0
2	4	2
2	2	1

Q.1 Convert octal number $(2567)_8$ into binary

: $(2567)_8 = (?)_2$

$(010\ 101\ 110\ 111)_2$

$= (010101110111)_2$ Ans

Convert decimal number $(205.05)_{10}$ into binary number

2	205	1
2	102	0
2	51	1
2	25	1
2	12	0
2	6	0
2	3	1
	1	1

$0.05 \times 2 = 0.10$ 0
$0.10 \times 2 = 0.20$ 0
$0.20 \times 2 = 0.40$ 0
$0.40 \times 2 = 0.80$ 0
$0.80 \times 2 = 1.60$ 1
$0.60 \times 2 = 1.20$ 1

$(11001101)_2$ $(.000011)_2$

$= (11001101.000011)_2$

$1 \times 2^7 + 1 \times 2^6 + 0 + 0 + 2^3 \times 1 + 2^2 \times 1 + 0 \times 2^1 + 2^0 \times 1 + 0 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 2^{-4} \times 1 + 2^{-5} \times 1$

$128 + 64 + 8 + 4 + 1 + 0.0625 + 0.03125$

$= 205$

Convert hexadecimal number $(BC0.0E)_{16}$ into octal number

$(BC0.0E)_{16} = (?)_8$

$(BC0.0E)_{16} = (?)_2$

$(1011\ 1100\ 0000 . 0000\ 1110)_2$

$= (BC0.0E)_{16} = (1011\ 1100\ 0000 . 0000\ 1110)_2$

$= (101\ 111\ 000\ 000 . 000\ 011\ 10)_2 = (?)_8$

$= (5700.034)_8$

Convert decimal number $(807)_{10}$ into BCD

16	807	7
16	50	2
1	3	3

$(327)_{16}$

Number system

Number	Decimal	Binary	Octal	Hexadeci
Decimal	Decimal	divided by 2	divided by 8	devided by 16
Binary	multiple +ve & -ve power by 2	same multiple +ve & -ve power by 8	multiple +ve & -ve power by 8	multiple +ve & -ve Power by
Octal	multiple +ve & -ve power by 8	octal to Decimal Decimal to Binary	same	Octal to Decimal Decimal to Hexadecima
Hexa-decimal	multiply +ve & -ve power by 16	Hexadecimal to binary ~~to~~ binary ~~to~~ H → D → B	Hexadecimal to binary to Octal	same

r's & (r-1)'s complement (r = base)

→ Decimal Number system → 10's & 9's complement

→ Binary Number system → 2's & 1's complement

→ Octal Number system → 8's & 7's Complement

→ Hexadecimal Number system → 16's & 15's complement
या
इसे 10's complement भी कहते है क्योंकि 15 होता है
+1
10

Rule :

1) 9's में change करते समय जो भी number system होगा उसके higher symbol से जितने digit होंगे उतने बाड़ा लिखकर उससे minus करना है।
(27) = 99
−27
72
हर digit या bit को higher symbol से लिखकर minus करना।

2) 10's complement के लिए 9's complement के Ans के LSD में +1 add करना है।

Decimal
+054.14
099.99
− 54.14
45.85

2's
45.85
+1
45.86

(1) Decimal Number system 10's & 9's Complement :-

Q. 9's complement of 27

$$\begin{array}{r} 99 \\ -27 \\ \hline 72 \end{array}$$

10's Complement of $\begin{array}{r} 72 \\ +1 \\ \hline 73 \end{array}$

Q. 9's Complement of 1054.073

$$\begin{array}{r} 9999.999 \\ -1054.073 \\ \hline 8945.926 \end{array}$$

8945.926

10's complement of

$$\begin{array}{r} 8945.926 \\ +1 \\ \hline 8945.927 \end{array}$$ Ans

Digit	Complement
0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0

Conversion

Complement

Complement are used in digital computer to simplify the subtraction operation and for logical manipulation simplifying operation leads to simpler less expensive circuits to implement the operation.

There are Two types of complements for each base - r system

1) the redix complement
2) diminished redix complement

The first is refer to as the r's complement and second is refer to as the (r-1)'s complement

when the value of the base r is substitute in the name the two types are referred to as the r's complement and (r-1)'s Complement. for binary numbers. and 10's and 9's Complement for decimal number

Binary System 2's & 1's complement

Q. 1's complement of 10110010.101

higher symbol of Binary is 1

```
  11111111.111
- 10110010.101
  01001101.010   Ans
```

Bits	Com
0	1
1	0

```
  10101101
  01010010
```

01001101.010

2's Complement of 01001101.010

```
  01001101.010
+             1
  01001101.011   Ans
```

Q. 1's complemen to 10101111.10

```
  11111111.11
- 10101111.10
  01010000.01
```

= 10100000.01 Ans

2's complement of (1)

```
  10100000.01   ; 1+1 = 10 (2)
+             1
  10100000.10
```

Q) Octal Number system 8's & 7's Complement

7's complement of 627.45

Digen symbol → 777.77
of octal 627.45
150.32 150.32 Ans

8's complement of
150.32
+1
150.33 Ans

Q. 7's complement of 512.221

777.77
512.21
265.56 Ans

8's complement of
265.56
+1
265.57

* if in 8's complement का LSD पर 1 add करने पर
8 आए तो 8 नहीं लिखना है 9 से
to लिखना है

265.57
+1
.60

Hexadecimal number system 16's & 15's Complement

Q. 15's complent of 9BC.74

15 15 15 15 15
F FF . FF
9 BC . 7A
12 12
643.85

16's complent of 643.85

643.85
+1
643.86

Q. 15's complement 7BD.EA

F FF . FF
- 7 B D . E A
8 42 . 15

16's complement

842.15
+1
842.16

symbol	Compl
0	F
1	E
2	D
3	C
4	B
5	A
6	9
7	8
8	7
9	6
A	5
B	4
C	3
D	2
E	1
F	0

Arithmetic

Binary Arithmatic

Addition substraction multiplication division

I Addition Rule Binary

For an case of binary addition

	A		B	Addition
Case 1	0	+	0	0
Case 2	0	+	1	1
case 3	1	+	0	1
case 4	1	+	1	10

↳ Carry

+ • = • • = $(10)_2$

Rule of binary addition

Rule	A		B	Sum	Carry	
1	0	+	0	0	0	
2	0	+	1	1	0	
3	1	+	0	1	0	
4	1	+	1	①	1	
5.	1	+ 1 +	1	1	1	$(3)_{10} = (11)_2$

1 + 1 = 0 with carry 1

1 + 1 + 1 = 1 with carry 1

Example :-

(i) 1111 + 1000

```
  1111
+ 1000
-------
 10111
```

Ans

(ii) 11010001 + 00110001

```
 (1)(1)(1)
  11010001 + 00
+ 00110001
----------
 100000010
```

100000010 Ans

Q. Add the following binary number

A = (10111)$_2$
B = (11001)$_2$

```
 (1)(1)(1)(1)
  10111
- 11001
-------
  10000
```

10000 Ans

Q. 011 + 101 =

```
  011        3
+ 101      + 5
-------    ---
 1000        8
```

bar answer check

Q. 1011.011 = A
1100.101 = B

11000.000

A check करने के लिए A का decimal no और B का Decimal no. आ होगा को Add करके check the ans decimal no is similar.

$A = (1011.011)_2 = (?)_{10}$

$= 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-3}$

$= 8 + 2 + 1 + 0.25 + 0.125 = 11.375$

$B = (1100.101)_2 = (?)_{10}$

$= 1\times2^3 + 1\times2^2 + 0\times2^1 + 0\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

$= 8 + 4 + 0.5 + 0.125$

$= 12.625 =$

A + B
11.375 + 12.625 = 24

Ans. → 11000.000

$= 2^4 + 2^3 + 0 + 0 + 0 = 16 + 8 = 24$ Ans

Binary subtraction

Four basic rules for binary subtraction ::

Case	A B	Substraction	Borrow
1	0 - 0	0	0
2	1 - 0	1	0
3	1 - 1	0	0
4	0 - 1	1	1
	10 - 1	1	

example :(2)

A = $(11011)_2$ and B $(10110)_2$ to substract

$(10)_2 = 2$

```
  11011
- 10110
  00101
```

Thus the subtraction is = 00101 Ans.

Q. 10111.1 - 1001.1

```
  10111.1
- 1001.1
  01000.0
```

Ans

Convert $(38)_{10}$ and $(29)_{10}$ into their binary equivalents

2	38	0
2	19	1
2	9	1
2	4	0
2	2	0
	1	1

2	29	1
2	14	0
2	7	1
2	3	1
	1	1

$(38)_{10} = (100110)_2$

$(29)_{10} = (11101)_2$

A = 100110

B = 11101

Column 5 (Borrow) Column 4

$$A - B = \begin{array}{r} 100110 \\ -\ 011101 \\ \hline 001001 \end{array}$$

Column → 4 : — 10 - 1 = 1

Column → 5 : — 10 - 1 - 1 = 0

2 - 1 - 1 = 0

Addition of binary number system :-

ex → 1001 तथा 0011 का योग

```
  1001
+ 0011
------
  1100   Ans
```

eg : 1001 तथा 1011

```
  1001
+ 1011
------
 10100   Ans
```

eg. 11001 तथा 10101

```
  11001
+ 10101
-------
 110000   Ans
```

eg : 11111 तथा 010101

```
  11111
+ 010101
--------
  10100   Ans.
```

Q 11010 तथा 11100

```
  11010
+ 11100
-------
 110110   Ans.
```

Subtraction of binary number system

eg: $(1011)_2 - (0101)_2$

$$\begin{array}{r} 1011 \\ -\ 0101 \\ \hline 0110 \end{array}$$ Ans.

eg: $1100.100 - 111.101$

$$\begin{array}{r} 1100.100 \\ -\ 111.101 \\ \hline 100.111 \end{array}$$ Ans

eg: $1100 - 1010$

$$\begin{array}{r} 1100 \\ -\ 1010 \\ \hline 0010 \end{array}$$ Ans

eg. $100 - 110.01$

$$\begin{array}{r} 100.00 \\ -\ 010.01 \\ \hline 010.11 \end{array}$$ Ans

~~Decimal~~ Binary

1's Complement Number

किसी भी बाइनरी संख्या के सभी 0 को 1 से और सभी 1 को 0 से बदल देने पर 1's compliment प्राप्त होगा। | 0 → 1
या 1 → 0
1's Compliment के लिए binary के maximum symbol 1 से बताया जाता है |

ex → 10011 ⇒ 11111
10011
01100 Ans

2's Compliment Number

किसी संख्या के 1's complement में 1 जोड़ने पर प्राप्त संख्या को उस संख्या का 2's compliment कहते है |

2's compliment = 1's Compliment + 1

ex → 10110

1's Complement = 01001 |

2's Complement = 01001
+1
01010 Ans

Page: 24
Date: / /

Octal Decimal

9's Compliment Number

किसी Decimal संख्या का 9's compliment प्राप्त करने के लिए उसके प्रत्येक अंक को 9 से घटाया जाता है। 9 से इसलिए क्योंकि 9 maximum symbol है Decimal का

eg → 215 का 9's Complement

```
  999
- 215
  784   Ans
```

10's Compliment Number

किसी संख्या के 9's complement में 1 जोड़ने पर प्राप्त संख्या को उस संख्या का 10's complement कहते हैं।

10's complement = 9's complement + 1

415 का 10's complement

```
  999
- 415
  584   9's complement
```

10's complement = का 584 + 1

= 585 Ans.

signed form and unsigned form

signed form → वह संख्या जो positive और Negative भी

eg:- 5, +5, -5, 6, -2

Unsigned form → वह संख्या जो केवल positive भी हो

eg:- 6, 7, 8, 9

signed form → इसमें हमें computer को बताना पड़ेगा कि वह positive है या Negative. चूँकि computer binary समझता है तो बताता है

1 → Negative (-) 0 → Possitive (+)

-5 = 1101 eg. (1 = sign)

signed form तीन प्रकार के होते है

(1) sign magnetued form :-

इसमें positive same होगा Negative के लिए use करेंगे।

13 → 01101 (0 = sign (+)) यहाँ 13 के आगे कुछ नहीं मतलब + इसलिए 0 लगाएं।

-5 → 1101 (1 = sign (-))

② 1's Complement signed form

इसमें positive का तो same निकालते है लेकिन Negative का पहले 1's Complement निकालते है फिर 1 या 0 positive और Negative के बिगाव से लगाते है।

13 → 01101 (Positive same s.mf के जैसे होगा)

-5 → पहले 5 का 1's complement

+5 = 0101 | -5 = 1010 | या 5 = 101
Complement = 1010 = -5 | Comple = 010
-4 → ~~पहले~~ | -5 = 1010
+4 → 0100
Complem -4 → 1011 Ans.

③ 2's Complement signed form

इसमें भी positive same होगा लेकिन Negative के लिए पहले उस संख्या का 2's comp लगा निकालेंगे।

13 → 01101
-5 → ~~1011~~ | 5 = 101 | या 5 = 0101
-5 = 1011 | 2's Comp = 010 | -5 = 1010
sign के लिए 1 लगा देंगे | +1 | 1011
011 | Ans.

Homework

2.4 Add the following binary number :

(a) 11011 + 1101 = 101000

```
  11011
+  1101
------
 101000
```

Ans = 101000

(b) 1011 + 1101 + 1001 + 1111 = 110000 Ans

```
  1011          11000          100001
+ 1101        +  1001        +  1111
-----         ------         ------
 11000         100001         110000  Ans
```

Ans → 110000

(c) 10111.101 + 110111.01 = 1001110.111

```
  10111.101
+ 110111.01
-----------
 1001110.111
```

Ans → 1001110.111

(d) 1010.11 + 1101.10 + 1001.11 + 1111.11 = 11000.11

1010.11	11000.01	100010.00
+ 1101.10	+ 1001.11	+ 1111.11
11000.01	100010.00	110001.11

Ans = 110001.11

(2.5) Subtract the following binary numbers :-

(a) 1011 - 101 = 0110

```
  1011
- 101
------
  0110   Ans
```

(b) 10110 - 1011 = 01011 Ans

```
  10110
- 1011
-------
  01011   Ans
```

(c) 1100.10 - 111.01 = 0101.01

```
  1100.10
- 111.01
---------
  0101.01   Ans
```

(d) 10001.01 - 1111.11 = 00001.10

```
  10001.01
- 1111.11
----------
  00001.10   Ans
```

* पूँकि question में 12 bit दिया है अतिरिक्त बगल में Answer भी 12 bit का होना चाहिए उसके लिए हमें कितने 0 और - है तो 1 कब तक लगाना है जब तक bit 12 नहीं आ जाए

2.10 Find the 12 bit 2's complement form of following decimal number.

(a) -37

Binary form → $(100101)_2$

2	37	1
2	18	0
2	9	1
2	4	0
2	2	0
	1	1

: 100101 = 37

: 1100101 = -37 (-)

+37 = $(100101)_2$

2's Complement = 011010 (-37 का 2's complement)

+1

011011 Ans

Ans → 111111011011 Ans

2's complement form of -011011 Ans

यहाँ (-) है अतिरिक्त 1

(b) -173

2	173	1
2	86	0
2	43	1
2	21	1
2	10	0
2	5	1
2	2	0
	1	1

= $(10101101)_2$ = 173

= -173 = 110101101

+173 = 10101101

2's Complement = 01010010 + 1 = 01010011

-173 of 2's complement = 10101011 Ans

2's complement form of 01010011 Ans.

(c) -65.5

2	65	1
2	32	0
2	16	0
2	8	0
2	4	0
2	0	0
	1	1

$= (1000001)_2$

$.5 \times 2 = 1.0 \quad 1$

$= (1000001.1)_2$

$(1000001.1)_2$

1s ↓

(0111110.0)

2s ↓

(0111110.1)

$\downarrow$ Ans

$(1111,0111110.1)$ Ans

65.5 = 1000001.1

-65.5 = 11000001.1

2's complement → 00111110.0 + 1 = 00111110.1

65.5 of 2's complement form of 00111110.1 Ans

-65.5 ⇒ 2's complement 10111110.1 Ans

$= (1111,0111110.1)_2$ Ans

(d) -197.5 Decimal

2	197	1
2	98	0
2	49	1
2	24	0
2	12	0
2	6	0
2	3	1
	1	1

$(11000101)_2$

$.5 \times 2 = 1.0$

$(11000101.1)_2$ Binary

$197.5 = 11000101.1$

$+197.5$
$+197 = 011000101.1$) 1's complement

2's complement $= 100111010.0$

$+1$) 2's complement

100111010.1 Ans

2's complement of 100111010.1 Ans

$\Rightarrow (11100111010.1)_2$ Ans

(2.11) find the 12-bit 1's complement form of the following decimal number

(a) -97

2	97	1
2	48	0
2	24	0
2	12	0
2	6	0
2	3	0
	1	1

$= (1000001)_2$

+97

01000001

↓ 1's com

10111110

↓ 2's

10111111

97 = 1000001

+97 = 01000001

2's complement → 10111110

-97 = 10100000 1 (2's com)

= Ans = 11101000001

(b) −224

2	224	0
2	112	0
2	56	0
2	28	0
2	14	0
2	7	1
2	3	1
	1	1

$(11100000)_2$

+224

↓

$(011100000)_2$

↓ 1's

10001111

↓ Ans

(1111000011111)

$(11100000)_2 = 224$

+224 = 011100000

1's complement = 100011111 = Any

= 1111000011111

2.14 Represent the following decimal number in 8 bit.

(i) sign magnitude form
(ii) sign 1's complement
(iii) sign 2's complement

Qu→(a) +14

binary form: 1110

Question में 8 bit के लिए (+) के आगे 0 लगाते

sign magnitude form = 01110 = 00001110

sign 1's complement = 01110 = 00001110

sign 2's complement = 01110 = 00001110

2.15 Convert the following octal number to hexadecimal

(a) $(256)_8 = (?)_2$

2 5 6
010 101 110

$256 = (0b\,\underline{00}\,\underline{10}\,\underline{10}\,\underline{1}\,\underline{110})_2 = (?)_{16}$

$(0AE)_{16}$ Ans

(b) $(0035)_8 = (?)_{16}$

$(0035)_8 = (?)_2$

0 0 3 5

010 000 011 101 $= (010000011101)_2$

$(010000011101)_2 = (?)_{16}$

(010000011101)

$= 41D$ Answer

(c) $(762.46)_8 = (?)_{16}$

Octal to binary

$(762.46)_8 = (?)_2$

762 . 46

$(001111110010 . 100110)_2 = (?)_{16}$

$= (001111110010 . 10011000)_2$

$= 3F2.98$

(d) $(6054.963)_8 = (?)_{16}$

$(6054.963)_8 = (?)_2$

$= (110\,000\,101\,100 . 010\,110\,011)_2 = (?)_{16}$

$= (1100\,0010\,1100 . 0101\,1001\,1000)_2$

$= (C2C.598)_{16}$ Ans.

2.16 Convert the following hexadecimal number to octal

(a) $(2AB)_{16} = (?)_8$

$(2AB)_{16} = (?)_2$

$= (0010\,1010\,1011)_2 = (?)_8$

$= (1253)_8$ Ans.

(b) $(42FD)_{16} = (?)_8$

$= (0100\,0010\,1111\,1101)_2 = (?)_8$

$= (041375)_8$ Ans

(c) $(4F7 \cdot A8)_{16} = (?)_8$

$= (0100\,1111\,0111 \cdot 1010\,1000)_2$

$= (2367 \cdot 520)_8$ Ans.

(d) $(BC70 \cdot 0E)_{16}$

$= (1011\,1100\,0111\,0000 \cdot 0000\,1110)_2$

$= (136160 \cdot 034)_8$ = Ans.

2.17 Convert the following octal number to decimal

Q. $(463)_8$

$= 8^2 \times 4 + 8^1 \times 6 + 8^0 \times 3$

$= (307)_{10}$ Ans.

Q. $(2056)_8$

$= 8^3 \times 2 + 8^2 \times 0 + 8^1 \times 5 + 8^0 \times 6$

$= 1070$ Ans

Q. $(2054.64)_8$

$= 8^3 \times 2 + 8^2 \times 0 + 8^1 \times 5 + 8^0 \times 4 + 8^{-1} \times 6 + 8^{-2} \times 4$

$= (1068.8125)_{10}$ Ans

Q. $(6534.04)_8$

$= 8^3 \times 6 + 8^2 \times 5 + 8^1 \times 3 + 8^0 \times 4 + 0 \times 8^{-1} + 8^{-2} \times 4$

$= 3420.0625$ Ans

2.18 Convert the following decimal numbers to octal

(a) $(287)_{10} = (?)_8$

2	287	1
2	143	1
2	71	1
2	35	1
2	17	1
2	8	0
2	4	0
2	2	0
	1	1

$(100011111)_2$

$= (437)_8$

8	287	7
8	35	3
	4	4

$= (437)_8$

(b) $(3956)_{10}$

8	3956	4
8	494	6
8	61	5
	7	7

$(7564)_8$ Ans

(c) 420.6

8	420	4
8	52	4
	6	6

$(644)_8$ Ans

$0.6 \times 8 = 4.8$ 4
$0.8 \times 8 = 6.4$ 6
$0.4 \times 8 = 3.2$ 3

$(644.463)_8$

(d) 8476.47

8	8476	4
8	1059	3
8	132	4
8	16	0
	2	2

$(20434)_8$

$0.47 \times 8 = 3.76$ 3
$0.76 \times 8 = 5.72$ 5
$0.72 \times 8 = 5.76$ 5

$(20434.355)_8$ Ans

2.24 Convert the following hexadecimal number to binary

(a) $(C20)_{16} = (?)_2$

$= (1100\ 0010\ 0000)_2$ Ans.

(b) $(F297)_{16} = (?)_2$

$= (1111\ 0010\ 1001\ 0111)_2$ Ans

(c) $(AF9.B00)_{16} = (?)_2$

$= (1010\ 1111\ 1001.1011\ 0000\ 1101)_2$ Ans

(d) $E79A.6A4$

$= (1110\ 0111\ 1001\ 1010.0110\ 1010\ 0100)_2$ Ans

2.25 Convert the following binary to hexadecimal

(a) $(10110)_2 = (?)_{16}$

$= (0001\ 0110)$

$= (16)_{16}$ Ans

(b) $(1011011011)_2 = (?)_{16}$

$= (\underline{0010}\,\underline{1101}\,\underline{1011})$

$= (2DB)_{16}$ Ans

(c) $(111011011.01111)_2 = (?)_{16}$

$\underline{000}\,\underline{111}\,\underline{011}\,\underline{011}.\,\underline{0111}\,\underline{1000}$

$(1B7.78)_{16}$ Ans

(d) $(1101101101101.10110)_2 = (?)_{16}$

$(\underline{0001}\,\underline{1011}\,\underline{0110}\,\underline{1101}.\,\underline{1011}\,\underline{0100})_2$

$= (1B6D.B4)_{16}$ Ans

2.26 Convert the following hexadecimal to decimal

(a) ABC

$= 16^2 \times A + 16^1 \times B + 16^0 \times C$

$= 2748$ Ans

(b) 2EB7

$= 2 \times 16^3 + E \times 16^2 + B \times 16^1 \times 16^0 \times 7$

$= 11959$ Ans

(c) A08F.EA

$= A \times 16^3 + 0 \times 16^2 + 8 \times 16^1 + 16^0 \times F + 16^{-1} \times E + 16^{-2} \times A$

$= (41,103.9141)_{10}$ Ans

(d) ~~BEA7~~ 8E47.AB

$= 16^3 \times 8 + 16^2 \times E + 16^1 \times 4 + 16^0 \times 7 + 16^{-1} \times A + 16^{-2} \times B$

$= 36423.668$

2.11

Q. find the 12 bit 1's complement form of the following decimal number

(c) -905.75

2	905	1
2	452	0
2	226	0
2	113	1
2	56	1
2	28	0
2	14	0
2	7	1
	3	1
	1	1

$(1100\,1101)$

$.75 \times 2 = 1.50$ 1

$.50 \times 2 = 1.0$ 1

$905.75 = (1100\,1101.11)_2$

905.75 of 1's complement = 0011 0001 0.00

-905.75 of 1's complement = 1100 1110 1.00

(d) -29.375

2	29	1
2	14	0
2	7	1
2	3	1
	1	1

(11101)

$.375 \times 2 = 0.750$ — 0

$.75 \times 2 = 1.50$ — 1

$.50 \times 2 = 1.0$ — 1

12 करने के लिए 1 लगाए

$29.375 = (11101.011)_2$

29.375 का 1's Complement = 00010.100

-29.375 का Complement = 11101.100 Ans

Ans → 11110001.100 Ans

2.14) 8 bit

(i) sign magnetude (2) sign 1's complement
(3) 2's complement

2.(b) $+27$

2	27	1
2	13	1
2	6	0
2	3	1
	1	1

$= (11011)_2$

8 bit चाहिए
8 bit में
+ = 0, - = 1

sign magnetude : 11011 = 00011011

sign 1's magnetude = 11011 = 00011011

Sign 2's magnetude = 11011 = 00011011

A positive का magnetude same

(c) +45

2	45	1
2	22	0
2	11	1
2	5	1
2	2	0
	1	1

(101101)

Question में 8 bit है तो
उत्तर 8 bit होना चाहिए

Sign magned : 101101 = 00101101 Ans

Sign 1's complement = 101101 = 00101101 Ans

Sign 2's complement = 101101 = 00101101 Ans

(d) -17

2	17	1
2	8	0
2	4	0
2	2	0
	1	1

: 10001

अगर 8 bit है तो अगर (-) है तो 1 लगाते
आठ बनाने में

-17 का Sign magned = 110001 = 11110001

Sign 1's Complement +17 = 010001

= 101110 = 11101110

Sign 2's Complement : 101110
+1

अगर Question में 8 bit दिया हो तो 8 bit बनाना है अगर + है तो 0 लगाकर अगर - है तो 1 लगाकर

(e) -37

2	37	1
2	18	0
2	9	1
2	4	0
2	2	0
	1	1

$= (100101)$

7 bit → 8 bit

-37 का sign magnetude = 1100101 = 11100101

+37 का sign 1's complement = 0100101 = 11011010

= 10110101 Ans.

+1

-37 का sign 2's = 10110111 Ans.

= 11011011

(f) -76

2	76	0
2	38	0
2	19	1
2	9	1
2	4	0
2	2	0
	1	1

$= 1001100$

Signed magnitude from -76 = 1001100

= 11001100

sign 1's magnitude from -76 = 10110011

↳ = 11011011 Ans

+1

sign 2's magnitude from -76 = 10110100

1011100

= 11011100 Ans

★ Subtraction को easy करने के लिए Complement का use करते है।

★ positive number को को represent करने के लिए sign form में उसका magnetude या equvalen binary लिखेंगे और ⊕ MSB में 0 जोड़ देंगे।

eg
उदा.

+14 = 00001110 (sing magnetude और sign 1's & 2's sing same)

Binary Multiplication

	A B	Multiplication
Case 1	0 X 0	0
Case 2	1 X 0	0
Case 3	0 X 1	0
Case 4	1 X 1	1

Question

```
  1101          1101
x 101         x 101
  1101          1101
 0000x         0000
1101x         1101
1000001       1000001  Ans
```

Ans

2.8 Multiply the following binary number by the computer method

(a) 1011 X 11

```
  1011
x   11
  1011
 1011x
100001
```

Ans ÷ 100001

(b) An 1001 x 101

```
  1001
x  101
  1001
 0000x
1001x
101101
```

Ans ÷ 101101

(c) 1100 x 110

```
  1100
x  110
  0000
 1100x
1100x
1001000
```

Ans 1001000

(d) 1111 x 100

```
  1111
x  100
  0000
 0000x
1111x
1111100
```

Ans → 1111100

Binary division ÷

1011

	A		B	Division
Case 1	0	÷	1	0
Case 2	1	÷	1	1
Case 3	0	÷	0	0
Case 4				

eg: 101101 ÷ 110

```
110 ) 101101 ( 111.1
    - 110
      1010
      - 110
        1101
      - 1110
        00110
        - 110
          000
```

Ans ÷ 111.1

Answer check :

$101101 = 2^5 \times 1 + 2^4 \times 0 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1$

$= 32 + 0 + 8 + 4 + 1 = 45$

$110 = 2^2 \times 1 + 2^1 \times 1 + 0 = 6$

$\frac{45}{6} = 7.5$

$111.1 = 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 1 + 2^{-1} \times 1$

$= 4 + 2 + 1 + .5$

$= 7.5$ match Ans is right

Homework

2.7

Divide the following binary numbers:

(a) 1010 by 11

```
11) 1010 (11.0101
  - 11
    100
  -  11
    0010
   - 00
     100
```

```
    100
  - 11
    010
```

1010 = A(10)

11 = 3

$\frac{10}{3} = 3.33$

0.5+0.25

$11.01 = 3 + 2^{-1}\times0 + 2^{-2}\times1 + 2^{-3}\times0 + 2^{-4}\times1$

$= 3 + 0.25 + 0.625$

$= 3.875$

1010 divide by 11 ~~so~~ Ans → 11.0101

(b) 1110 by 101

```
101) 1110 (110
   - 101
     0101
   -  101
      0000
    - 0000
      ----
        0
```

Ans → 110

(c) 11011 by 10.1

11011 = 11110
10.1 101

101) 11110 (110	10.1) 11011 (11
- 101	- 10.1
0111	10.1
- 101	- 10.1
0101	0.0
- 101	
0000	
- 0000	
0	

Ans = 110

101) 11110 (1010.1100
- 101
011
- 00
111
- 101
100
- 000
1000
+ 101
0110
- 101
010
- 00
100

Ans = 1010.1100 1100

(d) 11011.1 by 101 = 101.000 1100 1100 Ans

101) 1.11011 (101.000 1101
- 101
011
- 00
111
- 101
010
- 00
100
- 000
1000
- 101
011
- 101
010
- 101
00 10

HW

2.9

Divide the following binary numbers by the computer method

(a) 1011 by 10

```
10 ) 1011 ( 101.1
   - 10
     01
   -  0
     011
   -  10
     010
   -  10
     00
```

Ans ÷ 101.1

(b) 110010 by 101

```
101 ) 110010 ( 1010
    - 101
      0010
    -   0
      101
    - 101
      000
    -   0
      000
    -   0
      0
```

Ans ÷ 1010

(6) 1100100 by 1101

1101) 1100100 (101.00111	111.1011000
- 1101	1101) 1100100
1000	- 1101
- 0000	11000
10000	- 1101
- 1101	10110
00110	- 1101
- 000	10010
1100	- 1101
- 000	001010
11000	- 0000
- 1101	010100
10110	- 1101
- 1101	001110
10010	- 1101
- 1101	00010
010100	00
- 1101	100
00111	- 000
	1000

Answer: 111.1011000

(d) 100010 by 1010

```
1010) 100010 (11.0110011100
      1010
      -----
       01110
     - 1010
      -----
        1000
      - 0000
      -----
        10000
      - 1010
      -----
         0110
       - 1010
       -----
          0100
        - 0000
        -----
          1000
```

Ans ÷ 11.0110011100

(e) 11001 by 11

```
11) 11001 (1000.0101
    - 11
    ----
      00
    - 00
    ----
       1
     - 0
     ---
       10
     - 00
     ----
       100
     - 11
     ----
       010
```

Ans ÷ 1000.0101

Subtraction Using 1's & 2's Complement

Using 1's Complement

rule:- सबसे पहले 0 लगाकर ऊपर और नीचे के digit बराबर बना दो।

1) पहले जिसमें minus लगा हो उसका 1's complement निकालना है।

2) ∵ A+(-B) के हिसाब से फिर ऊपर वाले के साथ 1's complement को जोड़ना है।

और जोड़ने पर यदि last में carry आए तो उसे LSB में जोड़ देना है।
जिसका मतलब carry है तो वह धनात्मक है।

(3) यदि carry नहीं है तो उसका मतलब वह Negative है और Negative sign लगाकर Ans का फिर से 1's complement लगायेंगे निकालेंगे।

Question 10001.01 - 01111.10

```
  10001.0110
- 01111.10
```

Since Negative sign is asign with the number 1111.10

Hence taking 1's complement of 01111.10

1's complement of 01111.10 = 10000.01

10001.01 + (-1111.10)

10001.01
+ 10000.01
100002.10

Since carry have generated from MSB therefore result in

Add the final carry in LSB

00001.10
+ 1
00001.11 Ans

Question: 1111.10 − 10001.01

01111.10
− 10001.01

Solve: 1111.10 + (−10001.01)

Given

A = 1111.10 B = (−10001.01)

Here number A is positive & B is Negative

so we have to take the 1's complement of B

1) Convert number B to 1's complement

(10001.01)$_2$ 1's complement = 01110.10

2) Add 1's complement of A to number B:

```
          0111/.10              01111.10
1's compl 10001.10            + 01110.10
         100001.00        [1] 011110.00  Result Ans
                                |
         100001.00          final carry
```

As the final carry is 0, the answer is negative and its 1's complement form

So convert the Ans into its true form as follow

11110.00 का 1's complement

= - 00001.11 Ans

Substraction of Using 1's complement and the ans is - 00001.11.

Question :- Substract using 1's Complement

(1) $(1010)_2 - (1011)_2$

step 1) obtain 1's complement of 1011

$1011 \xrightarrow{\text{1's compl}} 0100$

Page: 76
Date: / /

Add $(1010)_2$ and 1's complement of $(1011)_2$

1st number : 1010

1's complement of 2nd number : 0100

1110

Final carry 0

As carry is not generated from MSB therefore Answer is negative and in 1's complement form

3) Convert the answer into its true form

1110 $\xrightarrow{\text{Invert}}$ 0001

Thus the answer is $(0001)_2$ Ans.

Question

let A = 1010 B = 1101

obtain 1's complement of $(0101)_2$:

0101 $\xrightarrow{\text{1's com}}$ 1010

Add A to 1's complement of B

1010
+ 1010

Final carry 10100

True form = 0100
+1
0101 Ans.

Question: 1001 - 1101

$$\begin{array}{r} 1001 \\ -\ 1101 \\ \hline \end{array}$$

Let A = 1001 ; B = 1101

1) Obtain 1's complement of (1101)

1's compli
1101 ⟶ 0010

2) Add A to 1's complement of B

$$\begin{array}{r} 1001 \\ +\ 0010 \\ \hline 1011 \end{array}$$

Final carry = 0

As final carry is not generated the answer is negative and it true form is

Thus $(1001)_2 - (1101)_2 =$

1011 of 1's complement $= (0100)_2$

Thus Answer is $(-0100)_2$

Subtraction Using 2's complement

2's complement :

1) Subtraction के लिए पहले ऊपर और नीचे दोनों bits equal करने है।

2) जिसमें Negative sign लगा हो उसे 2's complement लेना है।

3) पहले ऊपर वाले नंबर में 2's complement को जोड़ देना है।

4) यदि final carry आए MSB में तो उस carry को ignor करना है और वह positive number देगा।

5) यदि final carry 0 हो/आए ही ना तो Answer का पुनः 2's complement निकालना है।

Question 1010 - 0101 using 2's complement

let A = 1010
- 0101

let A = 1010
B = 0101

obtain 2's complement of (0101)

0101 —(1's complement)→ 1010 —(Add +1)→ 1011

Add A to 2's complement of B.

Ignore the carry

1010
+ 1011
Final 10101

As Carry is generated the answer is positive and it's true form is (0101)$_2$ Ans

Question:2 1001 - 1101 sub. using 2's complement

let A = 1001 , B = 1101

→ obtain 2's complement of (1101)

1's C Add +1
1101 → 0010 → 0011

→ Add A to 2's complement of B

```
  1001
+ 0011
------
  1100
```

final carry = 0
carry is not generated
the Ans is negative and its result
1100 2's complement of Ans

1's C Add 1
1100 → 0011 → (0100), Ans

Q.:3) Subtract using 2's complement

1010 - 1011

obtain 2's complement of 1011

1011 —1's C→ 0100 —+1→ 0101

Add 1010 to 2's complement

```
  1010
+ 0101
------
  1111
```

Carry not generated
the ans is negative
Ans of 2's complement

(0001)2 Ans

Substraction Using 1's & 2's complement

Q. 10110 - 1011

Using 1's Complement

Obtain 1's complement of (01011)

1011 $\xrightarrow{1's\ C}$ 10100

Add 10110 and 2's complement of 1011

10110
+ 10100
---
1 10010

Carry is generated it is positive true form is 01011

Using 2's Complement

10110 - 01011 (01011 - 11010)

Obtain 2's complement of (01011)

01011 $\xrightarrow{1's\ C}$ 10100 $\xrightarrow{+1}$ 10101

Add 10110 and 2's complement

10110
+ 10101

101011

final Carry is generated, True form is Carry is ignore

Ans. → 01011

Subtraction Using 1's & 2's Complement

Q. 01011 − 10110

Using 1's complement

obtain 1's complement of 10110

10110 → 01001

Add 10110 and 01001

```
  01011
+ 01001   Carry is not generated
-----     the answer is negative
  10100
```

and it's true form of 1's complement of Ans

10100 —1's→ − 01011 Ans

Using 2's complement (01011 − 10110)

obtain 2's complement of 10110

10110 —2's→ 01001 —+1→ 01010

Add 01011 & 2's complement of

```
  01011   carry not generated
+ 01010   the answer is negative
-----
  10101
```

and it's true form of Ans of 2's Com

10101 —1's→ 01010 —2's→ − 01011 Ans

Subtraction using 1's & 2's Complement

⊗ 1100.10 − 111.01

Using 1's Complement

obtain 1's complement of (0111.01)

$0111.01 \xrightarrow{1's} 1000.10$

Add 1100.10 and 1's Complement

1100.10 + 1000.10 = 10101.00 Carry generated, Ans is positive

Ans is 0101.01 + 1 = 0101.10 Ans

Using 2's Complement (1100.10 − 0111.01)

obtain 2's complement 0111.01

$0111.01 \xrightarrow{1's\ C} 1000.10 \xrightarrow{\text{Add } 1} 1000.11$

Add 1100.10 and 2's complement

1100.10 + 1000.11 = 10101.01

Carry is generated the Ans is positive

True form of Ans is (0101.01)

(Carry को ignore करेंगे 2's Com में)

Substraction using 1's & 2's complement

Q:- 111.01 − 1100.10

Using 1's complement

Obtain 1's complement of 1100.10

1100.10 → 0011.01

Add 111.01 and 0011.01

```
  0111.01
+ 0011.01
  1010.10
```

Carry is not generated so the answer is negative and it's true from

1010.10 of is complement = -0101.01 Ans

Using 2's complement

(111.01 − 1100.10)

Obtain 2's complement

1100.10 —is→ 0011.01 —2's→ 0011.10
(+1)

Add 111.01 and 2's complement

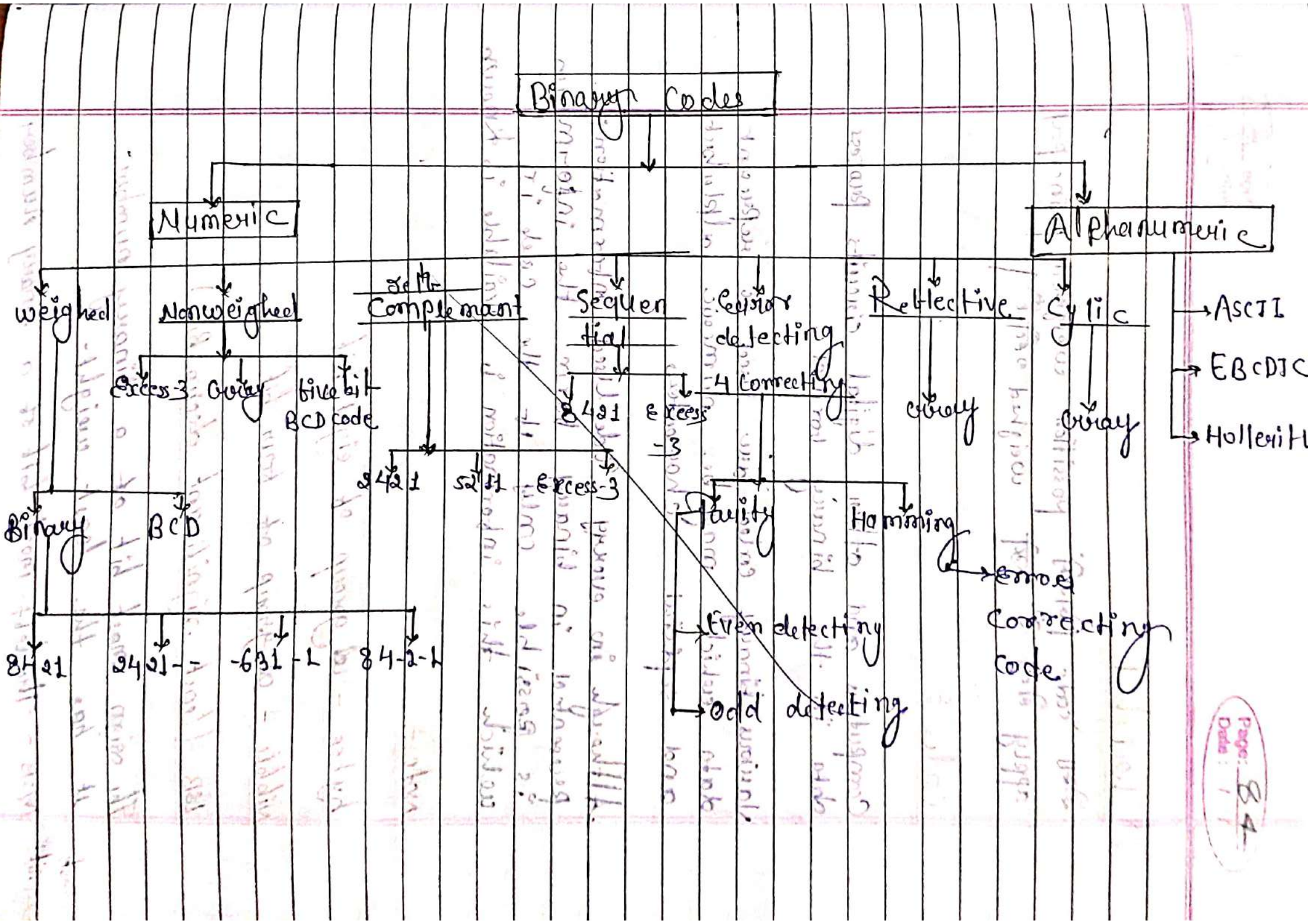
```
  0111.01
+ 0011.10
  1011.11
```

Carry is not generated the answer is Negative

The true from of

1011.11 —2's com→ 0100.10 —+1→

—+1→ 0101.01 Ans


Binary Codes
Numeric
Alphanumeric
Weighted
Nonweighted
9's Complement
Sequential
Error detecting & correcting
Reflective
Cyclic
ASCII
EBCDIC
Hollerith
Excess-3
Gray
five bit BCD code
8421
Excess-3
2421
52 11
Excess-3
Gray
Gray
Binary
BCD
Parity
Hamming
8421
2421
-6311
8-4-2-1
Even detecting
Odd detecting
error correcting code

(I) Weighted :-

ऐसा code जिसमें position weighed principal apply होता है उसे weighed कहते है।

Codes :-

Computers and other digital circuits process data in the binary format.

Various binary codes are used to represent data which may be numeric, alphabets and special characters.

Although in every code used information is presented in binary form. The information is possible only if the code in which this information is available is known.

Note :-

Bytes - A group of eight bit

Nibble - A group of four bit

LSB - (least significant decim bit)

The sign most bit of a binary number it has the least weight.

MSB = The left most bit of a binary number

most significant

(1) ~~to~~ Numeric code -

1) Weighted 2) Non-weighted 3) self complementing
4) Sequential 5) Error detecting 6) Reflective 7) cylic

(1) Weighted code :

weighted code जैसे binary code को कहा जाता है जो position weighing principal के rule को follow करता है। that means number के हर एक position का weight fix / specific होते है।

Example :- 8421, 2421, 84-2-1

★ weighted code में 4 bit के group के number से represent किया जाता है।

(2) Non - weighted code :

Non-weighted code जैसे binary code को कहा जाता है जो position weighing principal के rule को follow नही करता है that means number के हर एक position का weight fix नही होता है।

Example → Xs-3, gray code, 5-bit, BCD code

यह दो प्रकार के होते हैं -

(1) positive weighted code

(2) Non - weighted code

(1) positive weighted code

example $(1011)_2 = (\ )_{10}$

$= 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= (11)_{10}$ Ans.

(2) Non - weighted code :

$(1011)_2$ (8 4 -2 -1)

$= 1\times8 + 0\times4 + 1\times(-2) + 1(-1)$

$= 8 + 8 - 2 - 1 = 5$

(3) Sequential Codes

Sequential code ऐसे binary code होते हैं जिसमें हर अगला succeeding code अपने privious preciding code से एक number अधिक होता है।

Sequential code is one in which each succeeding code is one binary number greater than its preceding code 0 = 0000 1 = 0001

ex → excess-3 code, 8421

Self Complementing Code

A code is said to be self complementing if the code of 9's complement of the given number is obtained by interchanging all zero's & one's

ex → excess-3 code (2421 weight का addition 9 होता है)

Advantage :

★ there logical complement as same as the arithmetic complement

★ for to be self complementy the sum of all its bit must be 9.

(2 4 2 1) (5 2 1 1)

Hence = 2+4+2+1 = 9

9's complement = 1's complement

In exess-3 code Decimal no Exess-3

9's complement [0 / 9

0011 ↔ 1100 changing

Page: 20
Date: 1/1

Reflective code :

A reflective code is a binary code in which n least significant bits for code words 2^n through 2^{n+1} are mirror image of 0 through $2^n - 1$

ex :- Gray code

	same	
0 - 0000	8 = 1000	least significant means last के दोनों bit same
1 - 0001	9 = 1001	
2 = 0010	10 = 1010	
3 = 0011	11 = 1011	
4 = 0100	12 = 1100	
5 = 0101	13 = 1101	
6 = 0110	14 = 1110	
7 = 0111	15 = 1111	

Cyclic :

Cyclic code are those code in which successive code differs from the preceding code by only one weight position It is also called unit distance code. They can minimize transition errors and blurring.

Unit distance या transition distance क्या होता है

ex → Gray (इसका use analog device को control करने के लिए किया जाता है

BCD → Binary coded Decimal

In this code each decimal digit is coded exc. in generally four bit binary code. It exectly only for 10 decimal symbol which are 0 to 9

8421 is are Natural BCD code

It is a weight and sequentiable code Used for mathematical operation.

Advantage :

It simplify the conversion of decimal to BCD and vice versa

Decimal	BCD	
0	0000	valid
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	

prea

a) There are 6 Invalid code present in BCD

1010, 1011, 1100, 1101, 1110, 1111 → Invalid

Weighted Codes :

जब किसी कोड में किसी संख्या के प्रत्येक बॉक्स का एक स्थानीय मान होता है अर्थात प्रत्येक अंक का एक weight होता है और उस संख्या का Decimal मान उन सभी स्थानीय मानों के योग के बराबर होता है। तब उस कोड को weighted code कहते हैं।

Ex → BCD या 8421 कोड।

इस कोड में संख्या 7 को 0111 से प्रदर्शित किया जाता है।

संख्या के सभी अंकों का स्थानीय मान रखने पर

$0111 = 0 + 2^2 + 2^1 + 2^0$

$= 0 + 4 + 2 + 1 = 7$

Ex-3 कोड व ग्रे कोड वेटेड कोड नहीं है क्योंकि उन कोड में अंकों के कोई स्थानीय मान नहीं होते।

BCD code :

जब एक Decimal number के सभी अंकों को उनके तुल्य बाइनरी से प्रतिस्थापित किया जाता है तब BCD कोड प्राप्त होते हैं।

→ यह एक Natural Binary code है।

जिसमें 0 से 9 तक सभी डेसीमल अंको को उनके बाइनरी तुल्यांक द्वारा प्रदर्शित किया जाता है।

→ BCD संख्या में 4 bit (1 nibble) होते है।

→ इसे 8421 भी कहते है।

इसका उपयोग calculator तथा Digital voltmeter में Decimal no को प्रदर्शित करने के लिये किया जाता है।

(2) Excess-3 Code :-

It is also called non weighted BCD code. This code derives from adding 0011 (3) to the given BCD code.

BCD code + 0011 = Excess-3

यह एक महत्वपूर्ण 4-bit कोड है प्रत्येक डेसीमल अंक में पहले 3 जोड़कर तथा उस योग को 4 बिट बाइनरी यानी BCD में परिवर्तित कर एक्सेस-3 कोड प्राप्त किया जाता है।

∵ इस कोड में किसी भी बिट को एक निश्चित स्थानीय मान या भार (weight) नहीं होता है इसलिए ये Non-weighted है।

	D	Excess
Ex →	3 + 3 = 6	0110
	5 + 3 = 8	1000

Alphanumeric code :-

सभी डिजिटल computers में decimal संख्याओं के अलावा alphabet के capital letters व lowercase letters, अनेक गणितीय संकेत जैसे (+, -, *, /, >, <) तथा कुछ विशेष चिह्न (special symbol, characters ex- @, #, ?, *) आदि का भी उपयोग किया जाता है। इन सभी को बाइनरी में coding करने के लिए बाइनरी अंकों की एक विशेष शृंखला की आवश्यकता होती है। इन शृंखला को Alphanumeric कोड कहते हैं।

Alphanumeric के तीन प्रमुख प्रकार होते हैं

(1) ASCII code
(2) EBCDIC code
(3) Hollerith code

(1) ASCII code :-

ASCII full form American Standard code for information Interchange

इस कोड की सहायता से निर्माता अपने computers के keyboard, printers, तथा video display युक्तियों को सामान्य प्रयोग के लिए प्रमाणीकृत करते हैं।

ASCII कोड एक 7 bit कोड होता है जो 0 और 1 की शृंखलाओं से बना होता है। ASCII कोड को निम्न format द्वारा प्रदर्शित किया जा सकता है -

$b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$ A = 1000001

b का मान 0 या 1 हो सकता है।

EBCDIC :- (Extended Binary Coded Decimal Interchange code)

ASCII की तरह यह भी सभी Alphabets, Numbers व special characters के मानक कोडिंग हेतु प्रयुक्त किया जाता है।

अंतर सिर्फ यह है कि यह 8 बिट कोड प्रणाली है।

Holerith code :-

जब सूचनाओं के आदान-प्रदान के लिये Punch Card का उपयोग होता था तब इस कोड का व्यापक रूप से प्रयोग किया जाता था। कार्ड के मानक माप (standard size) में 80 Column और 12 Row होते थे।
होलेरिथ नामक व्यक्ति ने 0 व 1 के 12 bits की शृंखला द्वारा किसी भी Alpha-Numeric Code को पहचानने के लिये अंतर्राष्ट्रीय स्तर पर इस कोड का निर्माण किया था।

Convert Decimal to BCD code
10, 19, 105, 438 तथा 989

(i) $(10)_{10} \rightarrow (00010000)_{BCD}$

$(19)_D \rightarrow (00011001)_{BCD}$

$(438)_D \rightarrow (010000111000)_{BCD}$

$(989)_D \rightarrow (100110001001)_{BCD}$

Convert Decimal to Ex-3

(i) 10

$$\begin{array}{r} 10 \\ +33 \\ \hline 43 \end{array}$$

↓ ↓

0100 0011

अतः Decimal 10 का ex-3 कोड (0100 0011) होगा।

(ii) Decimal number 28

2 + 3 = 5 $\xrightarrow{3}$ 0101

8 + 3 = 11 $\longrightarrow$ 1011

= (0101 1011)$_{Ex-3}$

अतः डेसीमल 28 का ex-3 कोड (0101 1011) होगा।

(iii) 105

Decimal no. 1 0 5

Add +3 $\begin{array}{r} +3 \quad 3 \quad 3 \\ \hline 4 \quad 3 \quad 8 \end{array}$

↓ ↓ ↓ Binary

0100 0110 1000

(0100 0110 1000)$_{Ex-3}$

अतः Decimal 105 का ex-3 0100 0110 1000 होगा।

I

Page: 97
Date 29/10/23

Binary to Gray : Code

Ex-OR operation ⊕

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

$0 \oplus 1 = 1$

Ex-OR → Same bit का output 0 e.
और अलग-अलग bit का 1 होगा।

$B_3 B_2 B_1 B_0 = G_{13} G_{12} G_{11} G_{10}$

Formula :

$G_{13} = B_3$

$G_{12} = B_3 \oplus B_2$

$G_{11} = B_2 \oplus B_1$

$G_{10} = B_1 \oplus B_0$

Eg → 1011 Convert this Binary no into Gray code

Given :

binary no. = 1011

Since

~~G13~~ Binary code is $B_3 B_2 B_1 B_0$

$B_3 = 1$, $B_2 = 0$, $B_1 = 1$, $B_0 = 1$

$\because G_{13} = B_3 = \boxed{G_{13} = 1}$

$G_{12} = B_3 \oplus B_2 = 1 \oplus 0 = \boxed{G_{12} = 1}$

$G_{11} = B_2 \oplus B_1 = 0 \oplus 1 = \boxed{G_{11} = 1}$

$G_{10} = B_1 \oplus B_0 = 1 \oplus 1 = \boxed{G_{10} = 0}$

Ans : Gray code : $G_{13} G_{12} G_{11} G_{10} = (1110)$

Q. 1001 convert into Gray code

$(1001)_2 = (1101)$ Gray code

$G_{13} = B_3 = 1$

$B_2 = B_3 \oplus B_2 = 1 \oplus 0 = 1$

$B_1 = B_2 \oplus B_1 = 0 \oplus 0 = 0$

$B_0 = B_1 \oplus B_0 = 0 \oplus 1 = 1$

Q. $(1101)_B = (1011)$ Gray

Gray to binary

Gray code = $G_3 G_2 G_1 G_0$

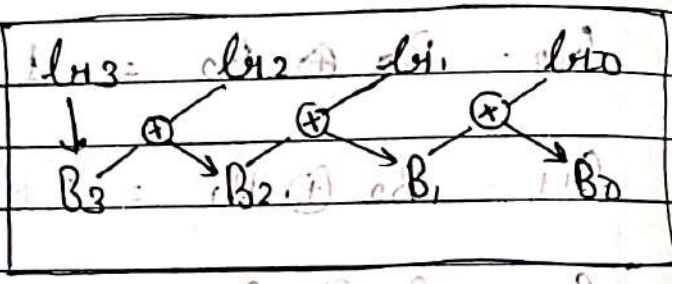
$G_3 G_2 G_1 G_0 = B_3 B_2 B_1 B_0$

$B_3 = G_3$

$B_2 = B_3 \oplus G_2$

$B_1 = B_2 \oplus G_1$

$B_0 = B_1 \oplus G_0$



Q Convert the Gray code 1011 into Binary

Given →

Gray code = (1011) Gray

$G_3 G_2 G_1 G_0$ ⇒ $G_3 = 1$, $G_2 = 0$, $G_1 = 1$, $G_0 = 1$

Since

$B_3 = G_3 = 1$

$B_2 = B_3 \oplus G_2 = 1 \oplus 0 = 1$

$B_1 = B_2 \oplus G_1 = 1 \oplus 1 = 0$

$B_0 = B_1 \oplus G_0 = 0 \oplus 1 = 1$

RC $(B_3 B_2 B_1 B_0)_{30} = (1101)_2$

Convert gray code (100111) into binary number?

Given :-

gray code = (100111) gray

Gray code = $G_5 G_4 G_3 G_2 G_1 G_0$

Comparision

$G_5 = 1$, $G_4 = 0$, $G_3 = 0$, $G_2 = 1$, $G_1 = 1$, $G_0 = 1$

Since

$B_5 = G_5 = 1$ $\boxed{B_5 = 1}$

$B_4 = B_5 \oplus G_4 = 1 \oplus 0 = \boxed{1 = B_4}$

$B_3 = B_4 \oplus G_3 = 1 \oplus 0 = \boxed{1 = B_3}$

$B_2 = B_2 \oplus G_2 = 1 \oplus 1 = \boxed{0 = B_2}$

$B_1 = B_2 \oplus G_1 = 0 \oplus 1 = \boxed{1 = B_1}$

$B_0 = B_1 \oplus G_0 = 1 \oplus 1 = \boxed{0 = B_0}$

Binary code is $(B_5 B_4 B_3 B_2 B_1 B_0)$

put value $(111010)_2$

Q Convert $(1011)_2$ into Gray

1 0 1 1

1 1 0 1

(1101) Ans

Error Correcting Code

When binary data is transmitted and processed than noise can alter or distract (ध्यान भंग) is content

than 1's may get change to 0 than error detecting code are used.

Parity :

इसकी मदद से error detect किया जा सकता है।

The simplestic technique for detective error is adding and extra bit known as parity bit.

There are two types of parity

(1) even parity (2) odd parity

even parity : total number of 1's in transmitted sequence should be even number.

Odd parity : total number of 1's is transmitted sequence should be odd number of 1

Decimal	BCD	~~odd~~ Parity	~~even~~ Parity
0	0000	1	0
1	0001	0	1
2	0010	0	1
3	0110	1	0
4	0100	0	1
5	0101	1	0
6	0110	1	0
7	0111	0	1
8	1000	0	1
9	1001	1	0

Hamming code :- (7's bit hamming code)

To transmit four bit data three parity bit's located at position 2^0, 2^1 & 2^2 from left are added to make 7's bit code which is transmit.

यह एक error detection code है साथ ही error correction code भी है जो data संग्रहित और प्राप्ति होने पर error का पता लगा सकता है और उन्हे ठीक भी कर सकता है।

उसमें data के साथ अतिरिक्त Parity bits जोड़ने की आवश्यकता होती है जिसका नाम अविष्कार Richard W. hamming ने की।

$P_1 P_2 D_3 P_4 D_5 D_6 D_7$

The code word format is given below

$[P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7]$

जहाँ

D = Data weight
P = parity weight

P_1 is set to be 0 or 1 to establish even parity over weight 1,3,5,7 ($P_1 D_3 D_5 D_7$)

$$P_1 = P_1 D_3 D_5 D_7$$

P_2 is set to be 0 or 1 to establish even parity over weight 2,3,6,7 ($P_2 P_3 D_6 D_7$)

$$P_2 = P_2 D_3 D_6 D_7$$

P_4 is set to be 0 or 1 to establish even parity over weight 4,5,6,7 ($P_4 D_5 D_6 D_7$)

$$P_4 = P_4 D_5 D_6 D_7$$

In code data weight को into 7 weight even parity hamming code.

अगर Hamming code बनाते है तो :-

given →
Data = 1101
$P_1 P_2 D_3 P_4 D_5 D_6 D_7$

$D_3 = 1$, $D_5 = 1$ $D_6 = 0$, $D_7 = 1$ मान रखने पर

$P_1 P_2 1 P_4 1 0 1$

$P_1 = P_1 D_3 D_5 D_7 = (P_1 111) = P_1 = 1$

$P_2 = P_2 D_3 D_6 D_7 = (P_2 101) = P_2 = 0$

$P_4 = P_4 D_5 D_6 D_7 = (P_4 101) = P_4 = 0$ लिखेंगे
∵ हमें even parity code बनाना है तो यहाँ दो बार 1 दिया है अतः $P_4 = 0$ लिखो

Finally hamming code is
$P_1 P_2 P_4 101 = (1010101)$

Q. Convert 0001 into even hamming
$P_1 P_2 D_3 P_4 D_5 D_6 D_7 = P_1 P_2 0 P_4 001$

$P_1 = P_1 D_3 D_5 D_7 = (P_1 0 0 1) = P_1 = 1$
$P_2 = P_2 D_3 D_6 D_7 = (P_2 001) = P_2 = 1$
$P_4 = (P_4 D_5 D_6 D_7) = (P_4 001) = P_4 = 1$

final hamming code is $P_1 P_2 0 P_4 001$
$= (1101001)$ Ans.

Convert 0010 into even hamming

$(0010)_{BCD}$

$P_1P_2 0 P_4 010$

$P_1 = P_1D_3D_5D_7 = P_1000 = P_1 = 0$
$P_2 = P_2D_3D_6D_7 = P_2010 = P_2 = 1$
$P_4 = P_4D_5D_6D_7 = P_4010 = P_4 = 1$

Final $P_1P_2 0 P_4 010 \rightarrow 0101010$ Ans

Convert 0011 into even hamming

$P_1P_2 0 P_4 011$

$P_1 = P_1D_3D_5D_7 = P_1001 = P_1 = 1$
$P_2 = P_2D_3D_6D_7 = P_2011 = P_2 = 0$
$P_4 = P_4D_5D_6D_7 = P_4011 = P_4 = 0$

Final $P_1P_2 0 P_4 011 = 1000011$ Ans

Convert 0100 into even hamming

$P_1P_2 0 P_4 100$

$P_1 = P_1D_3D_5D_7 = (P_1010) = P_1 = 1$
$P_2 = P_2D_3D_6D_7 = (P_2000) = P_2 = 0$
$P_4 = P_4D_5D_6D_7 = (P_4100) = P_4 = 0$

Error Correction

$C_3C_2C_1$

$C_1 = P_1 \oplus D_3 \oplus D_5 \oplus D_7$

$C_2 = P_2 \oplus D_3 \oplus D_6 \oplus D_7$

$C_3 = P_4 \oplus D_5 \oplus D_6 \oplus D_7$

Let supose we recive 7' weight hamming code as 1111001

1 1 1 1 0 0 1
P_1 P_2 D_3 P_4 D_5 D_6 D_7

$C_1 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$
$C_2 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$
$C_3 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$

$C_3C_2C_1 = (011) = (3)_{10}$ (3rd position पर error है)

Hamming code = 11□1001
Mind में error है तो 1 = 0 लिखना है
Corrected Humming code = 1101001

Corrected Hamming code is 1101001

Unit : 2

Logic gates and Boolean algebra

Logic gate :

ऐसे circuit जिसके input एवं output के मध्य logic Relation होते है तथा logical rule जो संचालित होते है logic gate कहलाते है।

It produce output level with respect to some combination of input level

* digital circuit, gate की मदद से बनता है

logic gate वह electronic circuit होता है जो एक logic निर्णय लेने के लिए प्रयुक्त किया जाता है जिसमें एक से अधिक input के संगत केवल एक ही output प्राप्त होता है।

जो input के combination के basses पर output निकालता है।

logic gate are divided into three Catogoiry

Basic

(1) logic gate → AND gate, OR Gate & NOT gate

(2) Universal gate → NAND Gate & NOR Gate

(3) Special perpose gate : Ex-OR / X-OR Gate & X-NOR / EX NOR

Boolean Algebra

वह Algebra जो बाइनरी चर राशियों और Logic क्रियाओं (AND, OR & NOT) से संबंधित होता है।

Binary Variable का Algebra को boolean algebra कहते हैं।

(1) Basic logic Gates

जिसमें basic gates होते हैं इनमें एक से अधिक input के संगत केवल एक ही output प्राप्त होता है जो दिये गये input से उत्पन्न परिणाम को प्रकट करता है।

logic gate प्रायः digital पद्धति के आधार पर होते हैं basic logic gate निम्न प्रकार के होते हैं

(1) AND Gate
(2) OR Gate
(3) NOT Gate

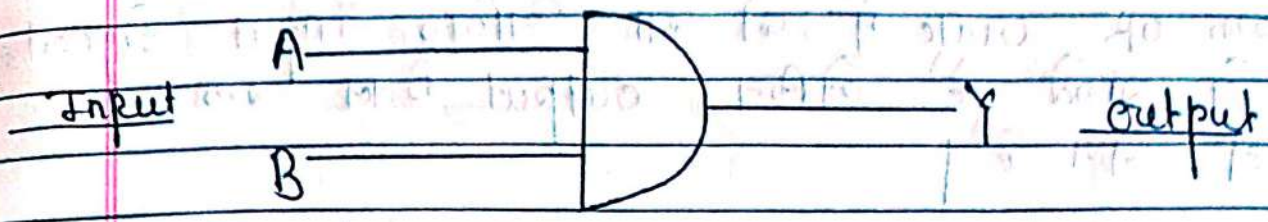
(I) AND Gate

यह एक logical device है। AND Gate में दो या अधिक इनपुट होते हैं परंतु output केवल एक होता है। Gate का output केवल तब high होता है जब उसके सभी input high होते हैं।

And Gate → minimum value theory है क्योंकि इसका output अधिकांश minimum output होता है।

Two Input And Gate

Symbol :



logical symbol of AND Gate

Truth Table

Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Logical expression ⇒ $Y = A \cdot B$ या AB

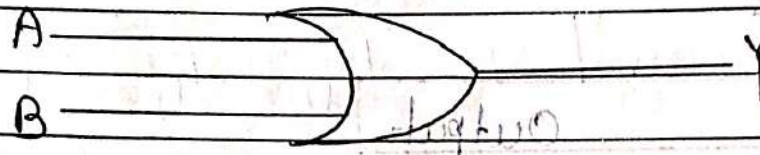
परिभाषा :

AND Gate में output logic level पर 1 तभी होगा जब उसके सारे input logic level पर 1 होंगे otherwise एक भी input logical level 0 का होता है तो output logic level 0 प्राप्त होगा।

(2) OR Gate :

एक OR Gate में दो या अधिक input signal हो सकते है लेकिन output केवल एक ही होता है।

Symbol : Two Input OR Gate



logic symbol of OR Gate

Truth Table

Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

logical expression = $Y = A + B$

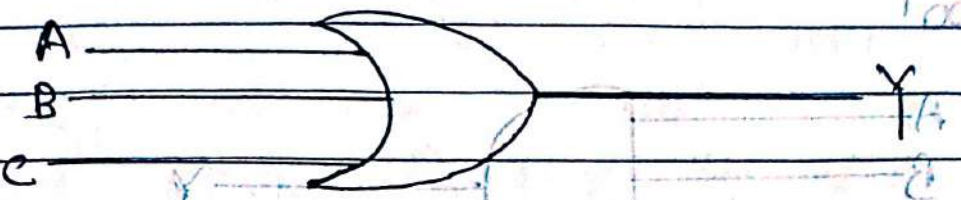
+ → symbol of OR operation

परिभाषा : OR Gate का output logic level पर 0 तभी प्राप्त होगा जब उसके सारे input 0 होंगे और अगर यदि एक भी input 1 होगा तो output 1 प्राप्त होगा।

Page:
Date: / /

Three Input OR Gate

Symbol :-



logical symbol of OR Gate

Truth table

Input			Output
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Logical expression

$$Y = A + B + C$$

Page:
Date: / /

Three Input AND Gate

symbol

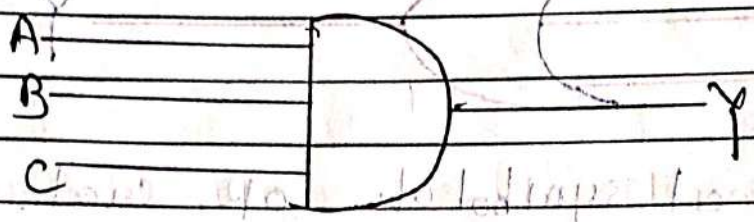


Fig - logic gate of three input AND gate

Truth table:

Input			Output
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Logical expression

$$Y = A \cdot B \cdot C$$

Page:
Date: / /

(3) NOT Gate :-

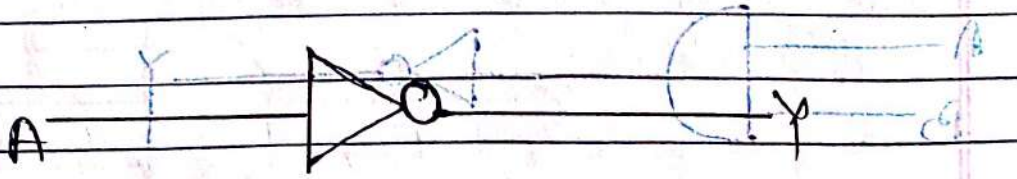
यह एक Inverter को प्रदर्शित करता है। यह transistor का एक सामान्य कॉमन एमीटर एम्प्लीफायर परिपथ है जिसका output Input से 180° पर प्रतिस्थापित होता है।

(1) इसे inverter भी कहते हैं।

(2) इसमें सिर्फ एक output और एक input होता है।

(3) NOT GATE का output, Input का complimented (opposite) form होता है। that means जब Input logic level 0 दिया जाता है तो output logic level 1 होता है। यह output में complimented input logic देता है अर्थात जब input logic level 1 होता है तो output logic level 0 होता है।

Symbol



Truth table

I/p A	Y o/p
0	1
1	0

logical expression

$Y = \bar{A} / A'$

मैं gate को भी logic gate को implement

(2) Universal Gates

इसमें सारे logical system बनाए जा सकते है।

यह दो प्रकार के होते है

(1) NAND GATE (2) NOR GATE

(1) NAND GATE → digital logic gate है।

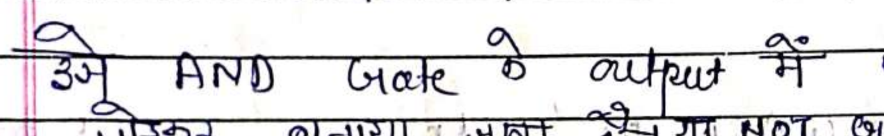
NAND GATE that mean NOT AND Gate है।

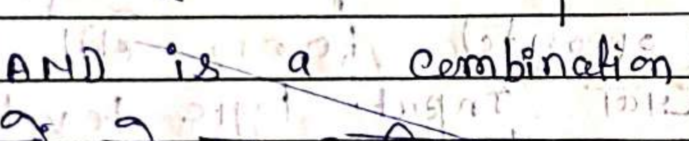
इसे AND Gate के output में NOT Gate को जोड़कर बनाया जाता है। या NOT Gate में बदला जाता है।

NAND is a combination of AND and NOT

इसमें दो या अधिक Input और एक o/p होते है। इसका behaviour AND gate के opposite होता है।

logical diagram Two input NAND Gate





logic symbol of NAND Gate

Truth table

Input		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

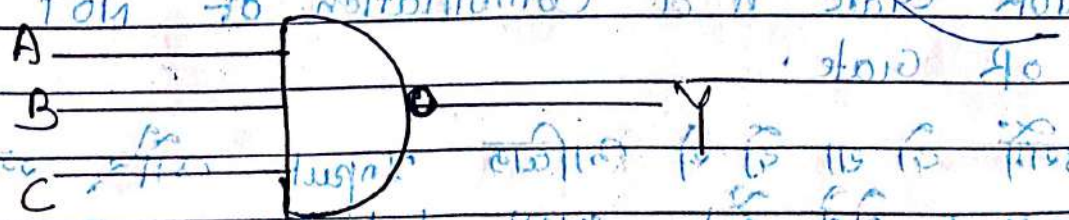
Logical expression :- $Y = \overline{A \cdot B}$

परिभाषा :-

NAND Gate का output logic level पर 0 तभी होगा जब उसके सारे input 1 होंगे otherwise output logic level पर 1 होगा यदि एक भी input 0 है तो output logic level पर 1 होगा।

Three Input Out NAND Gate

symbol



logic symbol of three input NAND Gate

Page:
Date: / /

Truth table

Input			Output
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Logical expression

$$Y = \overline{A \cdot B \cdot C}$$

NOR Gate :-

यह digital logic gate है इसमें OR Gate के output को NOT Gate में convert करते है।

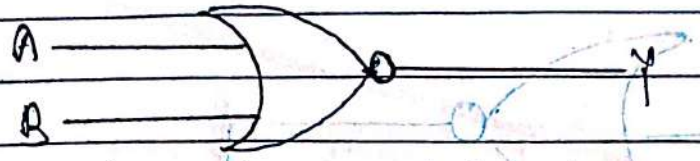
NOR Gate is a Combination of NOT and OR Gate.

इसमें दो या दो से अधिक Input और एक output होते है। इसका behaviour OR Gate का opposite होता है।

OR Gate output will be NOTed.

Page:
Date: / /

Symbol :



Logical symbol of NOR Gate

Truth table :

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Logical expression : $Y = \overline{A+B}$

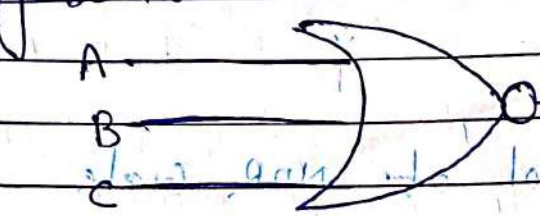
परिभाषा :

NOR Gate का output logic level पर 1 तभी होगा जब उसके सारे input logic level पर 0 होंगे otherwise यदि एक भी input logic 1 है तो output logic level पर 1 होगा।

Page:
Date: / /

Three Input NOR Gate

Symbol :-



Truth table :-

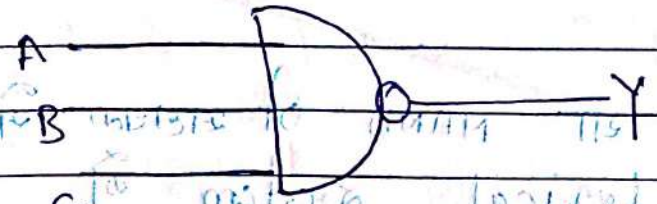
input			output
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Logical expression :- $Y = \overline{A+B+C}$

Page:
Date: / /

Three Input NAND Gate

Symbol



Logic Symbol of NAND GATE

Truth table

Input			O/P	
A	B	C	Y'	Y
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Logical expression $Y = \overline{A \cdot B \cdot C}$

Page:
Date: / /

Basic → इनसे सभी Combination बनते हैं।

Universal Gate

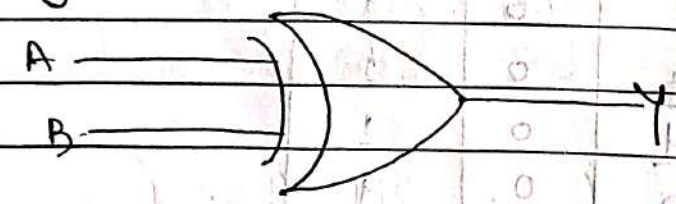
क्योंकि NOR Gate या NAND की सहायता से हम किसी भी Logical System को Implement कर सकता है।

Ex - OR →

Special purpose Gate

Exclusive - OR Gate :

Symbol :



logical symbol

Logical expression : $Y = A \cdot \oplus B$

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

Page:
Date: / /

Truth table

Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

input 2 है तो combination 4 होगा।

ex-OR gate is a two input one output logic circuit / device.

ex-OR gate का output logic level पर 1 तभी होगा जब दोनो में से केवल एक input logic level पर 1 होगा otherwise output 0 होगा

$A \cdot \bar{A} + B \cdot \bar{A} = B \oplus A = Y$

Ex-OR gate को Inequality / Anti-coincidence detector भी कहते है।

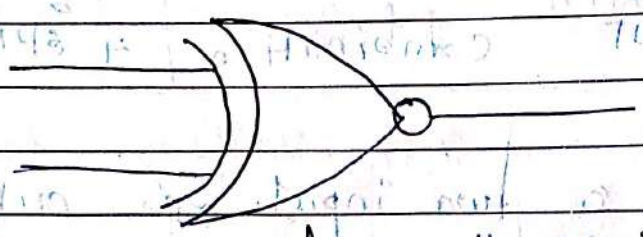
Page:
Date: / /

Ex - NOR Gate

Ex - NOR Gate is a combination of ex-OR Gate and NOT Gate.

Two input one output logic device

Symbol



logical symbol

logical expression

$$Y = A \odot B = \bar{A} \cdot \bar{B} + A \cdot B$$

$$Y = \overline{A \oplus B}$$

Truth table

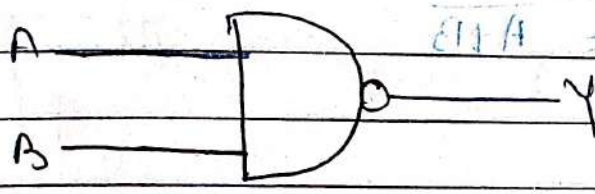
Input		output
A	B	Y
0	0	1
1	0	0
0	1	0
1	1	1

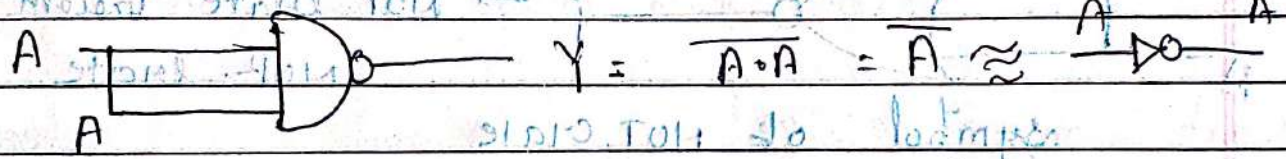
Page:
Date: / /

Ex-NOR Gate का output logic level पर 1 तभी होगा जब दोनों input 0 या 1 हो या फिर logic level 1 पर हो या 1 पर हो।

NAND Gate as a Universal gate

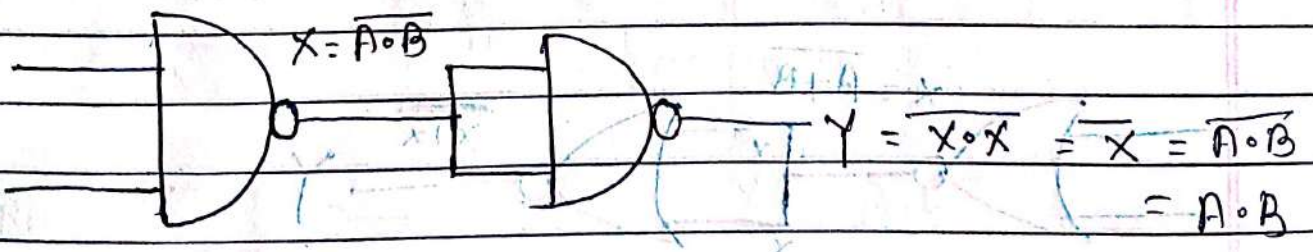
① AND gate from NAND Gate





NOT gate from NAND gate ~~होगा~~

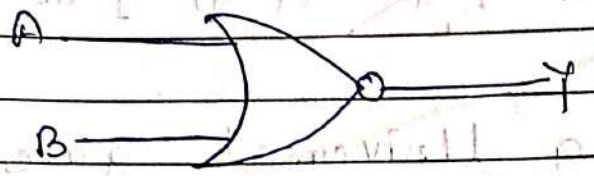
दोनों input same पर देते



NAND + NOT ⟶ AND

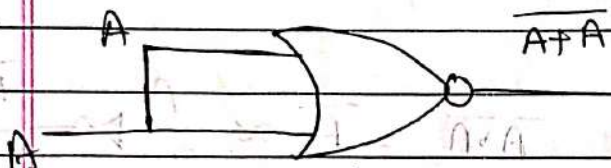
Page:
Date: / /

OR Gate from NOR Gate



logic symbol of NOR Gate

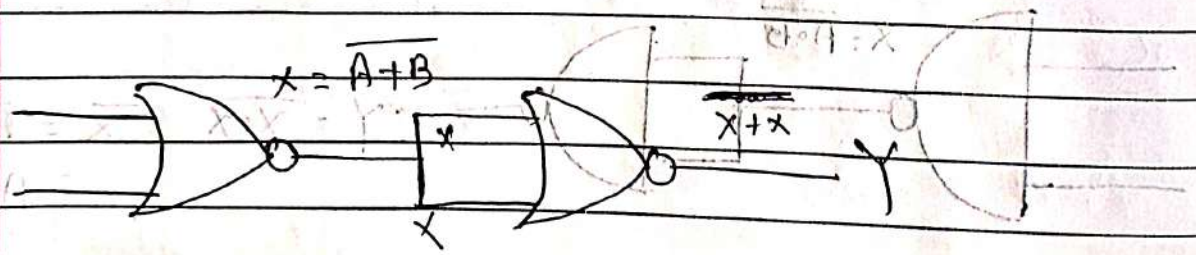
logic expression $Y = \overline{A+B}$



→ NOT Gate from NOR Gate

symbol of NOT Gate

logical expression - $Y = \overline{A+A} = \overline{A}$



NOR + NOT → OR

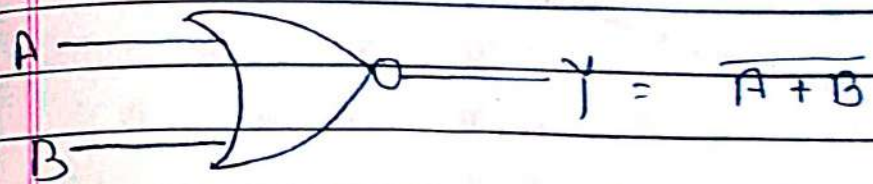
$Y = \overline{x+x} = \overline{x}$

$Y = \overline{A+B}$ $\quad \because x = \overline{A+B}$

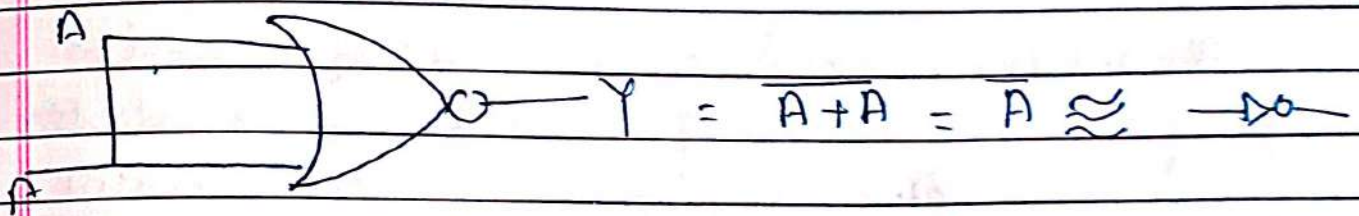
$\boxed{Y = A+B}$

Page:
Date: / /

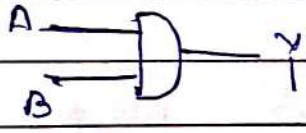
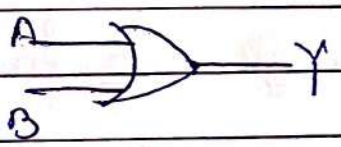
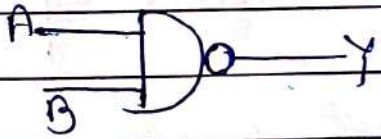
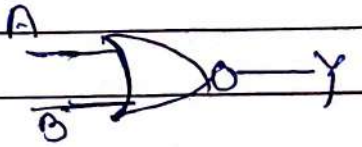
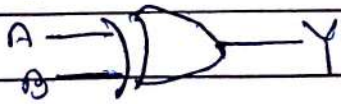
NOT Gate from NOR Gate



Symbol of NOR Gate



NOT Gate from NOR Gate

Gate	Symbol	Logic expression	Gate
AND		$Y = A \cdot B$	$\cdot$
OR		$Y = A + B$	$+$
NAND		$Y = \overline{A \cdot B}$	$\bar{\cdot}$
NOT		$Y = \overline{A}$ / A'	$-$
NOR		$Y = \overline{A+B}$	$\bar{+}$
Ex-OR		$Y = \overline{A \oplus B}$	$\oplus$
Ex-NOR		$Y = A \odot B$	$\odot$

Unit : 2

Boolean Algebra

Page:
Date : / /

Decimal की value को Variable कहते है।
Binary की value को literal कहते है।

जार्ज बुले ने सर्वप्रथम 1854 में लॉजिक व गणित के मध्य सम्बंध व्यक्त कर एक नवीन प्रकार के बीजगणित का अविष्कार किया जिसे boolean Algebra कहते है।

Boolean algebra is a branch of mathematics that deals with operation on logical values with binary variable.

The boolean variables are represented as binary numbers to represent truths: 1 = true and 0 = false

Boolean algebra deals with logical operations.

BOOLEAN'S LAWS

(i) क्रमविनिमेय नियम (Commutative laws)

(ii) साहचर्य नियम (Associative laws)

(iii) वितरण के नियम (Distributive laws)

AND Operation

Law	Axiom
$A \cdot 0 = 0$	$1 \cdot 0 = 0$
$A \cdot 1 = A$	$0 \cdot 0 = 0$
$A \cdot A = A$	$0 \cdot 1 = 0$
$A \cdot \bar{A} = 0$	$1 \cdot 1 = 1$

[A = literal]

OR Operation

Law	Axiom
$A + 0 = A$	$0 + 0 = 0$
$A + 1 = 1$	$1 + 0 = 1$
$A + A = A$	$0 + 1 = 1$
$A + \bar{A} = 1$	$1 + 1 = 1$

NOT Operation / Complement law

Law	Axiom
$\bar{\bar{A}} = A$	$\bar{0} = 1$
	$\bar{1} = 0$

28/4/23

Page:
Date: / /

Laws

(1) Commutative law - Any binary operation which satisfies the following expression is referred to as commutative operation

OR → $A + B = B + A$

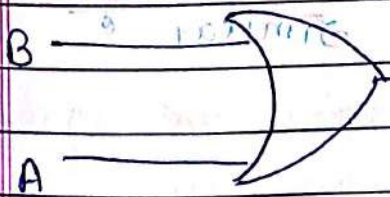
AND → $A \cdot B = B \cdot A$

यह position या order change करती है।
इसमें more minimum literal 2 या 2 से ज्यादा होते है।

Symbol : $A + B = B + A$

L.H.S → symbol of $A + B$

R.H.S → symbol of $B + A$



Truth table

Page:
Date: / /

Truthtable of A+B

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

Truthtable of B+A

B	A	Y
0	0	0
0	1	1
1	0	1
1	1	1

By Comparing output of both truthtable we can say Hence A+B = B+A

चूंकि A+B output का truthtable और B+A के Truth table का output similar है इसलिए यह सिद्ध होता है कि

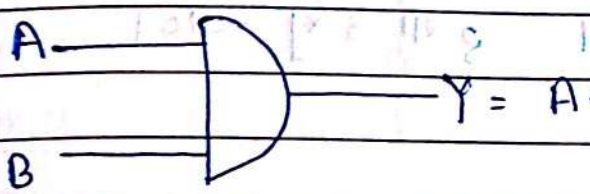
$$A + B = B + A$$

Page:
Date: / /

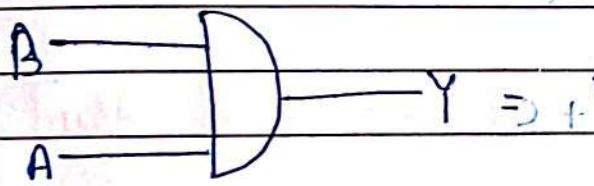
Homework

(ii) $A \cdot B = B \cdot A$

L.H.S → Symbol of $A \cdot B$



R.H.S → Symbol of $B \cdot A$



Truthtable of $A \cdot B$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Truthtable of $B \cdot A$

B	A	Y
0	0	0
0	1	0
1	0	0
1	1	1

By comparing output of both truthtable we can see $A \cdot B = B \cdot A$

अथवा $A \cdot B$ के truthtable का output और $B \cdot A$ के truthtable का output

Similar से प्रमाणित किया होता है कि

$$A \cdot B = B \cdot A$$

(2) Associative law :-

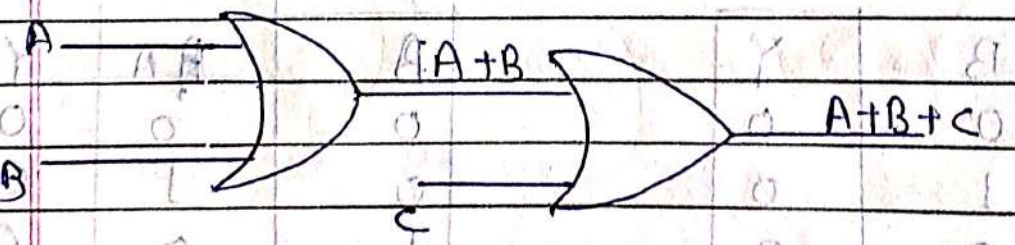
यह literal का grouping होता है जबकि minimum literal 3 या 3 से ज्यादा होते है।

(i) $A + (B+C) = (A+B) + C$

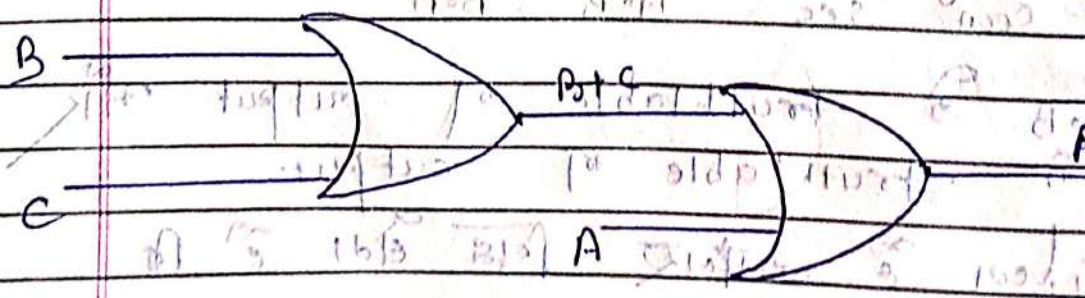
(ii) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

(i) $A + (B+C) = (A+B) + C$

L.H.S. ⟶ symbol of $(A + (B) + C)$



R.H.S ⟶ symbol of $(A+B) + C$ $A + (B+C)$



Page:
Date: / /

Truth table of $(A+B)+C$

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

Truth table of $A + (B+C)$

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

By Comparing output of both truth table we can see $(A+B)+C = A+(B+C)$

दोनों truth table के output की तुलना करने पर यह सिद्ध होता है कि

$$A+(B+C) = (A+B)+C$$

Page:
Date : / /

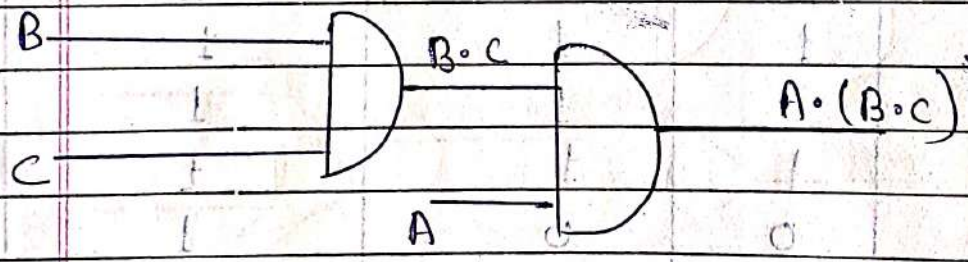
Proved that

$(A+B)+C = A+(B+C)$

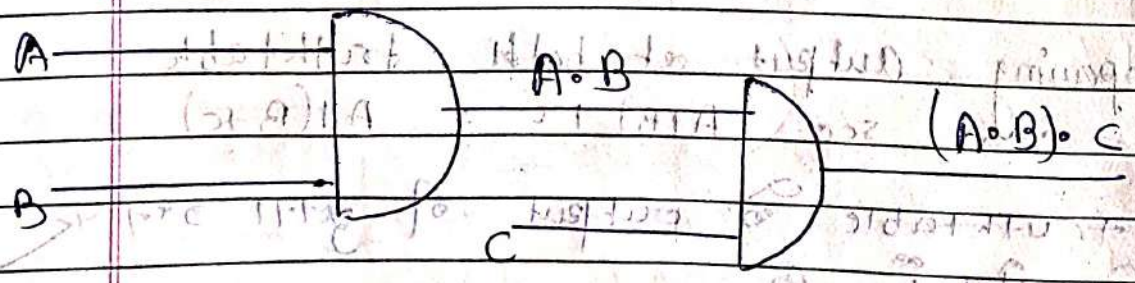
Homework

(ii) $A\cdot(B\cdot C) = (A\cdot B)\cdot C$

L.H.S. Symbol → of $A\cdot(B\cdot C)$



R.H.S → Symbol of $(A\cdot B)\cdot C$



Page:
Date: / /

Truth table of $A \cdot (B \cdot C)$

A	B	C	$B \cdot C$	$A \cdot (B \cdot C)$
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

Truth table of $(A \cdot B) \cdot C$

A	B	C	$A \cdot B$	$(A \cdot B) \cdot C$
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

चूकि दोनों truth table के output समान हैं अतः सिद्ध होता है कि

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

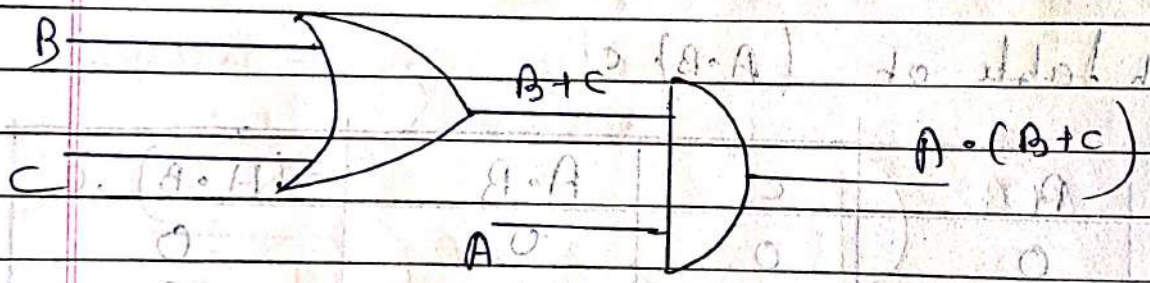
(3) Distributive law

(i) $A \cdot (B + C) = A \cdot B + A \cdot C$

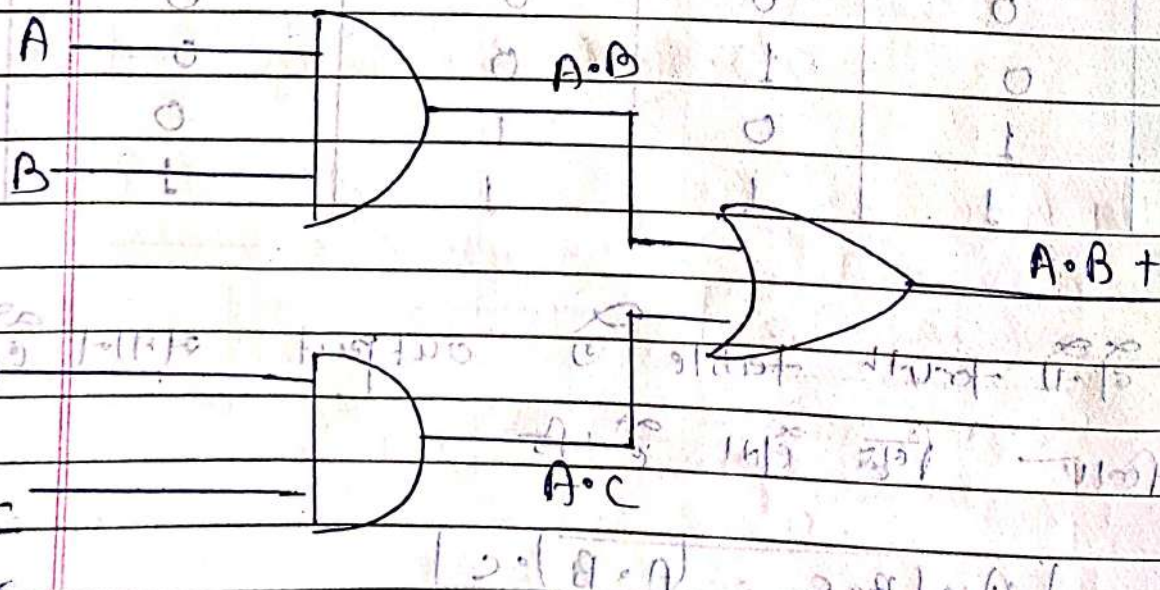
(ii) $A + (B \cdot C) = (A + B) \cdot (A + C)$

[i] $A \cdot (B + C) = A \cdot B + A \cdot C$

L.H.S → Symbol of $A \cdot (B + C)$



R.H.S Symbol of $A \cdot B + A \cdot C$



Page:
Date: / /

Truth table of $A \cdot (B + C)$

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

Truth table of $A \cdot B + A \cdot C$

A	B	C	A · B	A · C	A · B + A · C
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
0	1	1	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

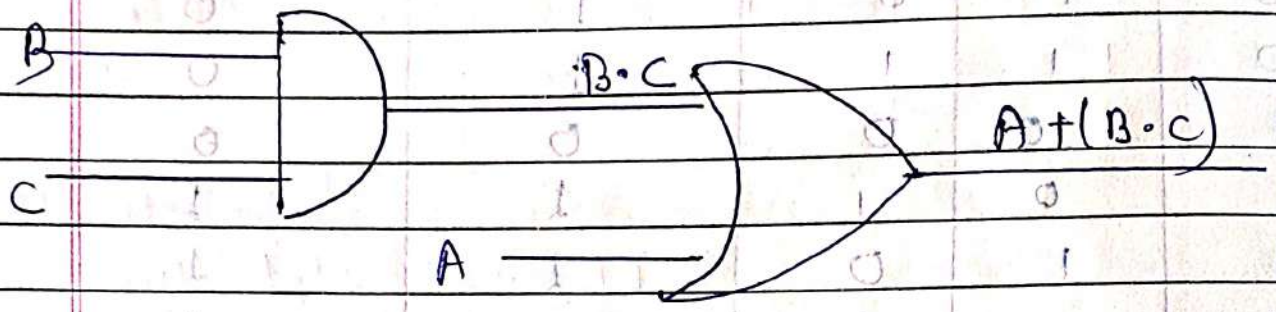
दोनों truth table के output की तुलना

करने पर मिल होता है कि

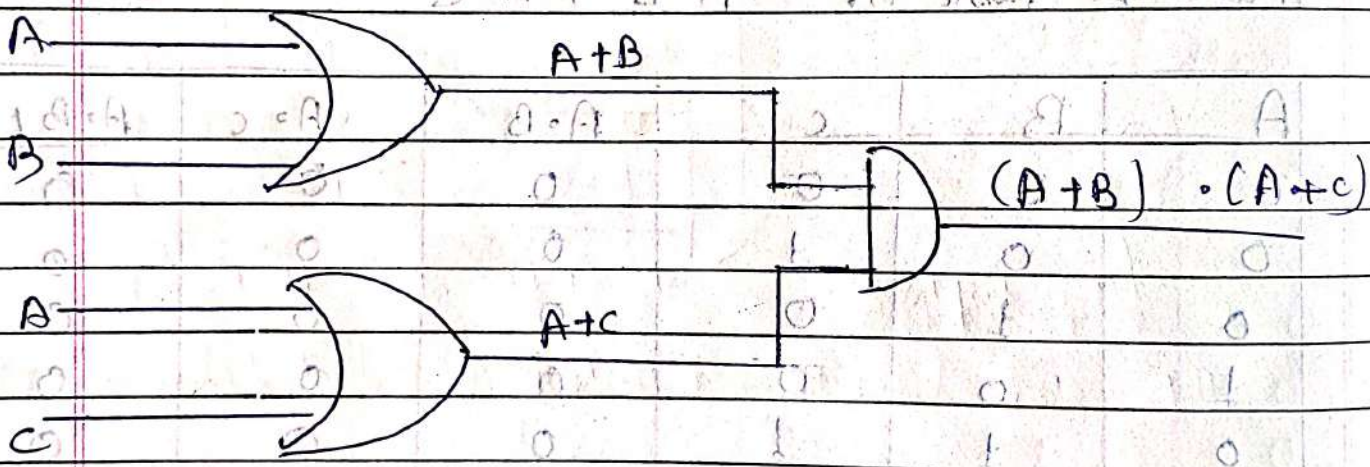
$$A \cdot (B + C) = A \cdot B + A \cdot C$$

(ii) $A + (B \cdot C) = (A+B) \cdot (A+C)$

L.H.S → Symbol of $A+(B \cdot C)$



R.H.S Symbol of $(A+B) \cdot (A+C)$



Truth table of $A+(B \cdot C)$

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	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Page:
Date: / /

Truth table of $(A+B) \cdot A+C$

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

दोनों truth table के output B जुल रहे अतः ये सिद्ध होता है कि

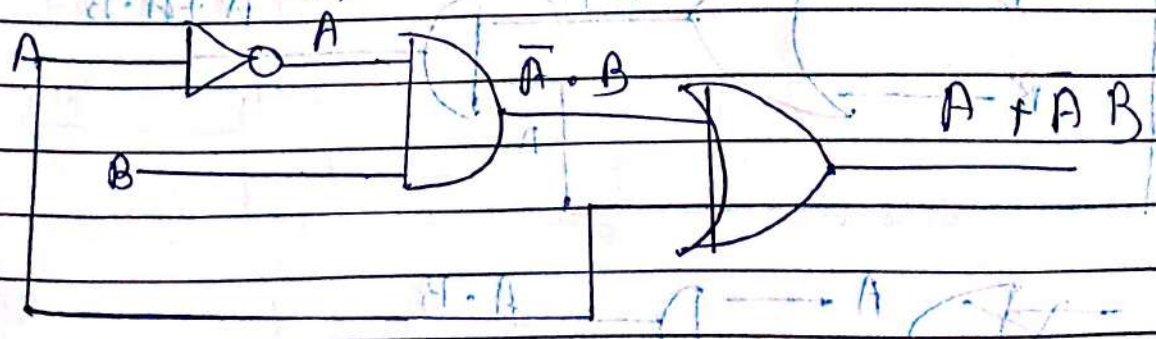
$$A + (B \cdot C) = (A+B) \cdot (A+C)$$

(4) Redundant laws

or Redundant litteral

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

L.H.S →



R.H.

Truth table of $A + \bar{A} \cdot B$

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

Truth table of $A + B$

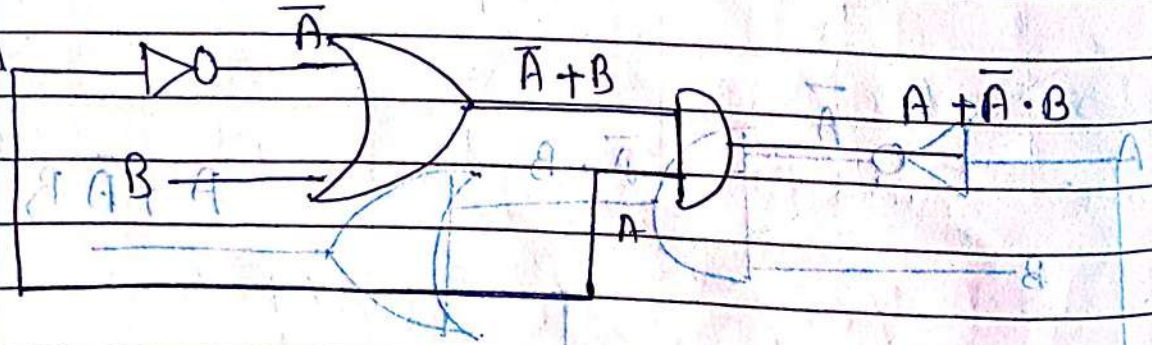
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

ચુકિ બંનો truth table ના output બરાબર થતા હોવાથી

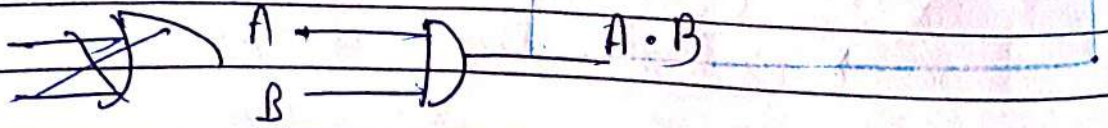
$A + \bar{A} \cdot B = A + B$ સિદ્ધ થાય છે.

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

L.H.S



R.H.S



Page:
Date: / /

Truth table of $A(\bar{A}+B)$

A	B	$\bar{A}$	$\bar{A}+B$	$A.(\bar{A}+B)$
0	0	1	1	0
1	0	0	0	0
0	1	1	1	0
1	1	0	1	1

Truth table of $A \cdot B$

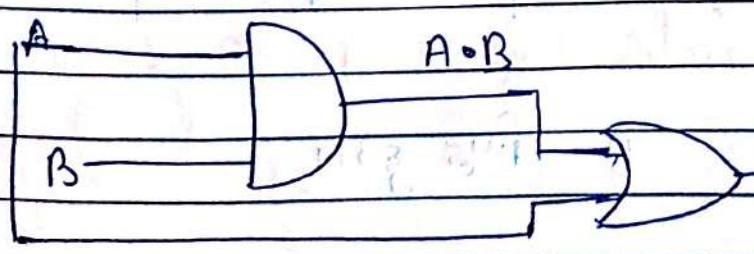
A	B	$A \cdot B$
0	0	0
1	0	0
0	1	0
1	1	1

चूंकि दोनों truth table का output समान है इसलिए लिख सकते हैं $A(\bar{A}+B) = A \cdot B$

(5) Absorption law :

1) $A + A \cdot B = A$ ~~$= A$~~

L.H.S



symbol

Page:
Date : / /

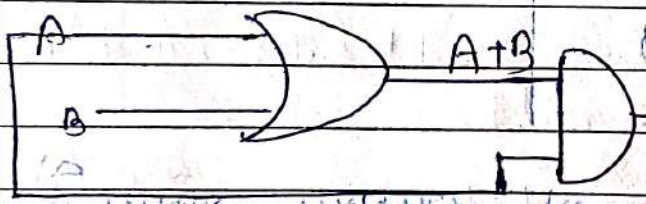
Truth table of $A + A \cdot B$

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

चूँकि $A + A \cdot B$ और A दोनों का output समान है इसलिए $\boxed{A + A \cdot B = A}$

(2) $A \cdot (A + B) = A$

L.H.S



Truth table of $A \cdot (A + B)$

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

चूँकि $A \cdot (A + B) = A$ सिद्ध हुआ

Page: _____
Date: / /

Absorption law

Proved that

(1) $A + A \cdot B = A$

L.H.S

Using distributive law

[$A(B+C) = A \cdot B + A \cdot C$]

$= A \cdot 1 + A \cdot B$ [Using AND law $A \cdot 1 = A$]

$= A(1+B)$ [Using distributive law]
[Using OR law $1+B = 1$]

$= A \cdot 1$ [Using AND law $A \cdot 1 = A$]

$= A$ Proved that (L.H.S = R.H.S)

(2) $A(A+B) = A$

L.H.S

$= A \cdot A + A \cdot B$ [Using distributive law]

$= A + A \cdot B$ [Using AND law $A \cdot A = A$]

$= A(1+B)$ $A \cdot 1 + A \cdot B$ [Using AND law $A \cdot 1 = A$]

$= A(1+B)$ [Using distributive law]

$= A \cdot 1$ [Using OR law $1+B = 1$]

$= A$ [Using AND law $A \cdot 1 = A$]

Proved that L.H.S = R.H.S

Page:
Date: / /

Proved that distribution law

(3) $A + (B \cdot C) = (A+B) \cdot (A+C)$

R.H.S $(A+B) \cdot (A+C)$

$= AA + AC + AB + BC$

[Using AND law $A \cdot A = A$]

$= A + AC + AB + BC$

[Using AND law $A \cdot 1 = A$]

$= A \cdot 1 + AC + AB + BC$

[Using distributive law

$= A(1 + C + B) + BC$

[Using OR law $1 + C + B = 1$]

$= A + B \cdot C$

Proved that L.H.S = R.H.S

Redundant

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

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

[Using distributive law $A + (B \cdot C) = (A+B)(A+C)$]

$= 1 \cdot (A + B)$ [Using OR law $A + \bar{A} = 1$

$= A + B$ [Using AND law $1 \cdot A = A$

Page: _____
Date: / /

$\therefore A + B \cdot C = (A+B)(A+C)$

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

$= A\bar{A} + A \cdot B$ [Using distributive law $A(B+C) = A \cdot B + A \cdot C$]

$= 0 + A \cdot B$ [Using AND law $A \cdot \bar{A} = 0$]

$= A \cdot B$ Proved

(6) Consensus Law

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

L.H.S

$= AB + \bar{A}C + BC \cdot 1$ [Using OR law $A + \bar{A} = 1$]

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

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

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

$= AB(1+C) + \bar{A}C(1+B)$ [Using AND law $1+B = 1$]

$= AB + \bar{A}C$

Proved that

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

(7) Demorgan's Theorem

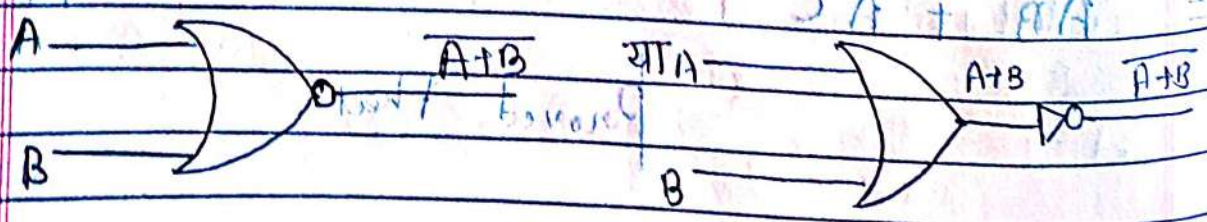
$\overline{A\cdot B\cdot C\cdot D} = \bar{A}+\bar{B}+\bar{C}+\bar{D}$

(1) $\overline{A+B} = \bar{A}\cdot\bar{B} \rightarrow \overline{AB+BC+CD} = \overline{AB}+\overline{BC}+\overline{CD}$

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

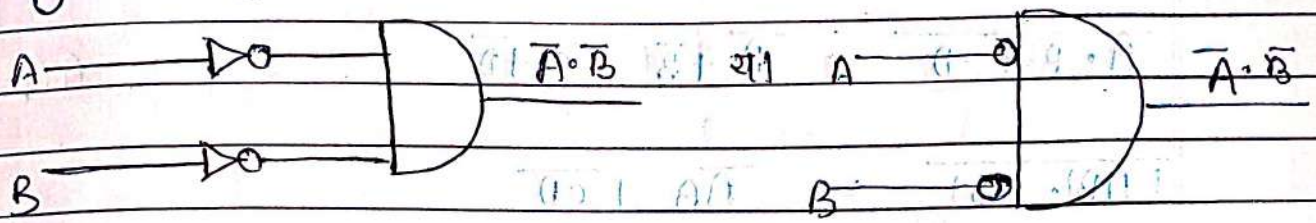
(7) $\overline{A+B} = \bar{A}\cdot\bar{B}$

L.H.S $\longrightarrow$ Logical diagram of $\overline{A+B}$



Page:
Date: / /

Logical diagram of $\overline{A} \cdot \overline{B}$



Truth table of $\overline{A+B}$

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

Truth table of $\overline{A} \cdot \overline{B}$

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

इस law के अनुसार दो या दो से ज्यादा Literal / variable के OR का complement (NOR) literals के complement के AND operation के equal होता है।

या

NOR Gate का output inverted / Bubbled input And Gate के equal होता है।

Theorem : 2

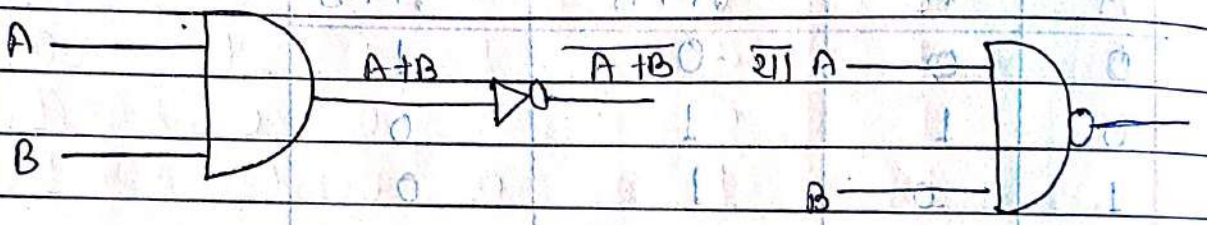
$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

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

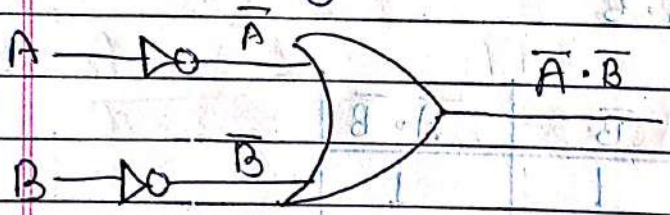
$$\overline{(AB) \cdot (CD)} = \overline{AB} + \overline{CD}$$

L.H.S → logical diagram of $\overline{A \cdot B} = \overline{A} + \overline{B}$

L.H.S.



logical diagram of $\overline{A} + \overline{B}$



Truth table of $\overline{A \cdot B}$

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

Page:
Date: / /

Truth table of $\overline{A} + \overline{B}$

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

इस low के अनुसार दो या दो से ज्यादा literal के AND का एक साथ complement (NAND) literals के individual complement के OR operation के equal होता है।

NAND Gate is similar to bubble into OR Gate

Duality theorem

In well expression 0 becomes 1 ; 1 becomes 0 bother AND becomes OR, & OR operation or becomes AND operation

$(A+B)_d = A \cdot B$

इसमें literal का complement नहीं होता।

$f = A + BC$ ← boolean function

complement of boolean function

$f_c = \overline{A + BC}$ $= \overline{A} \cdot \overline{BC} = \overline{A}(\overline{B} + \overline{C})$

$f_d = A \cdot (B + C)$ ← dual is boolen

PROBLEMS

5.1 Verify by the truth table method.

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

Truth table of $A + \bar{A}B + AB$

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

Truth table of $A+B$

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

∴ $A + \bar{A}B + AB$ और $A+B$ दोनों के Truth table output same होगा।

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

Truth table of $(A + \bar{B})(\bar{A} + B)$

A	B	$\bar{A}$	$\bar{B}$	$A+\bar{B}$	$\bar{A}+B$	$(A+\bar{B})(\bar{A}+B)$
0	0	1	1	1	1	1
0	1	1	0	0	1	0
1	0	0	1	1	0	0
1	1	0	0	1	1	1

Truth table of $AB + \bar{A}\bar{B}$

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

∵ $(A+\bar{B})(\bar{A}+B)$ और $AB + \bar{A}\bar{B}$ दोनों के Truth table के output same है

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

5.2 Reduce the following Boolean expressions

(a) AABBC

By applying AND's law

$[A \cdot A = A]$

$[B \cdot B = B]$

so simplified expression is

∴ ABC — Ans

Page:
Date : / /

b) $ABCBAC$

By applying AN's law

$A \cdot A = A$

$B \cdot B = B$

$C \cdot C = C$

Hence the simplified expression is

ABC ~~Ans~~

(c) $\bar{A} \cdot A \cdot A$

By Appling AN's law

$\bar{A} \cdot A = 0$

$A \cdot 1 = A$

so

$A \cdot 0 = 0$

0 is the simplified expression.

(d) $AB\bar{C}\bar{A}\bar{B}$

By appling AND's law

$A\bar{A} = 0$

$B\bar{B} = 0$

Hence simplified expression is

$= 0 \cdot C$

$= 0$ [$0 \cdot C = 0$ using AND's law]

Ans

Page:
Date: / /

e) $AB\bar{B}AC$

By appling AND's law

$A \cdot A = A$

$B\bar{B} = 0$

Hence the simplified expression is

$= A \cdot 0 \cdot C$

$= 0 \cdot C$ [$A \cdot 0 = 0$ using AND law]

$= 0$ Ans

(f) $BB\bar{B}B$

By using AND's law

$B \cdot B = B$

$B \cdot \bar{B} = 0$

$= B \cdot 0$ [$B \cdot 0 = 0$ using AND law]

$= 0$ Ans

5.3 Reduce the following Boolean expressions:

(a) $P + Q + P$

By appling OR law

$= P + P = P$

so the simplified expression is

$= P + Q$ Ans

(b) $P + Q + R + \bar{P}$

By appling OR law

$\Rightarrow P + \bar{P} = 1$

So simplified expression is

$= 1 + Q + R$ Ans.

[Using OR law $1 + Q = 1$]

$= 1 + R$ [Using OR law]

$= 1$ Ans.

(c) $P + Q + R + R + \bar{R}$

By appling OR law

$= [R + R = R]$

So simplified expression is

$= P + Q + R + \bar{R}$

$[R + \bar{R} = 1$ using OR law$]$

$= P + Q + 1$ $[Q + 1 = 1$ using OR law$]$

$= P + 1$ $[P + 1 = 1$ using OR law$]$

$= 1$ Ans

(d) $P + Q + R + 1$

$[R + 1 = 1$ using OR law$]$

$= P + Q + 1$ $[Q + 1 = 1$ using OR law$]$

$= P + 1$

$= 1$ Ans

(e) $0 + P + \bar{Q} + 1$

$= P + \bar{Q} + 1$ [$0 + P = P$ using OR law]

$= P + 1$ [$\bar{Q} + 1 = 1$ using OR law]

$= 1$ Ans [$P + 1 = 1$ using OR law]

(f) $P + P + P + P$

$= P + P + P$ [$P + P = P$ using OR law]

$= P + P$

$= P$ Ans

5.4 Reduce the following Boolean expressions.

(a) $X\bar{Y} + XY + \bar{X}Y + Y$

$= X\bar{Y} + XY + Y$ [$\because Y + Y = Y$ using OR law]

$= X\bar{Y} + XY + Y$ [$\because X\bar{Y} + XY = XY$ using OR law]

$= XY + Y$

$= Y(X + 1)$ [$X + 1 = 1$ using OR law]

$= Y$ Ans

(b) $X\bar{X} + Y\bar{Y}$

[$X \cdot \bar{X} = 0$ using AND law]

$= 0 + Y\bar{Y}$ [$Y\bar{Y} = 0$ using AND law]

$= 0 + Y \cdot 0$ [$Y \cdot 0 = 0$ using AND]

$= 0 + 0$

$= 0$ Ans

(c) $X\bar{X}Y + XY\bar{Y}$

[$X \cdot \bar{X} = 0$ using AND law]

$= 0Y + X \cdot 0$

$= 0 + 0 = 0$ Ans

(d) $XY + XY\bar{Y}$

[$\because Y\bar{Y} = 0$ using AND law]

$= XY + X \cdot 0$

[$\because X \cdot 0 = 0$ using AND law]

$= XY + 0$

$= XY$ Ans

(e) $XY + X\bar{Y} + X\bar{Y}$

[$\because XY + XY = XY$ using AND's law]

$= XY + X\bar{Y}$ [Distribution law

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

$= X \cdot 1$ [$\because Y + \bar{Y} = 1$

$= 1$ Ans [$\because X \cdot 1 = X$]

(f) $XYZ + YYZ + \bar{Y}Z + X\bar{Y}$

[$\because YY = Y$ using AND law]

$= XYZ + YZ + \bar{Y}Z + X\bar{Y}$

$= XYZ + Z(Y + \bar{Y}) + X\bar{Y}$ [$Y + \bar{Y} = 1$ using OR law]

$= XYZ + Z + X\bar{Y}$

$= YZ(X + 1) + X\bar{Y}Z + X\bar{Y}$

$= YZ + \bar{Y}Z + X\bar{Y}$ [$X + 1 = 1$ using OR law]

$= Z(Y + \bar{Y}) + X\bar{Y}$ [$Y + \bar{Y} = 1$ using OR]

$= Z + X\bar{Y}$ Ans

5.5 Reduce the following Boolean expressions

(a) $X(\bar{X} + YZ)$

[using Distribution law]

$= X\bar{X} + XYZ$

$= 0 + XYZ$ [$X\bar{X} = 0$ using AND law]

$= XYZ$ Ans

Page:
Date: / /

(b) $x(yz + \bar{y}z)$

$= xyz + x\bar{y}z$

$= xz(y + \bar{y})$ [$\because y + \bar{y} = 1$ using OR law]

$= xz \cdot 1$

$= x \cdot z$ Ans

(c) $x(\bar{x}y + \bar{x}z)$

$= x\bar{x}y + x\bar{x}z$ [$\because x\bar{x} = 0$ using AND law]

$= 0 \cdot y + 0 \cdot z$ [$\because 0 \cdot y = 0$ using AND law]

$= 0$ Ans

(d) $AAB(\bar{A}BC + BBC)$

[$\because AA = A$ using AND law]

$= AB(\bar{A}BC + BC)$

$= ABBC(\bar{A} + 1)$ [$\because A\bar{A} + 1 = 1$ using OR law]

$= ABBC$ [$\because B \cdot B = B$ using AND law]

$= ABC$ Ans

5.6 Reduce the following Boolean expression:

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

[Using distribution law]

$= AB + AB + AC + B\bar{B} + BD$

$= AB + AC + B\bar{B} + BD$ [$\because AB + AB = AB$ OR law]

[$\because B\bar{B} = 0$

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

$= A(B+C) + BD$ Ans

(b) $(x+y+z)(\bar{x}+\bar{y}+\bar{z})x$

$= (x+y+z)(x\bar{x}+x\bar{y}+x\bar{z})$ [using $x\bar{x}=0$]

$= (x+y+z)(0+x\bar{y}+x\bar{z})$

$= (x+y+z)(x\bar{y}+x\bar{z})$

$= x x\bar{y} + x x\bar{z} + y\bar{y}x + x\bar{z}y + z x\bar{y} + z\bar{z}x$

$=$ [using $x \cdot x = x$, $y\bar{y} = 0$

$= x\bar{y} + x\bar{z} + 0 \cdot x + xy\bar{z} + x\bar{y}z + 0 \cdot x$

$= x\bar{y} + x\bar{z} + xy\bar{z} + x\bar{y}z$

$= x\bar{y}(1+z) + x\bar{z}(1+y)$ [$1+y=1$]

$= x\bar{y} + x\bar{z}$

$= x(\bar{y}+\bar{z})$ Answer

$= x\cdot\overline{yz}$

(c) $ABEF + AB\bar{E}\bar{F} + \bar{A}B\bar{E}F$

$= AB[EF + \bar{E}\bar{F}] + \bar{A}BEF$

[using $A+\bar{A}=1$ OR law]

$= AB + \overline{AB}\cdot EF$

$= (AB+\overline{AB})(AB+EF)$ [using distribution law]

$= 1(AB+EF)$ [$\because A+\bar{A}=1$]

$= AB + EF$ Answer

(d) $ABC[AB + \bar{C}(BC+AC)]$

$= ABC[AB + BC\bar{C} + AC\bar{C}]$ [$\because C\bar{C}=0$]

$= ABC[AB + B\cdot 0 + A\cdot 0]$

$= ABC\,AB$

$= AA\,BBC$ [using $AA = A$]

$= ABC$ Ans

(e) $\bar{A}B + \bar{A}B\bar{C} + \bar{A}BCD + \bar{A}B\bar{C}\bar{D}E$

$= \bar{A}B(1 + \bar{C}) + \bar{A}B(CD + \bar{C}\bar{D}E)$ [using $1 + \bar{C} = 1$ using OR

$= \bar{A}B \cdot 1 + \bar{A}B(CD + \bar{C}\bar{D}E)$

$= \bar{A}B + \bar{A}BCD + \bar{A}B\bar{C}\bar{D}E$

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

$= \bar{A}B(1+C)(1+D) + \bar{A}B\bar{C}DE$ [$1 + C = 1$]

$= \bar{A}B + \bar{A}B\bar{C}DE$ [Distribution law]

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

$= \bar{A}B(1 + \bar{C}DE)$ [Distri. $A + BC = (A+B)(A+C)$]

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

$= \bar{A}B \cdot 1 \cdot 1 \cdot 1$ [$1 + C = 1$]

$= \bar{A}B$ Answer

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

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

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

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

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

$= AA + AC + A\bar{B} + \bar{B}C + B$

$= A + AC + \cdot$

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

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

$= A + B + C$

(g) $\overline{B}\,\overline{C}D + \overline{B+C+D} + \overline{B}\,\overline{C}\,\overline{D}E$

$= \overline{B}\,\overline{C}D + \overline{B}\,\overline{C}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D}\overline{E}$

$= \overline{B}\,\overline{C}(D + \overline{D}) + \overline{B}\,\overline{C}\,\overline{D}\overline{E}$

$= \overline{B}\,\overline{C} + \overline{D}\overline{E}\overline{B}\overline{C}$

$= \overline{B}\,\overline{C}(1 + \overline{D}\overline{E})$

$= \overline{B}\,\overline{C}(1+\overline{D})(1+\overline{E})$

$= \overline{B}\,\overline{C}$ Answer

(h) $(wx + w\overline{y})(x+w) + wx(\overline{x}+\overline{y})$ $[x\overline{x}=0]$

$= (wx + w\overline{y})(x+w) + wx\overline{x} + w\overline{x}\overline{y}$

$= wxx + w\overline{y}x + wwx + w\overline{y}w + wx\overline{x} + w\overline{x}\overline{y}$

$= wx + wx\overline{y} + wx + w\overline{y} + wx\overline{y}$ $[A+A=A]$

$= wx + wx\overline{y} + w\overline{y}$

$= wx(1+\overline{y}) + w\overline{y}$

$= wx + w\overline{y}$

$= w(x+\overline{y})$ Ans

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

$= A\overline{B} + \overline{A}\overline{C} + A\overline{B}CAB + A\overline{B}CC$

$= A\overline{B} + \overline{A}\overline{C} + 0 + A\overline{B}C$

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

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

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

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

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

$= AB + AC + \overline{A}\overline{C}$

$= AB + 1$

$= (A+1)(B+1)$

$= 1$ Answer

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

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

$= \bar{B}\bar{C} + BC$

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

$= \bar{C}$ Answer

5.8 Solve the Demorgan's law

(1) $\overline{P(Q+R)}$

$= \bar{P} + \overline{(Q+R)}$

$= \bar{P} + \bar{Q}\cdot\bar{R}$ Ans

(2) $\overline{(P+\bar{Q})(\bar{R}+S)}$

$= \overline{(P+\bar{Q})} + \overline{(\bar{R}+S)}$

$= \bar{P}\cdot Q + R\cdot\bar{S}$ Ans

(3) $\overline{(A+B)(C+D)(E+F)(G+H)}$

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

$= (A+B) + (C+D) + (E+F) + (G+H)$

$= \overline{A+B}\cdot\overline{C+D} + \overline{E+F}\cdot\overline{G+H}$

$= (\bar{A}\cdot\bar{B})(\bar{C}\cdot\bar{D}) + (\bar{E}\cdot\bar{F})\cdot(\bar{G}\cdot\bar{H})$ Ans

(4) $(\bar{A}+B+\bar{C}+D)(0+\bar{A}BC\bar{D})$

$= \overline{\bar{A}+B+\bar{C}+D} + \overline{0+\bar{A}BC\bar{D}}$

$= A\cdot\bar{B}\cdot C\cdot\bar{D} + \bar{0}\cdot\overline{\bar{A}BC\bar{D}}$

$A\bar{B}C\bar{D} + 1\cdot A + \bar{B} + \bar{C} + D$

$A\bar{B}C\bar{D} + A + \bar{B} + \bar{C} + D$ Ans

All theorem

(1) Commutative law :

(1) $A \cdot B = B \cdot A$

(2) $A + B = B + A$

(2) Associative law :

(1) $A + (B + C) = (A + B) + C$

(2) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

(3) Distributive law

(1) $A \cdot (B + C) = AB + AC$

(2) $A + (B \cdot C) = (A + B)(A + C)$

(4) Redundant law

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

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

(5) Absorption law

(1) $A + A \cdot B = A$

(2) $A(A + B) = A$

(6) Consensus law

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

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

Page:
Date: / /

(7) Demorgan's theorem

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

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

Complement & dual form of all theorm

(1) $A \cdot B = B \cdot A$ [Boolean function]

complement form

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

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

dual form

$A \cdot B = B \cdot A$

$A + B = B + A$ Ans

(2) $A + B = B + A$ [Boolean function]

Complement form

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

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

dual form

$A + B = B + A$

$A \cdot B = B \cdot A$ Ans

(3) $A + (B+C) = (A+B) + C$

Complement form

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

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

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

dual form

$A + (B+C) = (A+B) + C$

$A \cdot (B \cdot C) = (A \cdot B) \cdot C$ Ans

(2) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

Complement form

$= \overline{A \cdot (B \cdot C)} = \overline{(A \cdot B) \cdot C}$

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

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

dual form

$A + (B + C) = (A + B) + C$

Distributive law

(1) $A \cdot (B \cdot C) = A \cdot B + B \cdot C$

Completent form

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

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

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

dual form

$= A \cdot (B \cdot C) = A \cdot B + B \cdot C$

$= A + (B + C) = A + B \cdot B + C$

(2) $A + (B \cdot C) = (A + B)(A + C)$

Complement form

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

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

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

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

Dual form

$= A + (B \cdot C) = (A + B)(A + C)$

$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$

Page:
Date: / /

Redundant law

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

Complement form

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

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

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

dual form

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

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

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

Complement

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

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

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

dual form

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

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

Absorption law

(1) $A + A \cdot B = A$

Complement

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

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

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

dual form

$= A \cdot A + B = A$

(2) $A + (A + B) = A$

$= \overline{A + (A + B)} = \bar{A}$ Compalement form

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

Dual form $= A + (A \cdot B) = A$

(6) Consensus law

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

$= \overline{A \cdot B + \bar{A}C + BC} = \overline{AB + \bar{A}C}$ Complement form

$= \bar{A} \cdot \bar{B} \cdot \bar{A}\bar{C} \cdot \bar{B}\bar{C} = \bar{A}\bar{B} \cdot \bar{A}\bar{C}$

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

Dual form

$A \cdot B + \bar{A}C + BC = AB + \bar{A}C$ dual form

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

(I)

Complement form

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

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

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

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

dual form

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

(7) Demorgan's law

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

Complement form

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

$= A + B = A + B$

dual form

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

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

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

$= A \cdot B = A \cdot B$

dual form

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

Page:
Date: / /

SOP & POS

Functionally Complete set of Operations

SOP & POS → Product of Sum

↓

Sum of product

example of SOP

$XY + YZ$ [पहले दो variable का मेल है मतलब product है दोनो product के बीच Sum है इसलिए SOP]

$= XY \cdot 1 + YZ \cdot 1$

Canonical form :

$XY + YZ$

$= XY \cdot 1 + YZ \cdot 1$ [इनमे पहले product मे हमारे पास (z + z̄) है] और

$= XY(Z + \bar{Z}) + YZ(X + \bar{X})$ [∵ A + A = A]

$= XYZ + XY\bar{Z} + XYZ + \bar{X}YZ$

$= XYZ + XY\bar{Z} + \bar{X}YZ$ Answer : this is conical form

POS Example.

$(X + Y)(Y + Z)$ [इसमे दो sum variables है बीच मे product है]

Canonical form :

$(X + Y + 0)(Y + Z + 0)$

$= (X + Y + Z \cdot \bar{Z})(Y + Z + X \cdot \bar{X})$ [A · Ā = 0, A + Ā = 1]

$= (X + Y + Z)(X + Y + \bar{Z})(Y + Z + X)(\bar{X} + Y + Z)$

$= (X + Y + Z)(X + Y + \bar{Z})(\bar{X} + Y + Z)$ Answer

Combination Truth table

Decimal	X	Y	Z	SOP: Product term / min term	POS: Sum term / max term
	x	y	z	f(x, y, z)	F(x+y+z)
0	0	0	0	0 $(\bar{x}\bar{y}\bar{z})$	0 $(x+y+z)$
1	0	0	1	0 $(\bar{x}\bar{y}z)$	1 $(x+y+\bar{z})$
2	0	1	0	0 $(\bar{x}y\bar{z})$	1 $(x+\bar{y}+z)$
3	0	1	1	0 $(\bar{x}yz)$	1 $(x+\bar{y}+\bar{z})$
4	1	0	0	0 $(x\bar{y}\bar{z})$	1 $(\bar{x}+y+z)$
5	1	0	1	0 $(x\bar{y}z)$	1 $(\bar{x}+y+\bar{z})$
6	1	1	0	0 $(xy\bar{z})$	1 $(\bar{x}+\bar{y}+z)$
7	1	1	1	1 (xyz)	1 $(\bar{x}+\bar{y}+\bar{z})$

for SOP

AND

A	B	F(A.B)	
0	0	0	$\bar{A}\bar{B}$
0	1	0	$\bar{A}B$
1	0	0	$A\bar{B}$
1	1	1	AB

for POS

OR

A	B	F(A+B)	
0	0	0	A+B
0	1	1	A+
1	0	1	$\bar{A}$+
1	1	1	$\bar{A}$+

min term का function में = 0 मतलब $\bar{A}$

1 मतलब A

max term का function में = 0 मतलब A

1 मतलब $\bar{A}$

	minterm SOP	maxterm POS
0	$\bar{A}$	A
1	A	$\bar{A}$

Page:
Date: / /

SOP

Product term को min term भी कहते है क्योंकि इसमें सबसे ज्यादा 0 प्राप्त होते है। अर्थात् किसी भी variable का output 0 ही सबसे ज्यादा प्राप्त होता है।

POS

Sum term को max term भी कहते है क्योंकि इसमें सबसे ज्यादा 1 आते है अर्थात् input कुछ भी हो अधिकतर बार 1 ही आते है।

MSB LSB

$F(A, B, C)$ इसमें हैं कि A MSB होगा और C LSB होगा

@ Question इसी method में होंगे।

Sum of product Method

इस विधि में किसी व्यंजक को उसके variable के गुणनफल के योग के रूप में व्यक्त किया जाता है जिसे Sum of Product कहते है।

Decimal	Input A	Input B	Output $Y = A \cdot B$	Fundamental Product
0	0	0	0	$\bar{A}\bar{B}$
1	0	1	0	$\bar{A}B$
2	1	0	0	$A\bar{B}$
3	1	1	1	AB

इसमें 1 मतलब true form और
0 मतलब Complement form

(Product of Sum Method : POS)

इस विधि में सत्य तालिका से उन इनपुट अवस्थाओं को लिया जाता है जिनके संगत output का मान '0' होता है ऐसे input को fundamental sum कहा जाता है।

इन fundamental sum को गुणा या AND प्रक्रिया करने पर POS समीकरण प्राप्त होते है।

fundamental sum प्राप्त करने के लिये variable का मान '0' होने पर उन्हे यथावत् लिया जाता है।

$0 = A$ होगा

और $1 = \bar{A}$

Decimal	Input				Fundamental sum
	A	B	C	Y	
0	0	0	0	0	$A+B+C$
1	0	0	1	1	$A+B+\bar{C}$
2	0	1	0	1	$A+\bar{B}+C$
3	0	1	1	1	$A+\bar{B}+\bar{C}$
4	1	0	0	1	$\bar{A}+B+C$
5	1	0	1	1	$\bar{A}+B+\bar{C}$
6	1	1	0	1	$\bar{A}+\bar{B}+C$
7	1	1	1	1	$\bar{A}+\bar{B}+\bar{C}$

Page:
Date: / /

SOP और POS में अंतर

SOP	POS
1) इसमें '1' output के लिये fundamental product लिया जाता है।	इसमें '0' output के लिए fundamental sum लिया जाता है।
2) 0 → variable का Complement 1 → true form	0 → True form 1 → complement of variable
यह AND-OR परिपथ बनाता है।	यह OR-AND बनाता है।
प्रत्येक fundamental product को minterm कहते है क्योंकि जिसे 'm' से दर्शाते है।	प्रत्येक fundamental product को maxterm कहते है, जिसे 'M' कहते है।
इसमें चिन्ह "Σ" होता है	इसमें चिन्ह "Π" होता है।

Q. Simplify the expression to get the canonical SOP

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

$Y = AB \cdot 1 + A\bar{C} \cdot 1 + BC \cdot 1$

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

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

$Y = ABC + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}BC$ Ans

Each term contains all the literals

Page:
Date: / /

Q. Convert the expression $Y = (A+B)(A+C)(B+\bar{C})$ into canonical POS form

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

$Y = (A+B+0)(A+C+0)(B+\bar{C}+0)$

$Y = (\underbrace{A+B+C\bar{C}}_{X})(\underbrace{A+C+B\bar{B}}_{Y.Z})(B+\bar{C}+A\bar{A})$

$\because$ Using distribution law $X+YZ = (X+Y)(X+Z)$

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

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

Each term contains all the literals

Canonical form Ans

Q Convert following equation into standard SOP form (Canonical form)

$Y = \bar{A} + B\bar{C}D$

$Y = \bar{A}\cdot 1 + B\bar{C}D(1)$

$\bar{A}(A+A) + AB$

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

$= \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + \bar{A}BC\bar{D}$

$= \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}D$

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

$Y = \bar{A}BCD + \bar{A}BC\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}CD$
$+ \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}$
$+ AB\bar{C}\bar{D}$

$[\because A+A = A]$

This is the required expression in the canonical form SOP

Page:
Date: / /

(2) Convert following equation into standard pos form (canonical)

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

$Y = (\bar{A} + B + 0)(B + \bar{C} + 0) + (\bar{A} + \bar{C} + 0)$

$Y = (\underbrace{\bar{A} + B}_{x} + \underbrace{C\bar{C}}_{yz})(B + \bar{C} + A\bar{A}) + (\bar{A} + \bar{C} + B\bar{B})$

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

Convert following equation to canonical SOP form

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

$Y = AB + AAC + B\bar{B}\bar{C} + ABC\bar{C}$

$Y = AB + AC + B\bar{C} + 0$

$Y = AB \cdot 1 + AC \cdot 1 + B\bar{C}$

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

$Y = ABC + AB\bar{C} + ABC + A\bar{B}C + AB\bar{C} + \bar{A}B\bar{C}$

$Y = ABC + AB\bar{C} + A\bar{B}C + \bar{A}B\bar{C}$

This is required canical sop form

Convert following expression into standard SOP form

$Y = A + B$

$Y = A \cdot 1 + B \cdot 1$

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

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

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

This is Canonical SOP form

K-Map

Karnaugh Map

Boolean function को सरल करने हेतु demorgans theoram एवं बीजगणित विधियों के अलावा एक अन्य महत्वपूर्ण विधि कार्नॉफ मैप है। यह एक graphical विधि है जिसमें किसी truth table के input व output को एक मैप में दर्शाया जाता है। इस विधि को 2, 3 या 4 variable राशियों के लिये सरलता से प्रयुक्त किया जा सकता है।

K-map for 2-Variable :

→ इसमें कुल 2 column होते हैं।

→ इसमें पहला 0 होता है अर्थात् और जब min term में निकालते हैं तो 0 को $\bar{A}$ व 1 को A लिखते हैं।

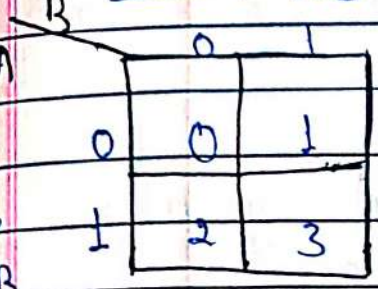
→ A 2 variable K-map consists of $2^2 = 4$ rectangular boxes. Inside these boxes we have to enter the value of output Y for different combinations of input A and B

→ ∵ इसमें चार box होते हैं तो इस में दो को raise करता है अर्थात्
$2^2 = 4$ यानी दो variable (A, B) हो तो 1 से Reduce होता है।

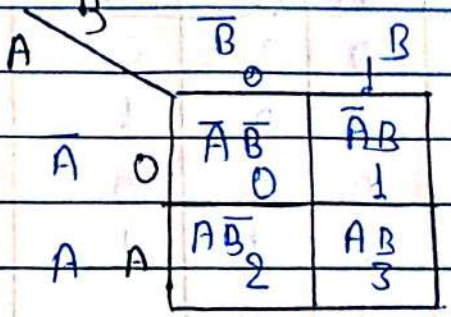
✱ min term का function : 1 में।
✱ max term का function : 0 में।

Page: _____
Date: / /

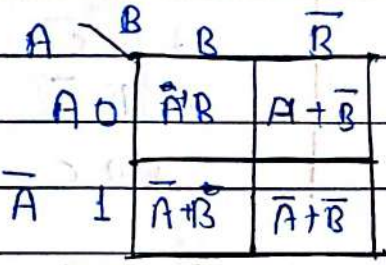
2 Variable k-map



SOP - Min-term



$\bar{A}\bar{B} = 0$
$\bar{A}B = 1$
$A\bar{B} = 2$
$AB = 3$



POS - max term

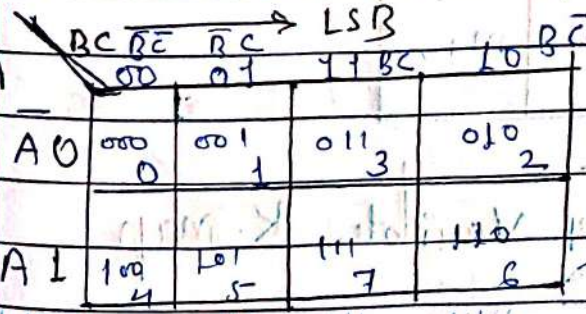
✱ A and B are the inputs or variable
✱ 0 and 1 are the values of A or B
✱ inside 4 boxes we have to enter the value of Y is output

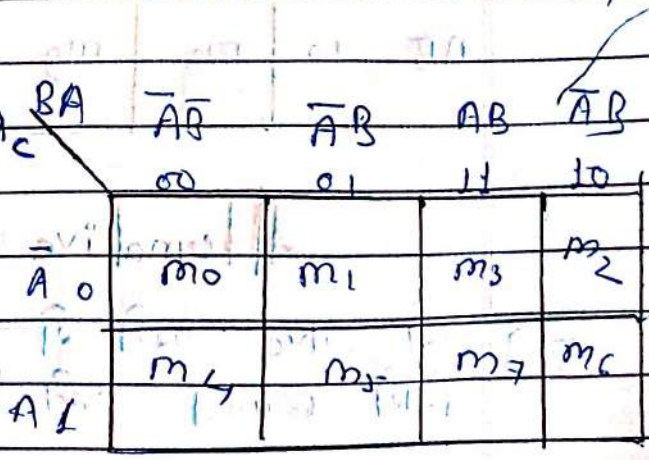
Structure of 2-Variable K-map

3 Variable k-map

→ 3-Variable k-map में किन्हीं दो variable को row में तथा एक variable को colum में रखते हैं।

→ 3 Variable k-map there will be 8 boxes,





SOP - Minterm

Decimal Value	Input- A	B	C	Fundamental product
0	0	0	0	$\bar{A}\bar{B}\bar{C}$
1	0	0	1	$\bar{A}\bar{B}C$
2	0	1	0	$A\bar{B}\bar{C}$
3	0	1	1	$\bar{A}BC$
4	1	0	0	$A\bar{B}\bar{C}$
5	1	0	1	$\bar{A}B\bar{C}$
6	1	1	0	$A\bar{B}\bar{C}$
7	1	1	1	ABC

K-map for 4 variable :

4-Variable map there will be 16 boxes and so, on.

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	m_0	m_1	m_3	m_2
$\bar{A}B$ 01	m_4	m_5	m_7	m_6
AB 11	m_{12}	m_{13}	m_{15}	m_{14}
$A\bar{B}$ 10	m_8	m_9	m_{11}	m_{10}

Alternative four Varible K-map

→ इसे solve करने से पहले इसमें चार ग्रुप बनाते हैं जिसे Quad कहते हैं।

Page: _____
Date: / /

solution के लिए k-map का use

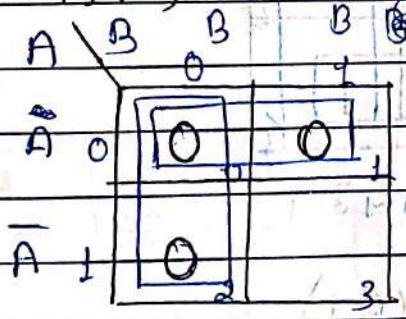
→ SOP → Symbol → Σm (sum min) / min term

→ POS → symbol → ΠM (Product Max) / max term

→ कुछ max term का function - 0
min term का function - 1

min term में 0 means $\bar{A}$ और 1 means A और max term में 0 means A और 1 means $\bar{A}$

Q. $\Pi M(0,1,2)$



∵ Max term से value लिए जाते function 0 होगा

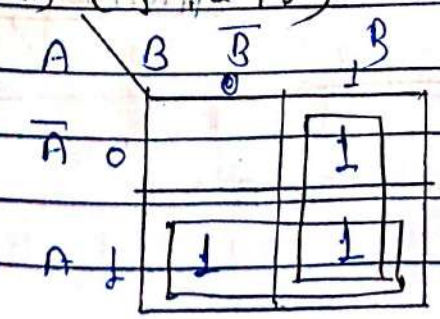
A	B	Y	
0	0	0	max
0	1	0	
1	0	0	
1	1	1	

$M(0,1) = \bar{A}$

$M(2,3) = B$

$= \bar{A}B$ Answer

Q $\Sigma m(1,2,3)$



A	B	Y	
0	0	0	→ max
0	1	1	→ m_1 $\bar{A}B$
1	0	1	→ m_2 $A\bar{B}$
1	1	1	→ m_3 AB

$m(1,3) = B$

$m(2,3) = A$ = ~~AB~~ $A + B$

3- Variable k. map question

$F = \bar{A}\bar{B}C + A\bar{B}C + \bar{A}B\bar{C} + AB\bar{C} + ABC$

$\bar{A}\bar{B}C = 001 = m_1$

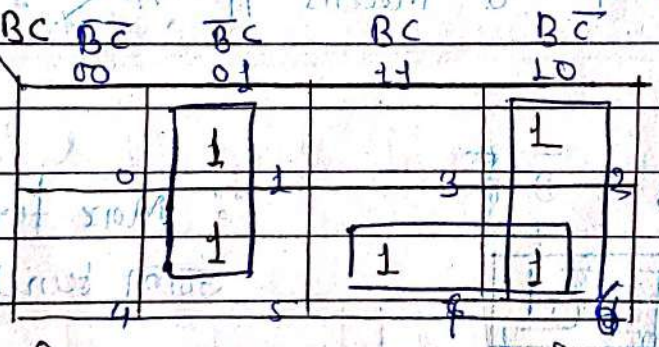
$A\bar{B}C = 101 = m_5$

$\bar{A}B\bar{C} = 010 = m_2$

$AB\bar{C} = 110 = m_6$

$ABC = 111 = m_7$

$\Sigma m(1, 2, 5, 6, 7)$



उपर तीन Pair बनने वाले है।

$m(1,5)$; $m(2,1)$, $m(6,1)$

pair $(m_1, m_5) = \bar{B}5$

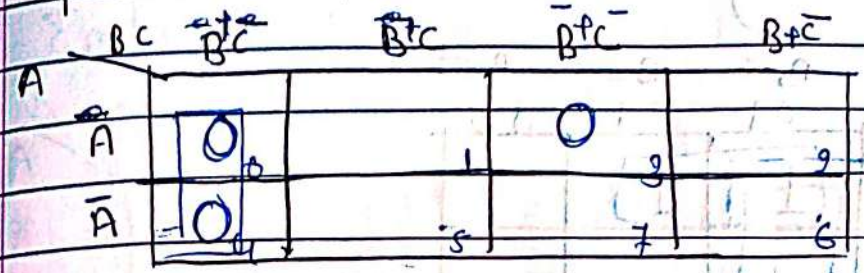
Pair $(m_6\ m_7) = AB$

pair $(m_2, m_6) = B\bar{C}$

reduce form = $F = \bar{B}C + AB + B\bar{C}$ Answer

Page:
Date: / /

$f = \Pi(0,3,4)$

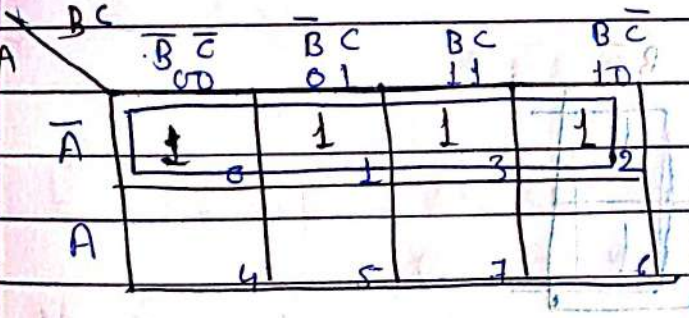


$m(0,4) = \bar{B}\bar{C}$

$m(3) = \bar{A} + B + C$

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

$f = \Sigma m(0,1,2,3)$

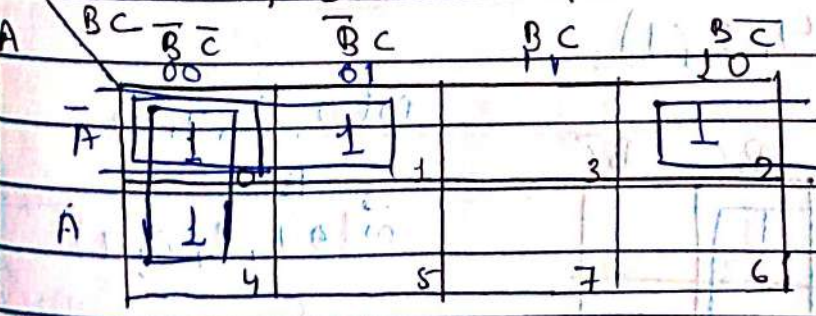


∵ यहाँ 4 Variable का Quad बना जिसमें दो Variable reduce होंगे

$m(0,1,3,2) = \bar{A}$

SOP - Sum of product

(1) $f(A,B,C) = \Sigma m(0,1,2,4)$

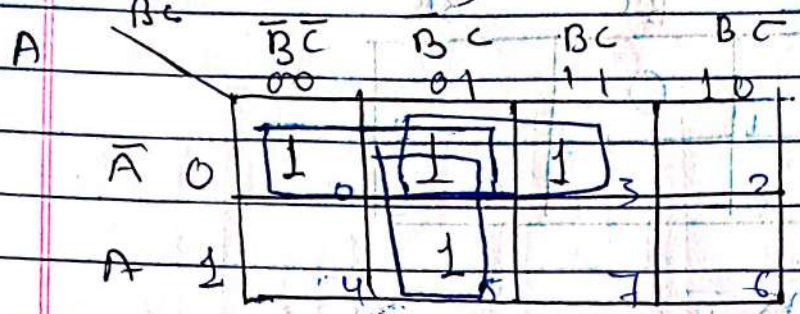


$m(0,4) = \bar{B}\bar{C}$

$m(0,1) = \bar{A}\bar{B}$

$m(2,0) = \bar{A}\bar{C}$

(2) $\Sigma m(0,1,3,5)$



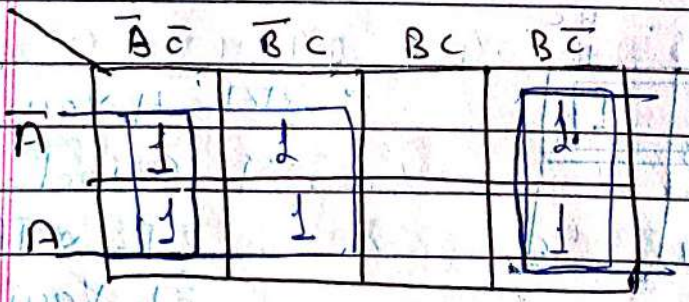
pair $m(0,1) = \bar{A}\bar{B}$

$m(1,5) = \bar{B}C$

$m(1,3) = \bar{A}C$

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

(3) $\Sigma m(0,1,2,4,5,6)$

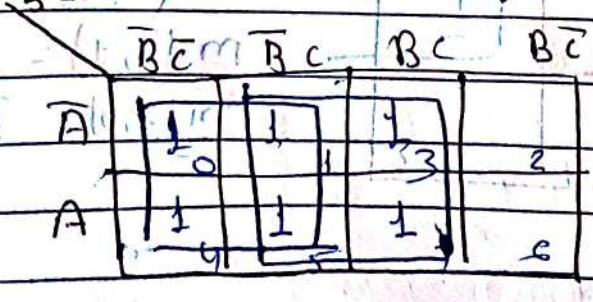


$m(0,1,4,5) = \bar{B}$

$m(0,4,2,6) = \bar{C}$

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

(4) $\Sigma m(0,1,3,4,5,7)$



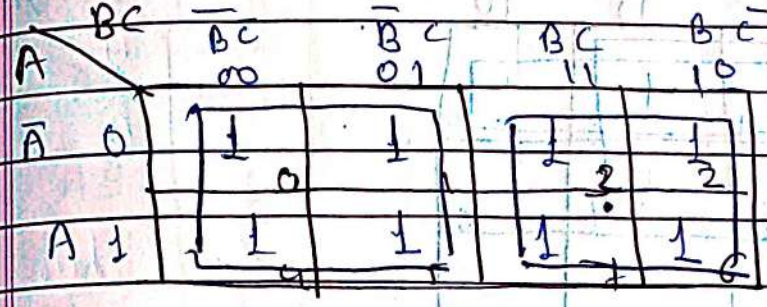
$m(0,1,4,5) = \bar{B}$

$m(0,1,3,5,7) = C$

$= \bar{B} + C$

Page:
Date: / /

(5) $\Sigma m(0, 1, 2, 3, 4, 5, 6, 7)$



जब सभी 1 हो तो Ans भी 1 होगा।

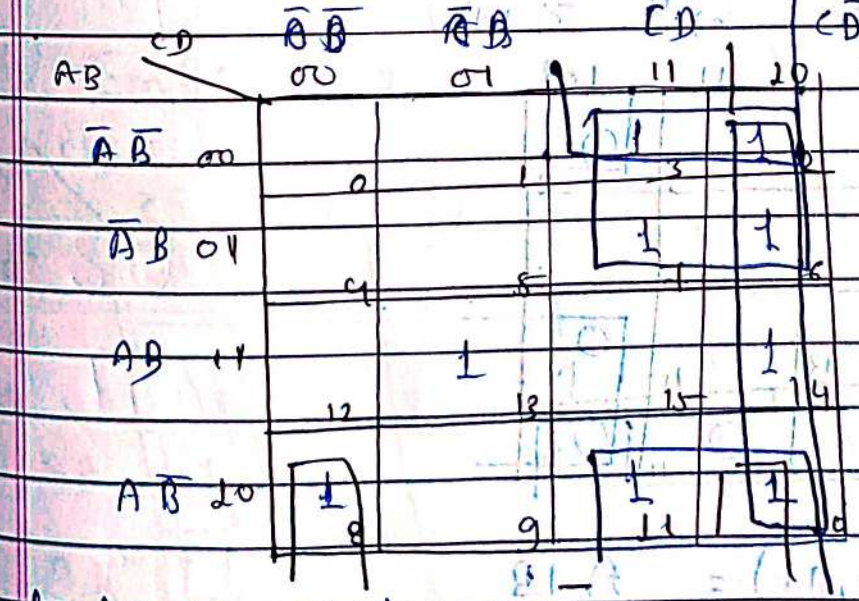
$m(0, 1, 4, 5) = \bar{B}$

$m(3, 2, 7, 6) = B$

$= \bar{B} + B = 1$

4 Variable K-Map

$f(2, 3, 6, 7, 8, 10, 11, 13, 14)$



Pair (8, 10)

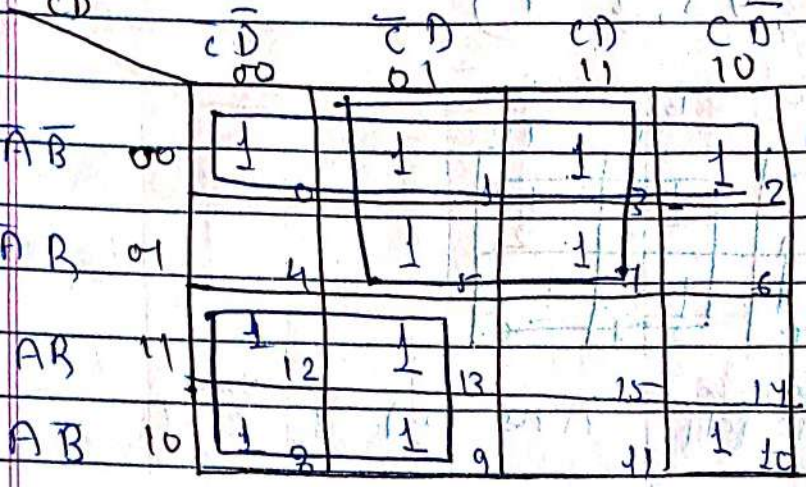
Quad m(2, 6, 14, 10) = $C\bar{D}$ minterm(13) minterm(8)

Quad m(2, 3, 7, 6) = $\bar{A}C$

Quad m(11, 10, 3, 2) = $\bar{B}C$

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

(1) $\Sigma m(0,1,2,3,5,7,8,9,10,12,13)$



$m(0,1,3,2) = \bar{A}\bar{B}$

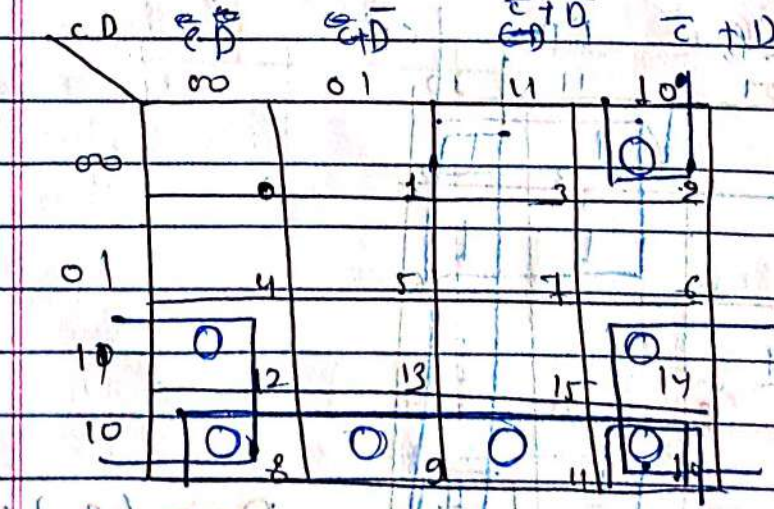
$m(1,3,5,7) = \bar{A}D$

$m(12,13,8,9) = A\bar{C}$

$m(9,10) = B\bar{C}\bar{D}$

Ans $\bar{A}\bar{B} + \bar{A}D + A\bar{C} + B\bar{C}\bar{D}$

(2) $\Pi M(2,8,9,10,11,12,14)$



Quad $m(8,9,10,11) = \bar{A}+B$

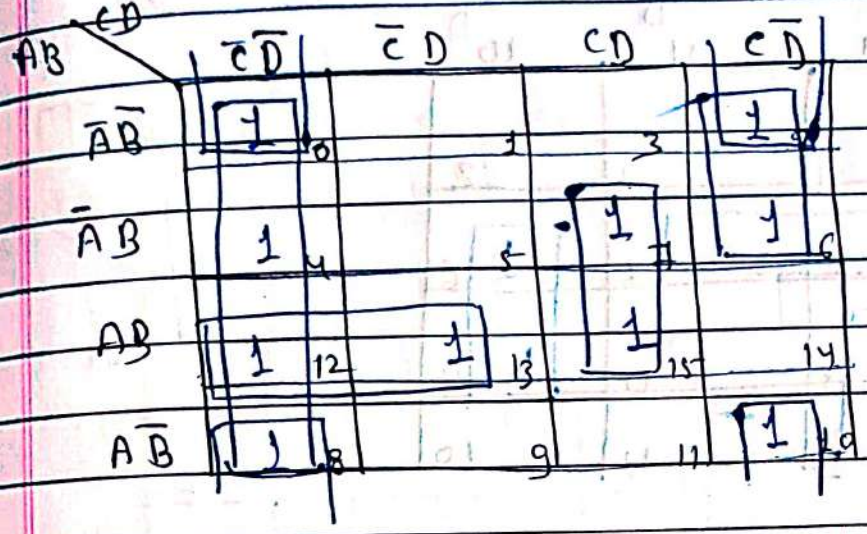
Quad $m(12,8,14,10) = \bar{A}+D$

pair $m(2,10) = B+\bar{C}+D$

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

Page:
Date: / /

③ $\Sigma m\,(0, 2, 4, 6, 7, 8, 10, 12, 13, 15)$



$m(0, 4, 12, 8) = \bar{C}\bar{D}$

$m(0, 4, 2, 6) = \bar{A}\bar{D}$

$m(7, 15) = AB\bar{C}$

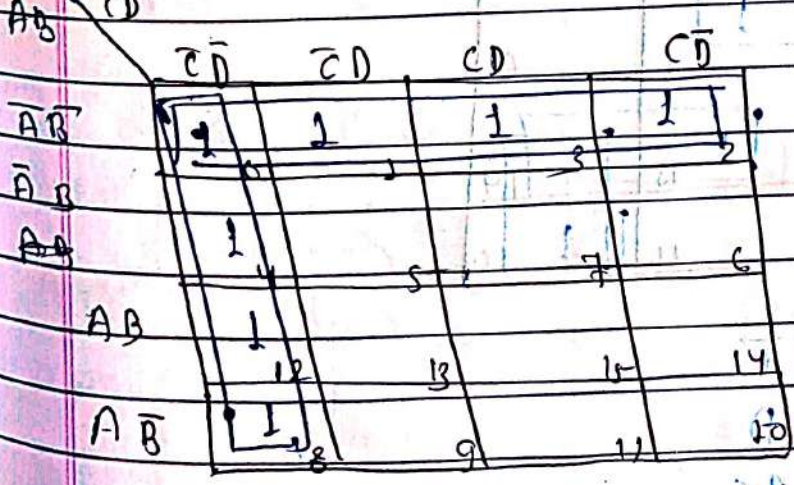
$m(12, 13) = \bar{B}C\bar{D}$

$m(2, 10) = BC\bar{D}$

$m(08, 2, 10) = \bar{B}\bar{D}$

$= \bar{C}\bar{D} + A\bar{D} + AB\bar{C} + \bar{B}C\bar{D} + BCD + \bar{B}\bar{D}$ Ans

④ $\Sigma m(0, 1, 2, 3, 4, 8, 12)$



$m(0, 1, 3, 2)$

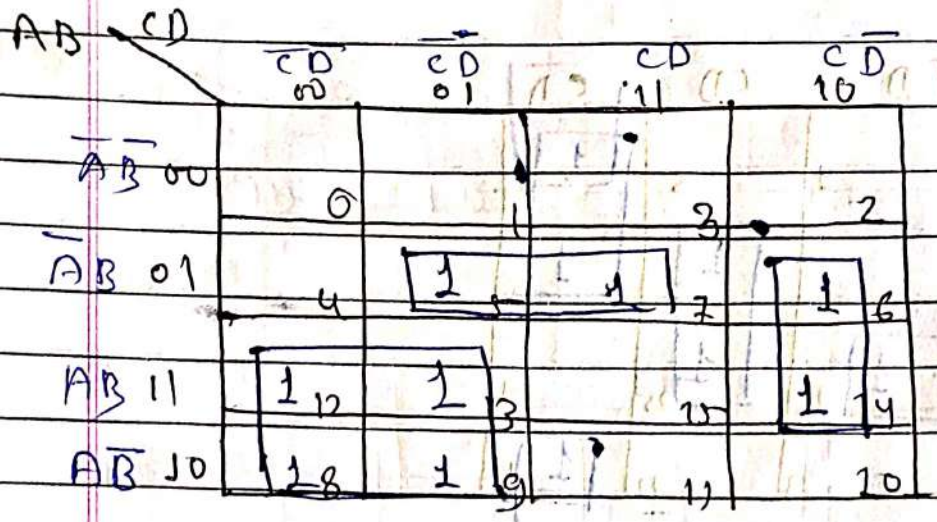
$= \bar{A}\bar{B}$

$m(0, 4, 12, 8)$

$= \bar{C}\bar{D}$

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

(5) $\Sigma m(5, 6, 7, 8, 9, 12, 13, 14)$



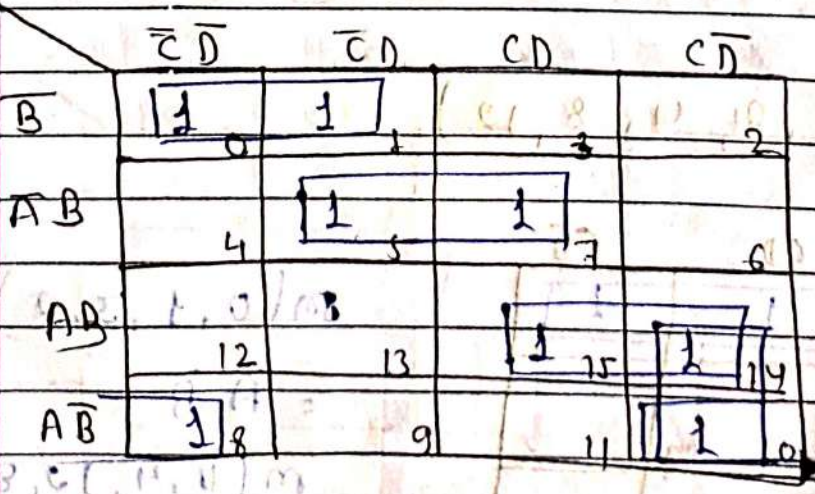
$m(12, 13, 8, 9) = A\overline{C}$

$m(5, 7) = \overline{A}BD$

$m(6, 14) = BC\overline{D}$

$= A\overline{C} + \overline{A}BD + BC\overline{D}$ Ans

(6) $\Sigma m(0, 1, 5, 7, 8, 10, 14, 15)$



$m(0, 1) = \overline{A}\overline{B}\overline{C}$

$m(5, 7) = \overline{A}BD$

$m(15, 14) = ABC$

$m(8, 10) = A\overline{B}\overline{D}$

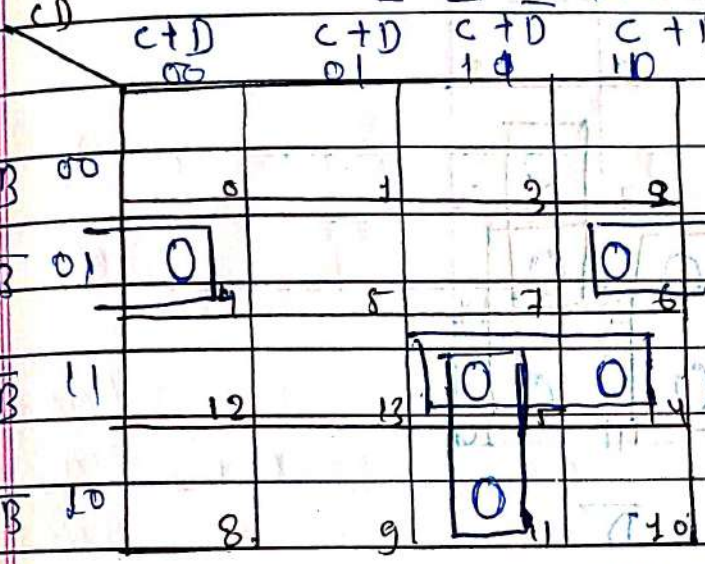
$= \overline{A}\overline{B}C + \overline{A}BD + ABC + A\overline{B}\overline{D}$

POS - MAX term Questions

Page: _____
Date: / /

(1) $\Sigma m(0, 1, 2, 3, 5, 7, 8, 9, 10, 12, 13)$

$\Pi M(4, 6, 15, 14, 11)$



$M(5, 14)$: $\bar{A} + \bar{B} + \bar{C}$

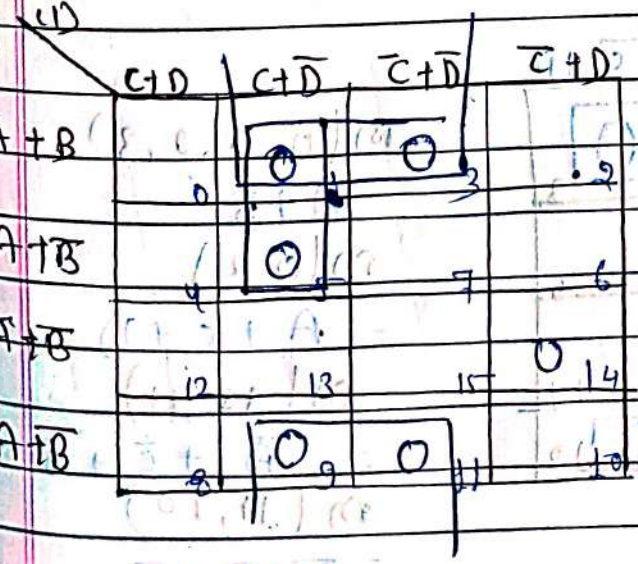
$M(15, 11)$: $\bar{B} + \bar{C} + \bar{D}$

$M(4, 6)$: $A + \bar{B} + D$

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

(2) $\Sigma m(0, 2, 4, 6, 7, 8, 10, 12, 13, 15)$

$\Pi m(1, 3, 5, 9, 11, 14)$



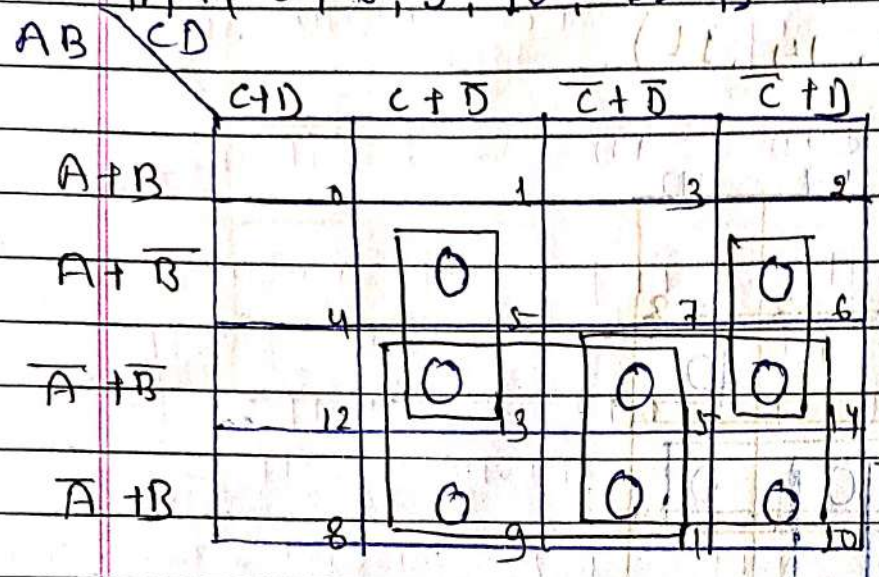
Quad $(9, 11, 1, 3)$: $B + \bar{D}$

Pair $m(1, 5)$: $A + C + \bar{D}$

minterm (14) : $A + \bar{B} + \bar{C} + D$

(4) $\Sigma m(0,1,2,3,4,8,12)$

$\Pi M(5,6,9,10,11,13,14,15)$



Quad $m(13,15,9,11) = \bar{A} + \bar{D}$

Quad $m(15,14,11,10) = \bar{A} + \bar{C}$

pair $m(5,13) = \bar{B} + C + \bar{D}$

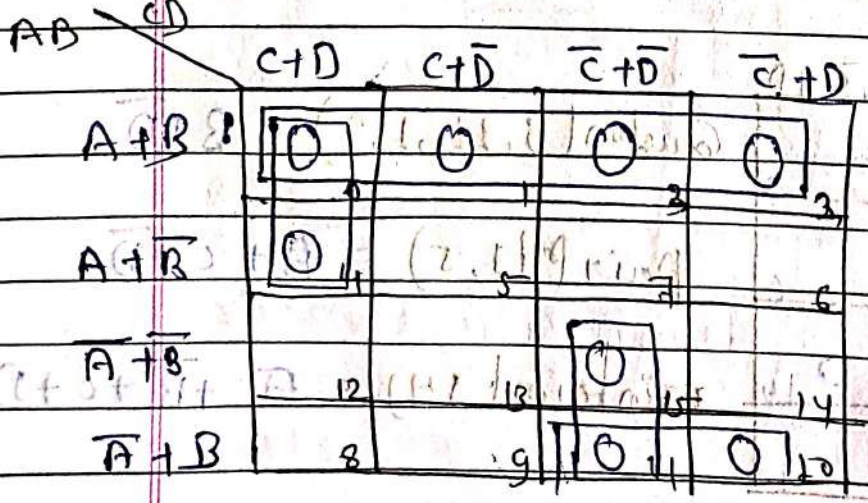
pair $m(6,14) = \bar{B} + \bar{C} + D$

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

(5) $\Sigma m(5,6,7,8,9,12,13,14)$

~~$\Pi m(5,6,9,11,12,13,14,$~~

$\Pi m(0,1,2,3,4,10,11,15)$



$m(0,1,2,3) = A + B$

$m(0,4) = A + C + D$

$m(15,11) = \bar{A} + \bar{C} + \bar{D}$

$m(11,10) = \bar{A} + B + \bar{C}$

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

UNIT - III

Combinational Logic Circuit

सभी डिजिटल यंत्र दो प्रकार के परिपथों से बने होते हैं। ये परिपथ हैं - (1) Combinational logic परिपथ एवं (2) Sequencealy logic circuit।

Combinational :

वे परिपथ होते हैं जिनके output (यदि एक से अधिक output है तो) केवल वर्तमान input की अवस्था पर निर्भर करते हैं। इनमें कुछ जटिल परिपथों

जैसे : मल्टीप्लेक्सर, डिमल्टीप्लेक्सर, इनकोडर, डिकोडर आदि

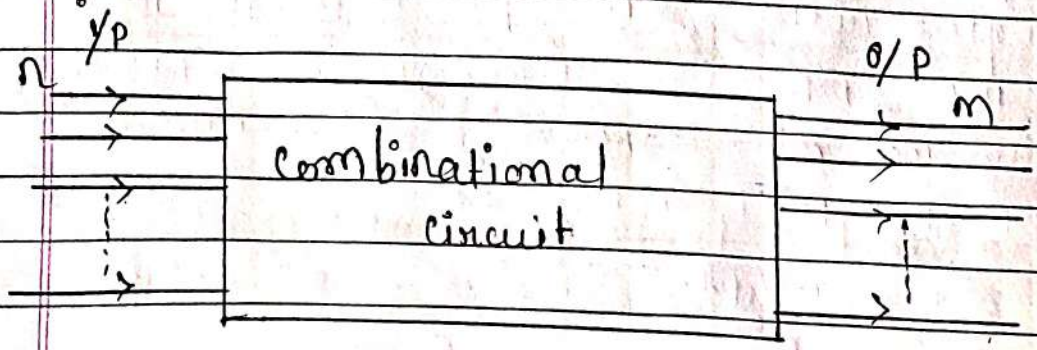
को लॉजिक गेट्स के संयोजनों से बनाया जाता है। अतः लॉजिक गेट्स ही सभी combinational circuits के आधार माने जा सकते हैं।

Combination में memory नहीं होगा।

अर्थात् output store नहीं कर सकते।

* इसमें present के input पर present output मिलता है।

Block diagram of Combinational Circuit



Page:
Date: / /

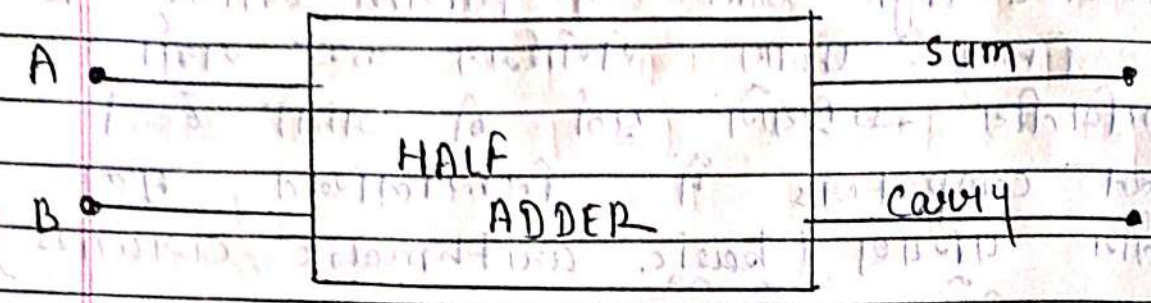
Arithmetic logic unit में विभिन्न लॉजिक गेट्स एवं फ्लिप - फ्लॉप संयोजित कर सभी बीजगणितीय प्रक्रियायें पूर्ण की जाती है। डिजिटल computers में निम्नलिखित मूल गणितीय परियथ (basic arithmetic circuits) प्रयुक्त किये जाते है :

(1) Half adder
(2) Full adder
(3) Full subtractor
(4) Half subtractor
(5) Control inverter

(1) HALF ADDER :

यह एक बिट एडर होता है जो दो बाइनरी संख्याओं का योग करता है। चूँकि दो बाइनरी संख्याओं को जोड़ने पर योग (sum) के साथ carry भी उत्पन्न हो सकता है अतः एक हाफ adder में कभी भी carry add नहीं होता इसलिए इसे HALF adder कहते है।
अतः एक Half adder में दो इनपुट तथा दो output होते है। हमें table से स्पष्ट है कि output में sum, केवल तब उच्च होता है जब दोनों input असमान होते है अतः sum को एक ex-OR गेट द्वारा प्राप्त किया जाता है क्योंकि

ex-OR गेट असमान इनपुट पर उच्च output देता है।



Block Diagram of half adder

Truth table of Half adder

	i/p		sum	o/p	
	A	B		S	C
0	0	0	0+0=	0	0
1	0	1	0+1=	1	0
2	1	0	1+0=1	1	0
3	1	1	1+1=10	0	1

↓
carry

इसी प्रकार table से यह स्पष्ट है कि output में carry केवल तब उच्च (High) होता है जब दोनों input High हों। अतः carry को एक AND gate के output पर प्राप्त किया जा सकता है। और output में sum केवल तब High होता है जब दोनों input समान होते हैं। अतः sum को एक ex-OR गेट द्वारा प्राप्त किया जा सकता है। क्योंकि ex-OR गेट समान इनपुट पर Low output देता है।

Page: _____
Date : / /

A \ B	$\bar{B}$	B
$\bar{A}$	0 (0)	1 (1)
A	1 (2)	0 (3)

minterm = 1

SOP

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

A \ B	$\bar{B}$	B
$\bar{A}$	0 (0)	0 (1)
A	0 (2)	1 (3)

minterm = 1

SOP

Carry = $A \cdot B$

या ∵ A = 0 व B = 1 पर output 1 है

∴ 0 = $\bar{A}$, 1 = B output = $\bar{A}B$

→ 1 = $\bar{A}$ = 0 , = $\bar{B}$ = output = $A\bar{B}$

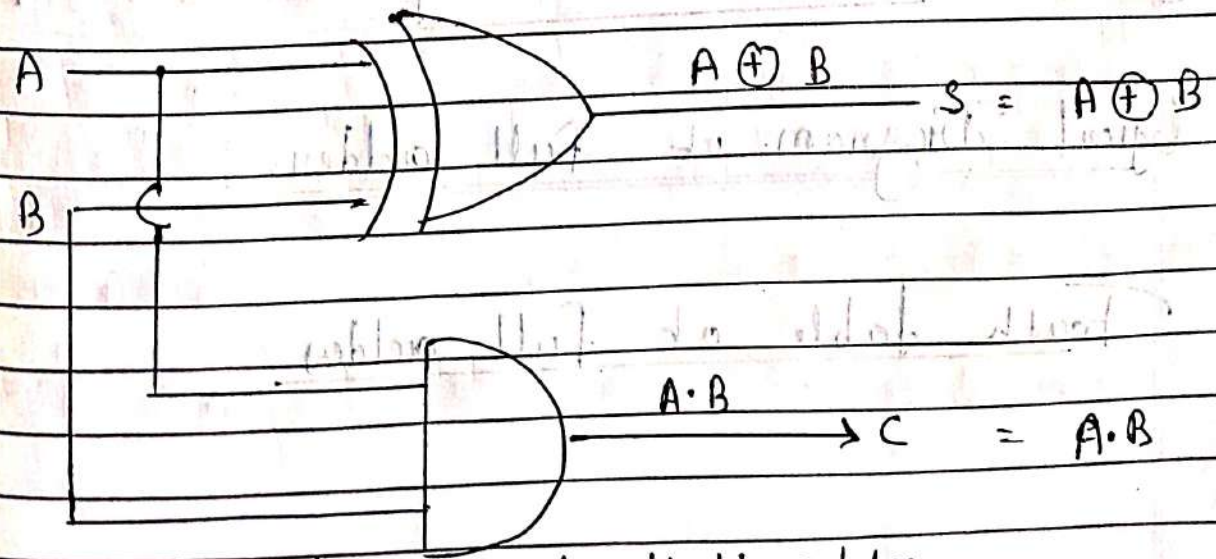
SOP = $\bar{A}B + A\bar{B}$

K-map से SUM और Carry के logic circuit नि० लि० पता है हैं

SUM = $\bar{A}B + A\bar{B}$ = $A \oplus B$ Ex-OR Gate

Carry = $A \cdot B$ = And Gate

Logic diagram (implement) :-



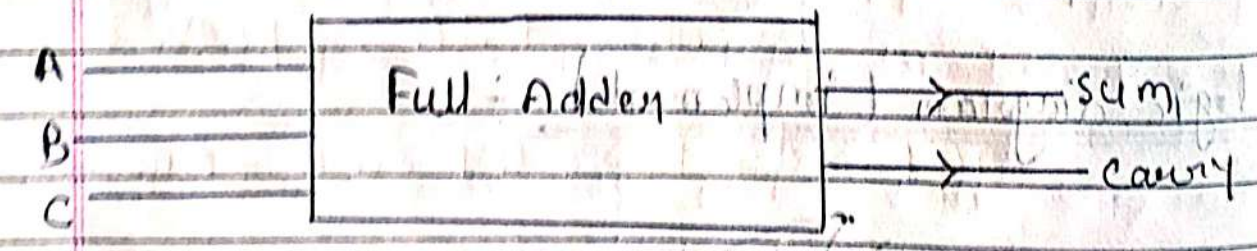
Logic diagram of Half adder

इस परिपथ को Half adder इसलिये कहा जाता है क्योंकि यह पिछले योग के कैरी को स्वीकार नहीं करता है इस कार्य के लिये 3 input adder का उपयोग करना होता है।

Half adder is a combination ckt it has two input and two output (sum, carry) it adds two bits at a time.

(3) FULL Adder :

फुल Adder द्वारा उक्त समय में 3 बिट्स का योग किया जा सकता है जिसमें तीसरा बिट पिछले योग से प्राप्त कैरी हो सकता है। इस प्रकार का फुल adder में तीन input तथा दो output होते हैं यह carry भी add करता है



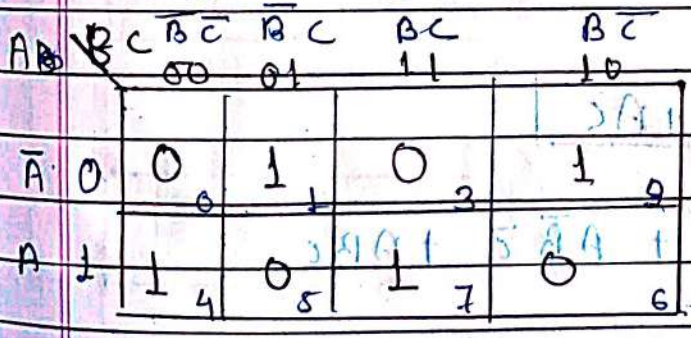
Logical Diagram of Full adder

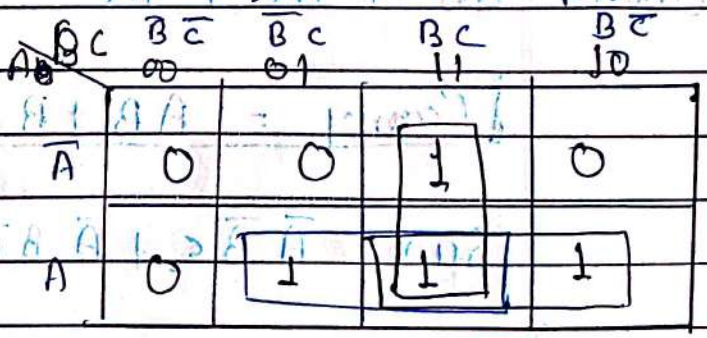
Truth table of full adder

Page:
Date: / /

I/P			O/P	
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

table से स्पष्ट है कि output में Carry तब उच्च (1) होता है जब input में A, B व C में से दो या दो से अधिक इनपुट 1 (High) होते है। उसी प्रकार output में sum का तब तब High (1) होता है जब input's में 1's की संख्या विषम होती है यह अवस्था XOR गेट के logic के तुल्य होती है। sum तथा carry के K-map





SUM

Carry

$= A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$ $= \cancel{\bar{A}BC} + \cancel{\bar{A}BC}$

$+ ABC$ $\cancel{+ ABC} + \cancel{A\bar{B}C}$

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

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

Page:
Date: / /

SUM = $\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

Carry = $\bar{A}BC + AB\bar{C} + AC\bar{B} + ABC$

Logical expression

~~Sum~~ Carry = $\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

$= \bar{A}BC + A\bar{B}C + AB(C+\bar{C})$ $\quad [\because C+\bar{C}=1]$

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

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

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

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

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

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

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

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

Carry = $AB + BC + AC$

Carry = $AB + BC + AC$

Sum = $\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$= \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C} + A\bar{B}\bar{C}$ Ex-NOR

$= C(\bar{A}\bar{B} + AB) + \bar{C}(\bar{A}B + A\bar{B})$ $\quad [\because \bar{A}\bar{B} + AB = \overline{A \oplus B}]$

$= C(\overline{A \oplus B}) + \bar{C}(A \oplus B)$ $\longrightarrow [\bar{A}B + A\bar{B} = A \oplus B]$

$= (A \oplus B) \oplus C$ X-OR

Sum = $(A \oplus B) \oplus C$

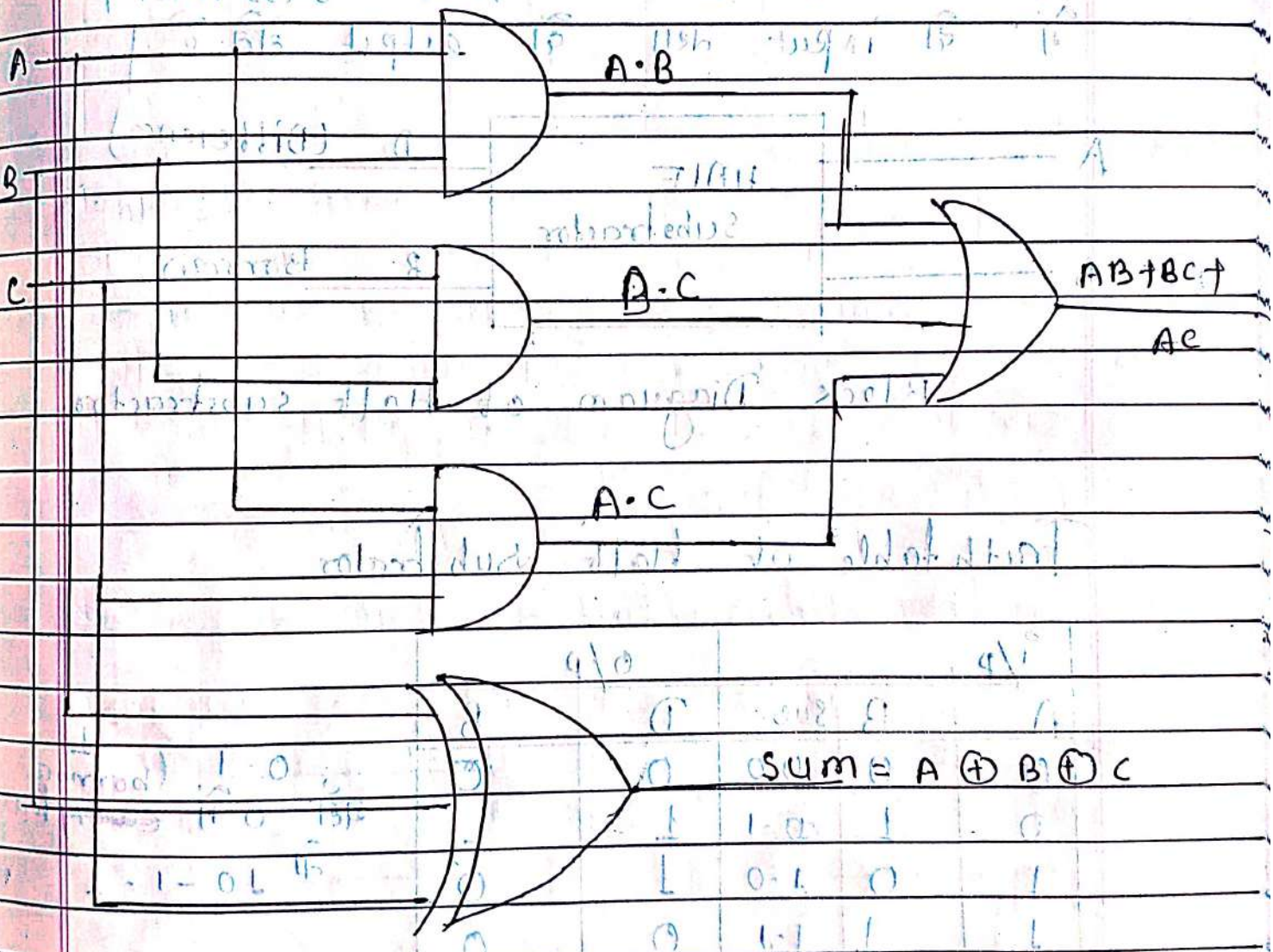
Page:
Date: / /

Sum व Carry के संशोधित समीकरण द्वारा Full adder के लॉजिक circuit

Logical diagram (implement)

Carry = $AB + BC + AC$

Sum = $(A \oplus B) \oplus C$

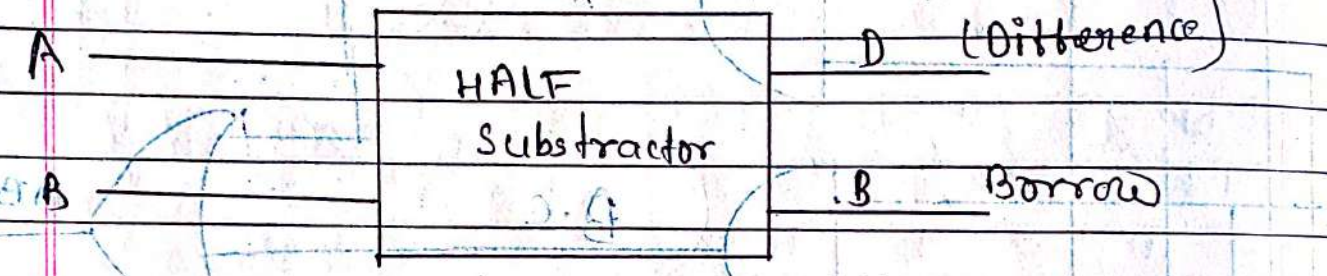


Logical diagram of full adder

Full adder को दो half adder तथा एक OR गेट से बनाया। इस विधि में प्रथम हाफ एडर के sum को द्वितीय हाफ adder में दिया। द्वितीय हाफ एडर द्वारा प्राप्त sum अंतिम sum को प्रदर्शित करता है दोनों adder के carry को एक OR gate में देकर अंतिम carry प्राप्त किया जा सकता है

HALF Subtractor

Half subtractor द्वारा एक समय में दो बाइनरी अंकों को घटाया जा सकता है। इसे घटाने की क्रिया में output में अंतर (difference) के साथ - साथ (borrow) भी उत्पन्न हो सकता है। अतः स्पष्ट है कि एक HALF subtractor में दो input तथा दो output होते है।



Block Diagram of Half subtractor

Truthtable of Half Subtractor

I/P			O/P	
A	B	Sub.	D	B
0	0	0-0	0	0
0	1	0-1	1	1
1	0	1-0	1	0
1	1	1-1	0	0

0 - 1 = 1 (borrow)
यहाँ 0 में ~~carry~~ लिए
तो 10 - 1 = 1

तालिका से

(i) जब A = 0 तथा B = 0 तो अंतर A - B = 0 तथा borrow = 0 होगा।

(ii) जब $x = 0$ तथा $y = 1$ है तब घटाने की क्रिया हेतु अगले उच्च बिट से एक borrow लेना होगा। चूँकि अगले उच्च बिट का मान 2 होता है अतः अंतर = 2 - 1 = 1 तथा उधार लेने के कारण Borrow = 1 होगा।

(iii) जब $x = 1$ तथा $y = 0$ है तब अंतर 1 - 0 = 1 तथा उधार 0 होगा।

(iv) जब $x = 1$ तथा $y = 1$ है तब अंतर 1 - 1 = 0 तथा उधार 0 होगा।

∵ ~~किन~~ difference में 1 (0,1) यानी $\bar{A}B$ व 1 (1,0) यानी $A\bar{B}$ से आया।

∴ $\bar{A}B + A\bar{B}$

व Borrow में 1 (0,1) यानी $\bar{A}B$ होगा।

Using K-map of Half subtraction

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

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

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

$= \bar{A}B$

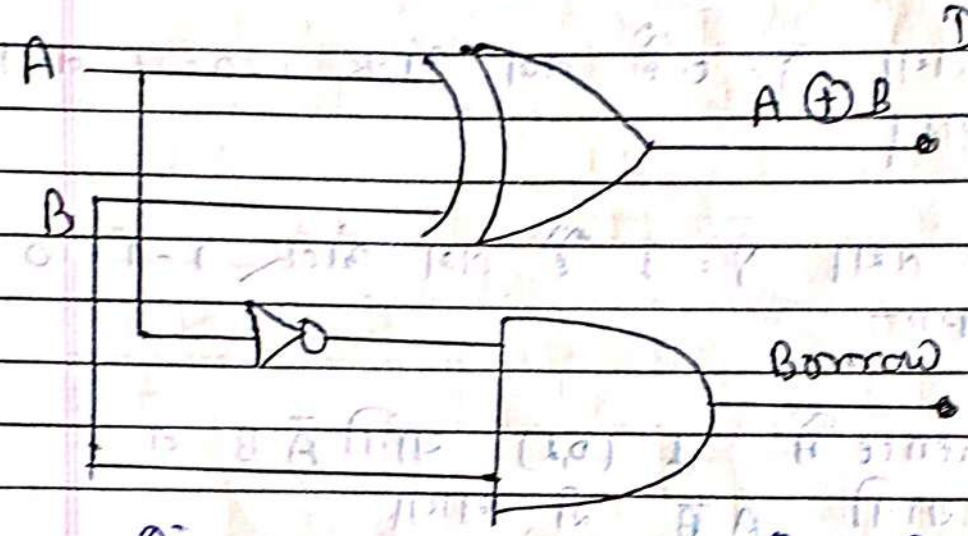
Difference D = $\bar{A}B + A\bar{B}$

Borrow B = $\bar{A}B$

Logical circuit of half subtractor

$D = A\bar{B} + \bar{A}B \approx A \oplus B$ ex-OR

$C = \bar{A}B$

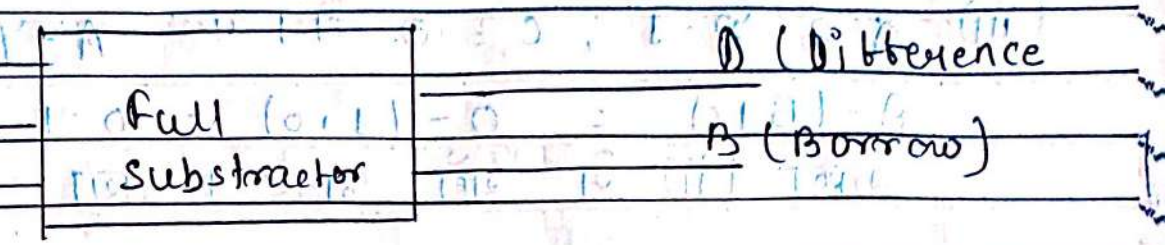


इसमें एक ex-OR gate है और एक AND gate है।

FULL Subtractor

फुल subtractor द्वारा तीन बाइनरी डिजिट में घटाने की क्रिया की जा सकती है। घटाने की क्रिया में output में पूर्व की भाँति difference तथा BOR (borrow) उत्पन्न होते हैं।

इस प्रकार एक full subtractor में तीन input तथा दो output होते हैं। इन तीन input में से data input तथा एक पिछले घटाने की क्रिया से उत्पन्न borrow हो सकता है।



Block diagram of full ~~adder~~ subtractor

Truth table of full subtractor

i/p			o/p	
A	B	C	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

तीन Bit के घटाव की क्रिया निम्नलिखित है

(i) $A = B = C_0 = 0$ होने पर $A - B - C = 0 - 0 - 0 = 0$ अतः अंतर D तथा उधार B दोनों 0 प्राप्त होंगे।

(ii) $A = 0$, $B = 0$, $C = 1$ होने पर $A - B - C = A - (B + C)$

$B + C = 0 + 1 = 1$ और $A - B + C = 0 - 1$ होने के कारण अगले उच्च स्थिति में 1 उधार लेना होगा और अगले उच्च स्थान का मान 2 होता है अतः अंतिम $D = (2 + 1) = 1$ तथा उधार लेने के कारण $B = 1$ है

(iii) $A=0$, $B=1$, $C=0$ होने पर $A-B-C$ मतलब $A-(B+C) = 0-(1+0) = 0-1$ होने के कारण (ii) से यहाँ अंतर 1 तथा $B=1$ होगा।

(iv) $A=0$, $B=1$, $C=1$ होने पर $A-B-C$ $= 0-1-1$ होने के कारण अगले उच्च क्रिट से उधार लेंगे। चूँकि अगले उच्च क्रिट का मान 2 होता है अतः अंतर $D = 2-1-1 = 0$ तथा उधार लेने के कारण $B=1$ होगा।

(v) $A=1$, $B=0$, $C=0$ होने पर $A-B-C = 1-0-0$ होने के कारण अंतर $D=1$ तथा borrow 0 होगा।

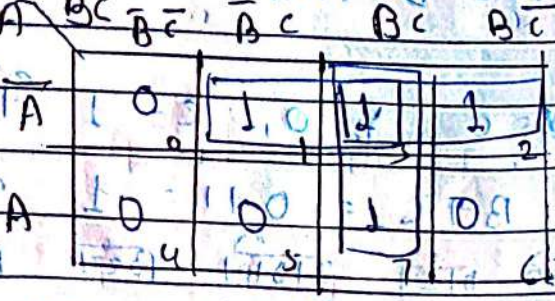
(vi) $A=1$, $B=0$, $C=1$ होने पर $A-B-C = 1-0-1$ होने पर अंतर $D=0$ तथा उधार $=0$ होगा।

(vii) $A=1$, $B=1$, $C=0$ होने पर $A-B-C = 1-1-0$ $D=0$ तथा Borrow 0 होगा।

(viii) $A=1$, $B=1$, $C=1$ होने पर $A-B-C = 1-1-1$ $D=0$ तथा Borrow 0 होगा।

K-map of full subtractor

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



$m(3,7) = BC$

$m(1,3) = \bar{A}C$

$m(3,2) = \bar{A}B$

Page:
Date: / /

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

$m(2) = \bar{A}B\bar{C}$

$m(4) = A\bar{B}\bar{C}$

$m(7) = ABC$

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

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

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

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

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

$= \bar{A} \oplus (B \oplus C) \Rightarrow A \oplus B \oplus C$

$m(3,7) = BC$

$m(1,3) = \bar{A}C$

$m(3,2) = \bar{A}B$

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

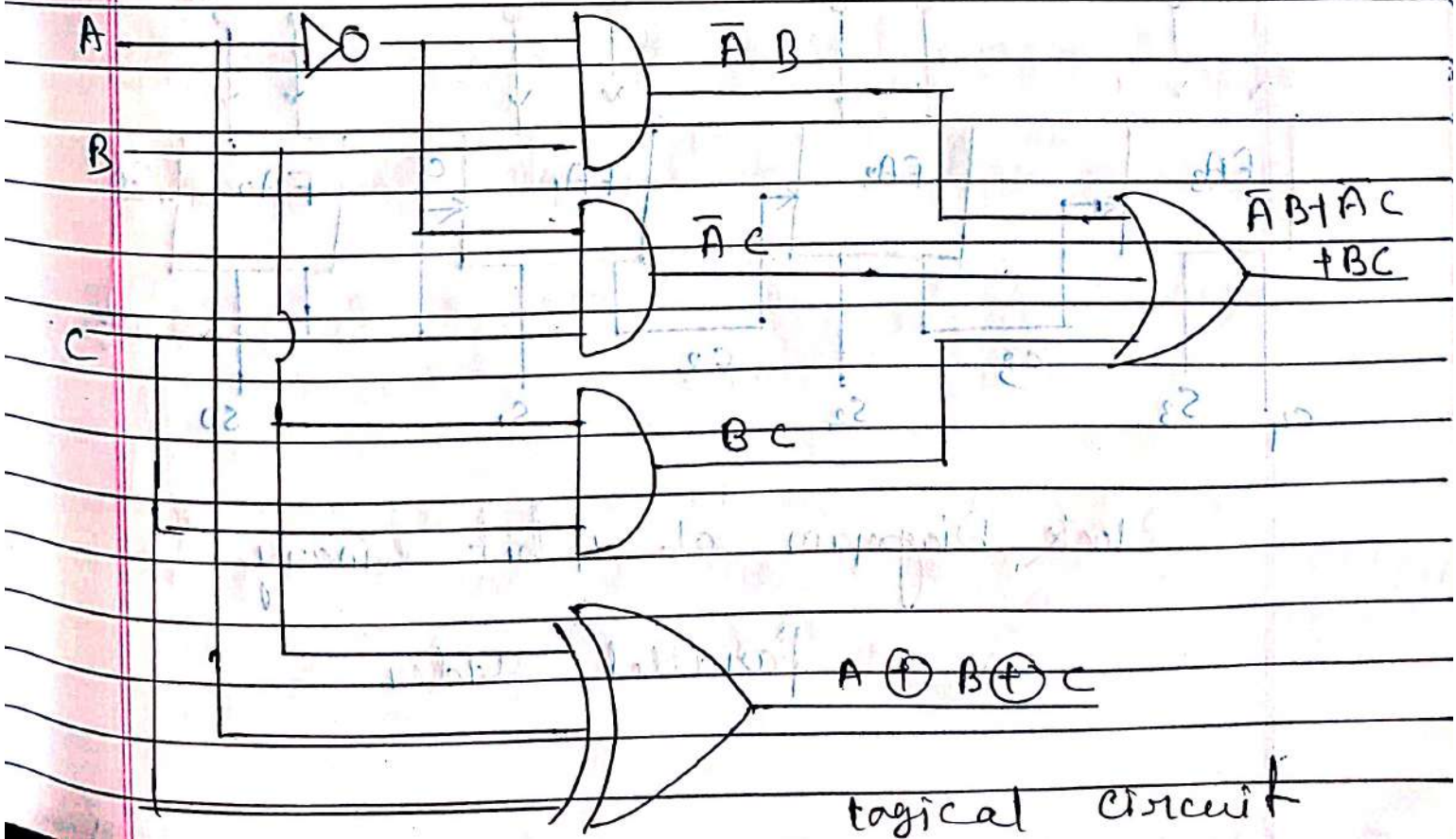
$[A\bar{B} + \bar{A}B = A \oplus B$

$[\because \bar{B}\bar{C} + BC = \overline{B \oplus C}$

$[\because A\bar{B} + \bar{A}B = A \oplus B$

Difference $= A \oplus B \oplus C$

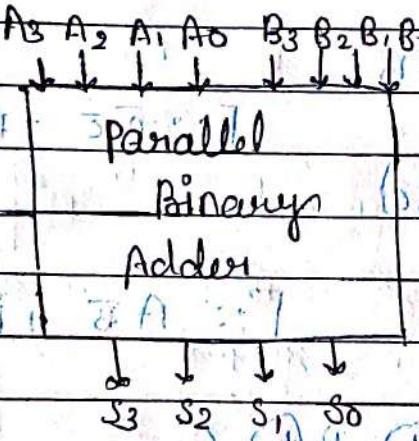
Borrow $= BC + \bar{A}C + \bar{A}B$

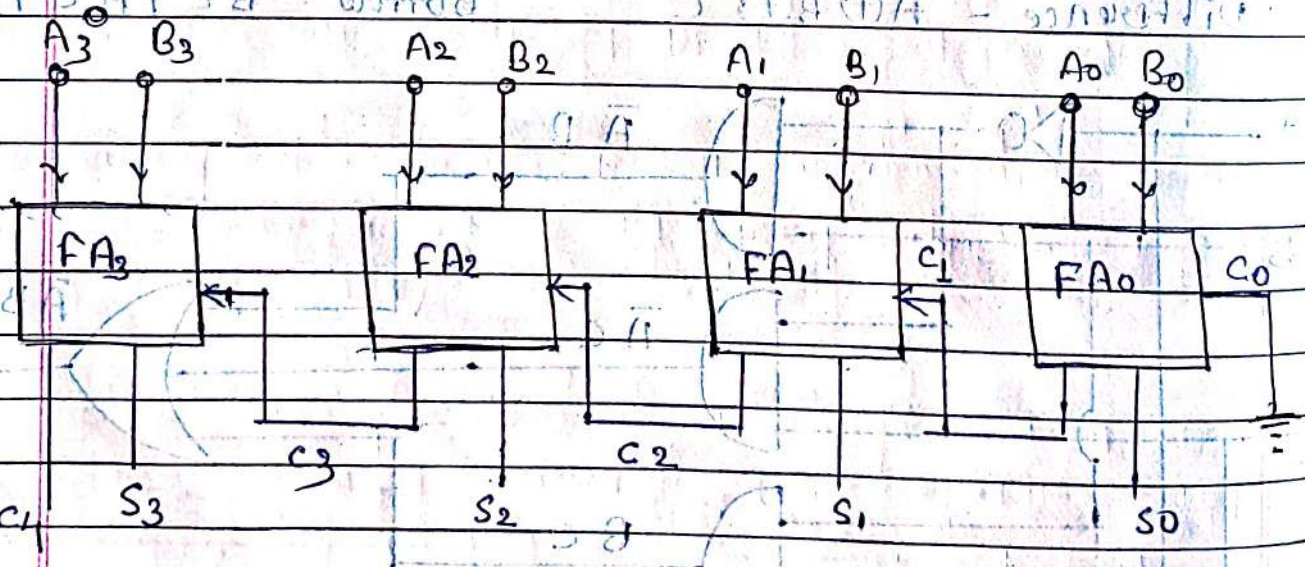


logical circuit

4 Bit Binary Parallel Adder

~~यह~~ इसमें यह दो बाइनरी संख्याओं A_3, A_2, A_1, A_0 तथा B_3, B_2, B_1, B_0 का योग कर output में C_4, S_3, S_2, S_1, S_0 उत्पन्न करता है जिसमें C_4 अंतिम carry होता है। इस adder में सबसे दाया adder (HA) तथा अन्य सभी इसके full adder (FA) होते है।





Block Diagram of 4 bit binary Parallel adder

A 4-bit binary parallel Adder it is a digital combinational circuit which adel 2- binary number each of 4 number in parallel form and produces.

Arithmetic Submition in parallel form it constance 4 full adelen Connected in a chain.

preciding stage full adder ~~connected parallel in a chain~~ carryout (c-out) acts as a input carry you next stage.

1) A_0 represent LSB bit at first binary number A.

2) B_0 represent LSB bit at Second binary number B.

3) A_3 represent MSB bit at first number A.

4) B_3 represent MSB bit at second number B.

5) Input carry for this parallel adder is C_0 and the output carry is C_4.

इसमें सबसे पहले दो input A_0, B_0 के लिए output Sum S_0 है और C_1 प्राप्त होता है और C_0 को generated करते है

अगला Full adder जिसमें तीन इनपुट A_1, B_1 व C_1 होते इसके द्वारा output में योग S_1 तथा Carry C_2 प्राप्त होते है। इस प्रकार सभी Full adder द्वारा अंतिम output C_4, S_3, S_2, S_1, S_0 प्राप्त होता है जिसमें C_4 अंतिम Carry होता है। शब्द की संख्या बढ़ाकर अधिक bit की संख्याओं का योग किया जा सकता है।

बाइनरी adder को सिंपल adder भी कहा जाता है।

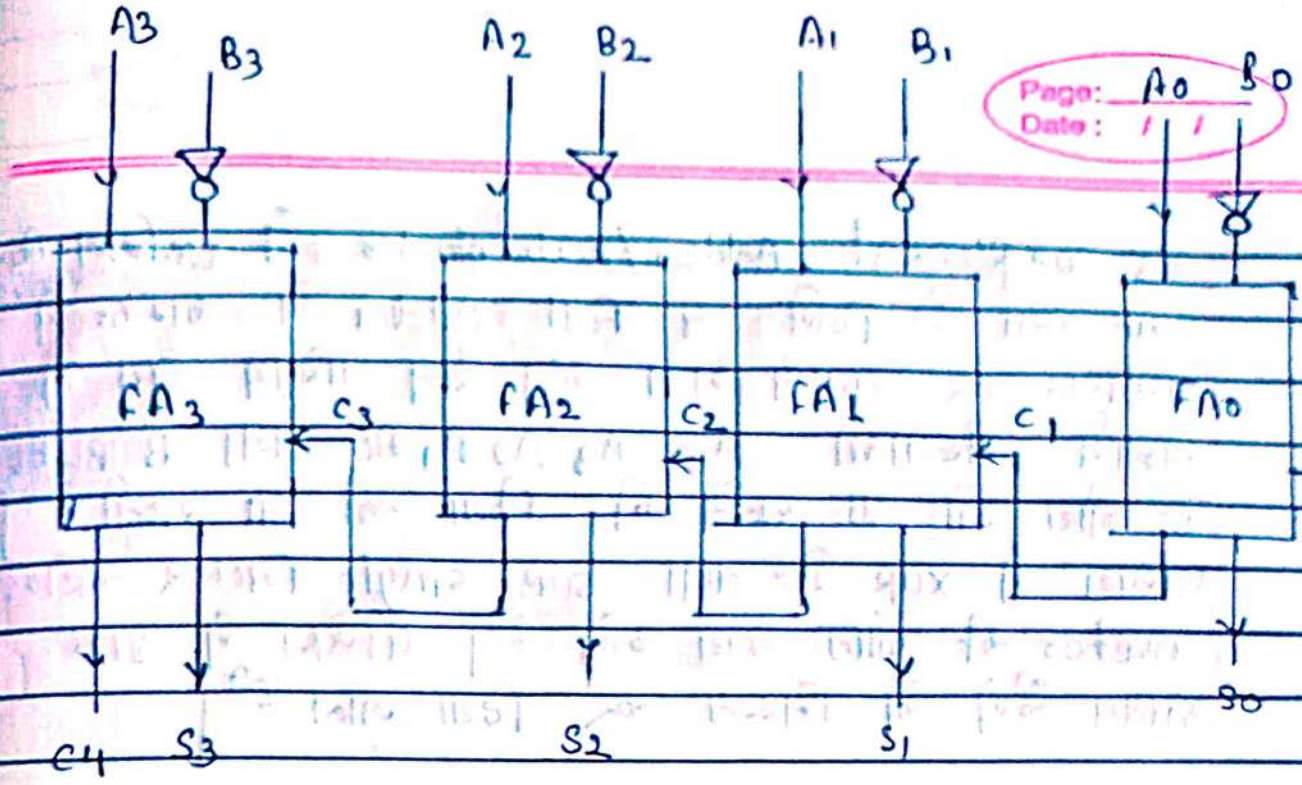
4 Bit Parallel Binary Subtraction

यह दो बाइनरी संख्याओं A_3, A_2, A_1, A_0 तथा B_3, B_2, B_1, B_0 को घटाकर output में C_4, S_3, S_2, S_1, S_0 प्राप्त होता है।

हमें यहां 4bit Parallel subtractor के लिए इससे Circuit में use करना पड़ेगा इस समस्या के हल के लिए अर्थात्

/* Binary subtractor के लिए हम compliment का use करते है अर्थात् compliment के द्वारा binary subtractor को easy किया जाता है। */

इसलिए Binary subtractor में compliment का use किया जाता है इसके लिए जिस number पर (-) चिन्ह होती है उसका compliment निकालते है फिर add करते है। subtract करने के लिए borrow लेते है। लेकिन compliment के द्वारा इसे हम Circuit में जोड़ा जाता है जिससे carry प्राप्त होती है।



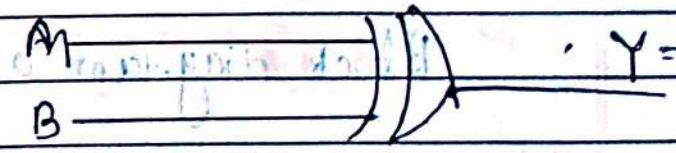
Block Diagram of Parallel Subtractor

First input is given direct output S0 and C1 इस प्रकार सभी input जिसमे A और B का complement लेते है जिसका output S0 और carry प्राप्त होता है।

~~2's compliment Adder subtractor~~

Binary adder - subtractor

$Y = M\bar{B} + \bar{M}B$



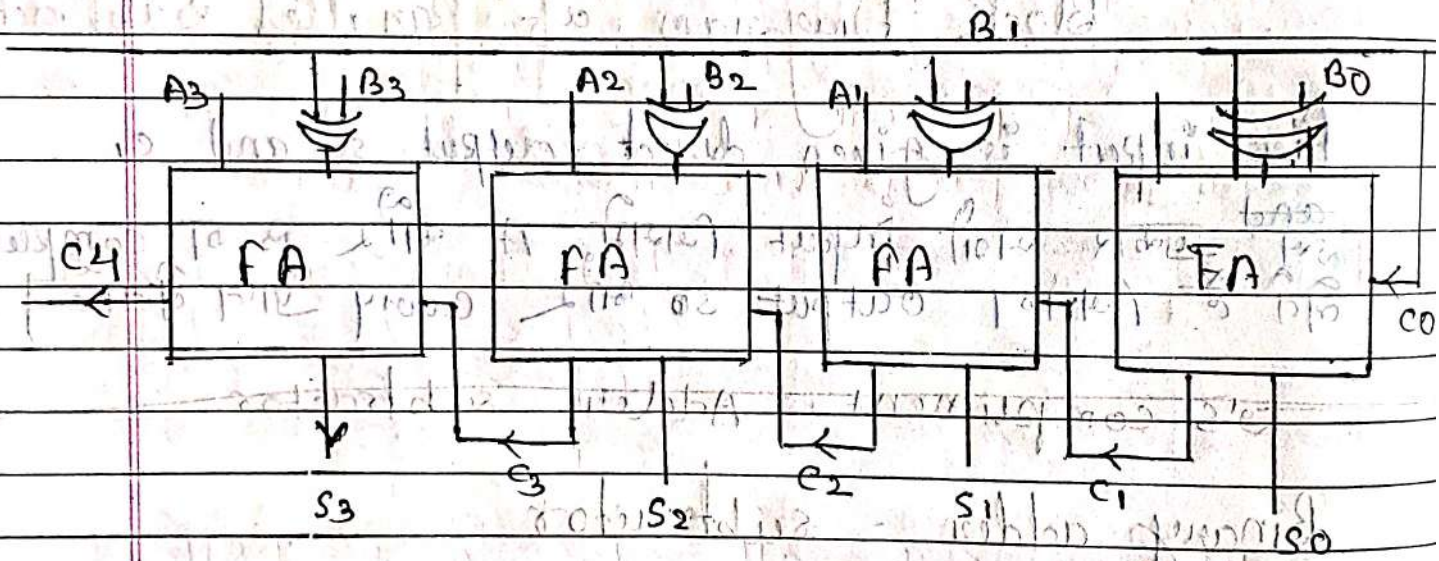
$M=0 \quad 0 \cdot \bar{B} + \bar{0} B \overset{0}{=} 1 \cdot B = B$

$M=1 \quad 1 \cdot \bar{B} + 1 \cdot B = \bar{B}$

हम जानते है कि किसी संख्या का 2's compliment उस संख्या के ऋणात्मक मान को दर्शाता है। साथ ही यह भी ज्ञात है कि 2's complement प्रदर्शन द्वारा गणितीय परिपथों की जटिलता को कम किया जा सकता है। अतः परिपथों में घटाने की क्रिया हेतु संख्या के 2's complement का उपयोग किया जाता है।

Page:
Date: / /

2's complement Adder / subtractor को प्रदर्शित किया गया है जिसे 4 Full Adder को कास्केड में संयोजित कर बनाया गया है। उस परिपथ द्वारा दो बाइनरी संख्याओं $A = A_3 A_2 A_1 A_0$ तथा $B = B_3 B_2 B_1 B_0$ के बीच जोड़ या घटाने की क्रिया की जा सकती है। परिपथ में XOR गेट तथा SUB single मिलकर कंट्रोल्ड inverter की भांति कार्य करते हैं। परिपथ से प्राप्त अंतिम कैरी को निरस्त कर दिया जाता है।



Block diagram of Binary Adder-Subtractor

First number input is given through

Second input is given through ex-OR Gate which has two input first input is number B if self & second input is m

Page:
Date: / /

Where as m denotes mode of operation when even m = 0 Addition mode will be perform and m = 1 subtraction will be perform

B = m = 0 Adder

B = m = 1 Substracter

BCD Adder

BCD Adder is a 4 bit adder. (if we are adding two BCD number A & B each have 4 bit)

बाइनरी coded Decimal adder या BCD Adder द्वारा दो BCD संख्याओं का योग कर output को भी BCD रूप में प्राप्त किया जाता है। BCD कोड जो एक 8421 code है मे 4 bit की 10 संभव अवस्थाये अर्थात valid number 0000 से 1001 तक (डेसीमल मान 0 से 9 तक) होती है तथा 6 अवस्थाये 1010 से 1111 तक (डेसीमाल मान 10 से 15 तक) invalid होता है। दो BCD संख्याओं के योग में विसंगति उस समय आती है जब योग का मान 1001 अर्थात डेसीमल 9 से अधिक होता है।

For example :

Decimal 9 तथा 3 का योग करने पर 12 जिसका बाइनरी मान 1100 है जबकि BCD में इसका मान 00010010 प्राप्त होना चाहिए।

इस प्रकार योग का मान 9 या 9 से कम होने पर बाइनरी योग तथा BCD योग समान होते हैं परन्तु योग का मान 9 से अधिक होने पर सही BCD मान प्राप्त करने के लिये BCD के 6 अमान्य code को skip करना होता है।

→ अर्थात बाइनरी योग में 0110 (decimal 6) अतिरिक्त जोड़ देने पर सही BCD योग प्राप्त होता है।

इस प्रकार 1001 तथा 0011 के योग की प्रक्रिया में प्राप्त बाइनरी योग 1100 में अतिरिक्त 0110 (Decimal 6) को पुनः योग करने पर 00010010 प्राप्त होता है जो decimal 12 का सही BCD मान है।

example → 9 → 1001
+ 3 → 0011
12 1100

लेकिन 1 → 0001 और 2 → 0010 होता है
तो BCD value 12 का 00010010 होना चाहिए
इसलिए 1100 + 0110
+6
00010010 प्राप्त हुआ

9 → 1001 1001 0
+9 → 1001 +0110
18 10010 0000 1100

Page:
Date: / /

अब Subtraction का result 0-9 तक आये तो हम 4-bit Subtraction से increament कर सकते है।

but there can be invalid code also valid are (1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, 10010)

The circuit for BCD adder must include logic needed to detect weather the sum is greater than 9 (8 | 001)

$C_4\ S_3\ S_2\ S_1\ S_0$

0 1 0 0 1

so that correction can be added

→

Case :- I

1) When ever S_4 is 1 ($S_4 = 1$) (sum > 15)

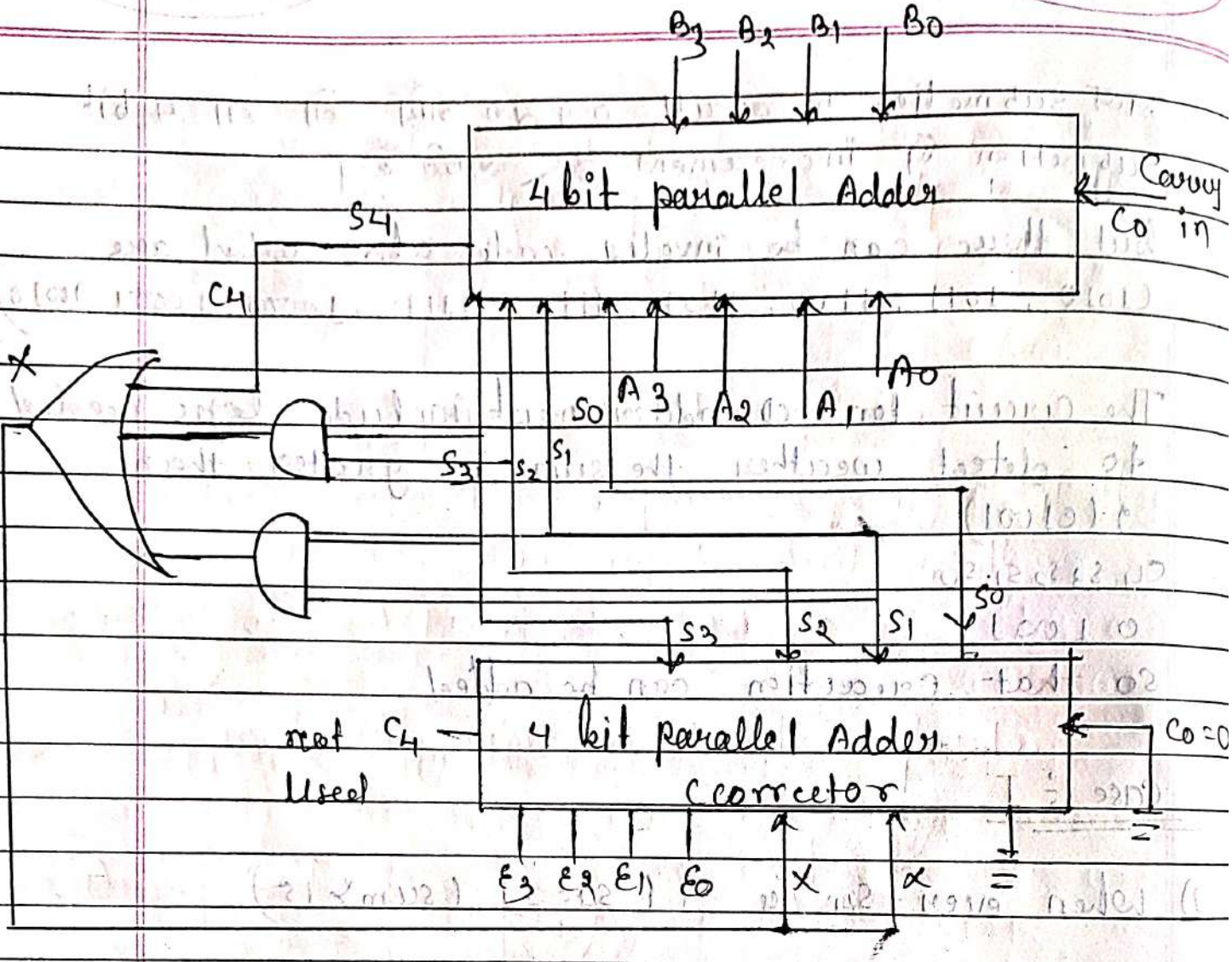
2) $S_3 \cdot S_2 = 1$ ($12 < sum < 15$)

3) $S_3 \cdot S_1 = 1$ (sum = 10, 11)

Hence the

Hence the Condition can be expressed

$$F = X = S_4 + S_3 \cdot S_2 + S_3 \cdot S_1$$



Block diagram of BCD Adder

logic diagram basically adder 2-4 bit parallel adder and correction detector circuit:

इस प्रकार परिपथ की संरचना के लिए 4 bit binary adder में अन्य 4 bit binary Adder द्वारा 0110 का योग किया जाता है परन्तु 0110 का योग केवल तभी किया जाता है जब द्वितीय योग का मान 1001 से अधिक हो अथवा यह उपर दिये गये Logic समीकरण को संतुष्ट करता हो अन्यथा प्रथम बाइनरी ऐडर के आउटपुट में 0000 का योग कर उसे अंतिम आउटपुट में प्राप्त कर लिया जाता है।

Page:
Date: / /

Magnitude Comparator (1-bit)

Truth table

input		output		
A	B	A < B	A = B	A > B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

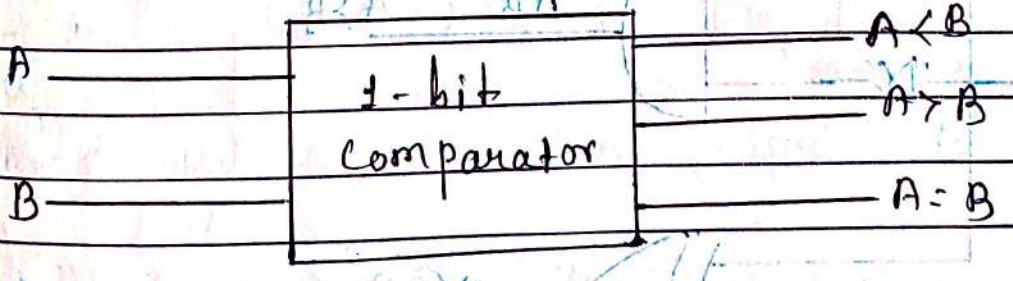
logical expression

$A < B = \overline{A}B$

$A > B = A\overline{B}$

$A = B = A \odot B$

Block Diagram of magnitude comperator (1-bit)



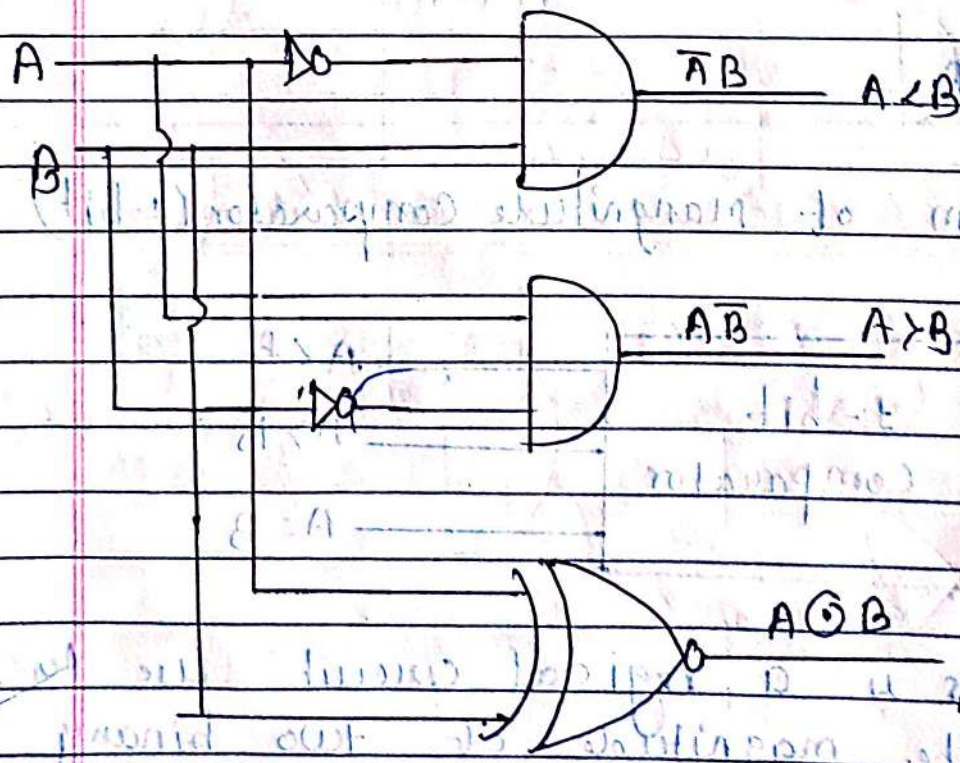
A comparator is a logical circuit use to compair the magnitude of two binary number.

it gives output logical 1 when two no. are equal or additionally provide output that signify (indecate) which no. is greater when equality does not hold.

Dibit : Comparator (1-bit comparator)

1-bit comparator compare only single bit that means it has two input (A, B) and three output ($A<B$, $A=B$, $A>B$)

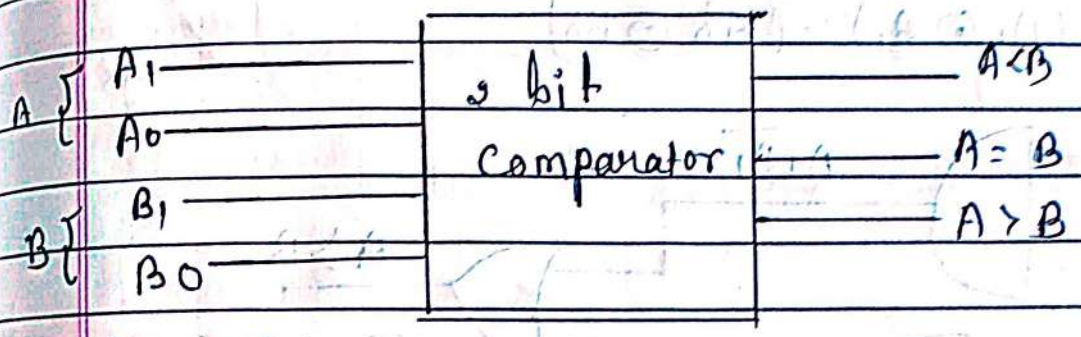
Logical Diagram of 1-bit Comparator



Page:
Date: / /

Magnitude Comparator (2-bit)

Block Diagram



Block diagram of magnitude comparator (2-bit)

two bit comparative has two binary number A & B each have two bit.

A = A1, A0

B = B1, B0

Case :

if A1 = 1 and B1 = 0 then output will be A>B

A1 = 0 and B1 = 1 then output will be A<B

if A1 = B1 and A0 = 1, B0 = 0 then output will be A>B

if A1 = B1 and A0 = B, B0 = 1 then output will be A<B

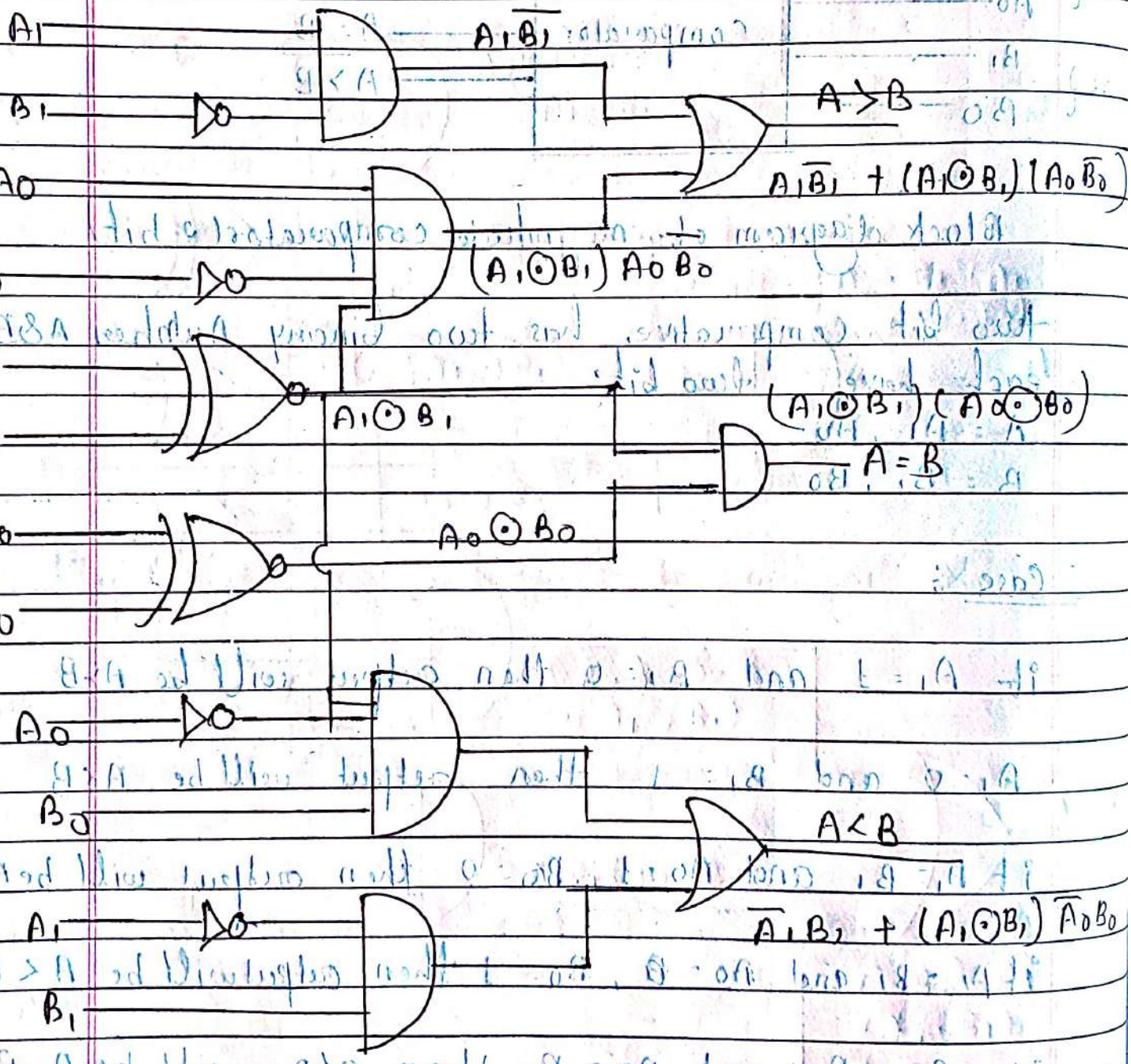
if A1 = B1 and A0 = B0 then O/P will be A=B

Logical expression

(G) $A<B = \overline{A_1} B_1 + (A_1 \odot B_1) \cdot \overline{A_0} B_0$

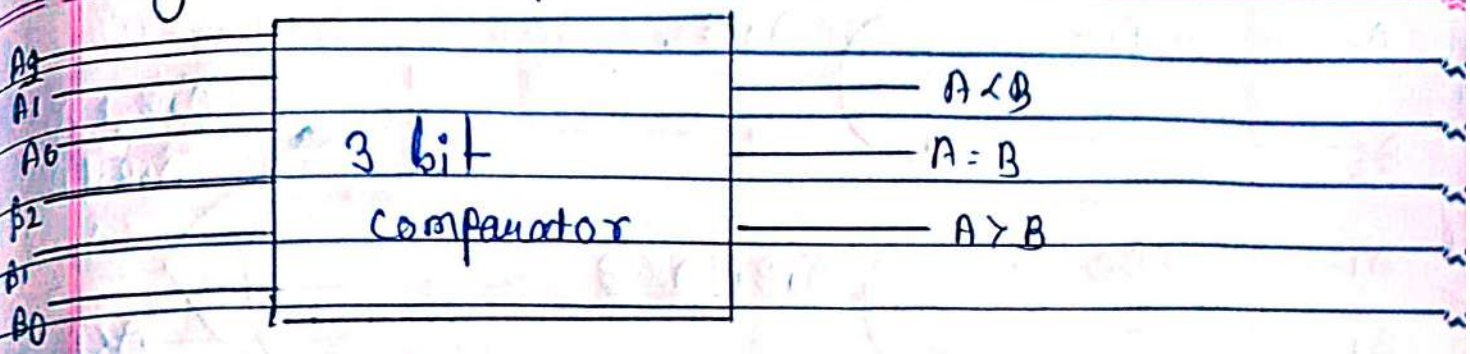
(L) $A>B = A_1 \overline{B_1} + (A_1 \odot B_1) \cdot A_0 \overline{B_0}$

(E) $A=B = (A_1 \odot B_1) \cdot (\overline{A_0} \odot B_0)$



Logical expression magnitude comparator (3 bit)

Magnitude Comparator (3 bit)



Block diagram magnitude comparator (3 bit)

three bit comparative has two binary numbers A & B each have three bit

$A = A_2, A_1, A_0$

$B = B_2, B_1, B_0$

Case:

if $A_2 = 1$ and $B_2 = 0$ then output will be $A > B$

$A_2 = 0$ and $B_2 = 1$ then output will be $A < B$

if $A_2 = B_2$ and $A_1 = 1$, $B_1 = 0$ then output will be $A > B$

if $A_2 = B_2$ and $A_1 = 0$, $B_1 = 1$ then o/p will be $A < B$

if $A_2 = B_2$ and $A_1 = B_1$ and $A_0 = 1$, $B_0 = 0$ then o/p $A > B$

if $A_2 = B_2$ and $A_1 = B_1$ and $A_0 = 0$, $B_0 = 1$ then o/p will be $A < B$

if $A_2 = B_2$ and $A_1 = B_1$ and $A_0 = B_0$ then o/p $A = B$

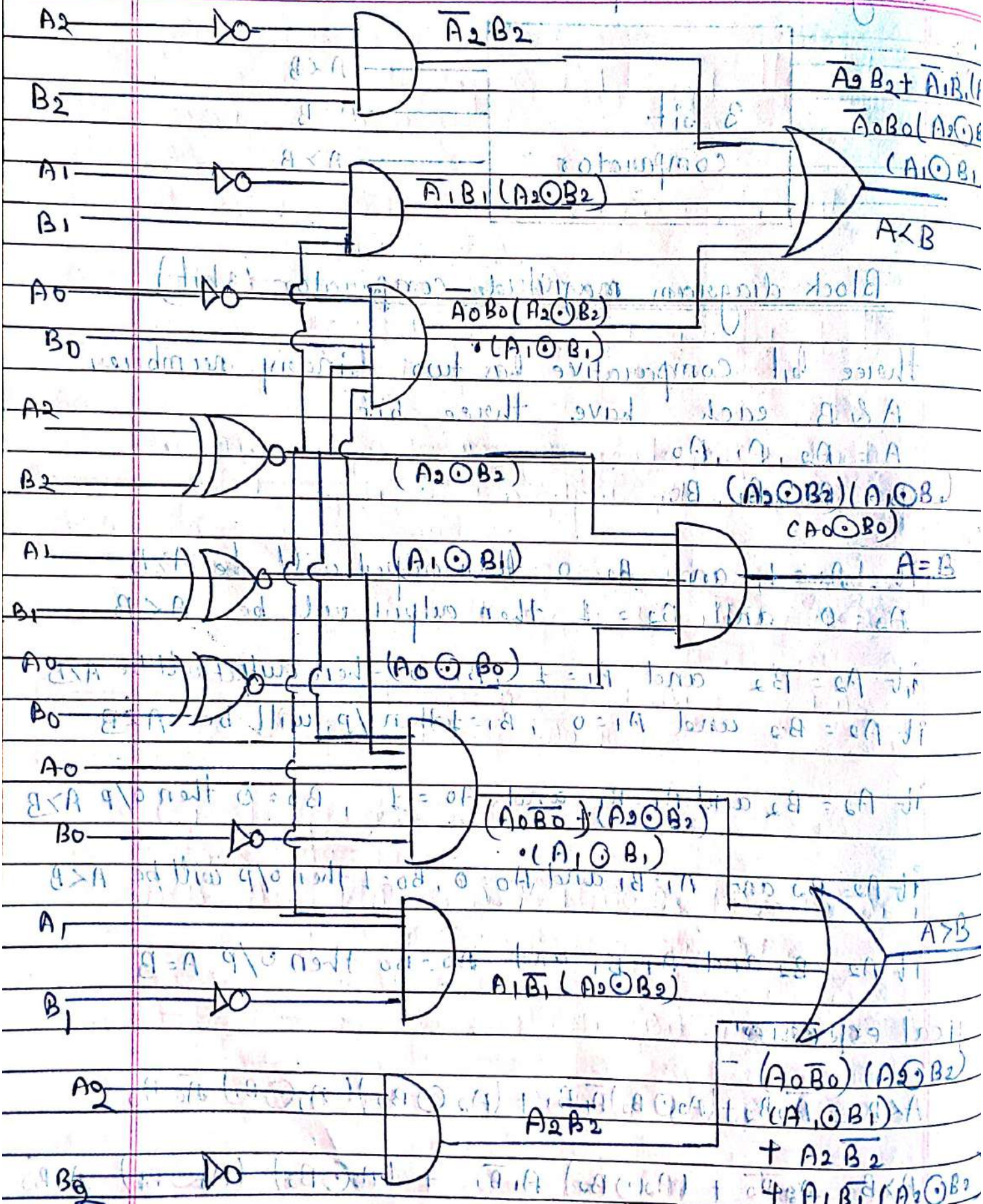
logical expression

$$A<B = \overline{A_2} B_2 + (A_2 \odot B_2)\overline{A_1} B_1 + (A_2 \odot B_2)(A_1 \odot B_1)\overline{A_0} B_0$$

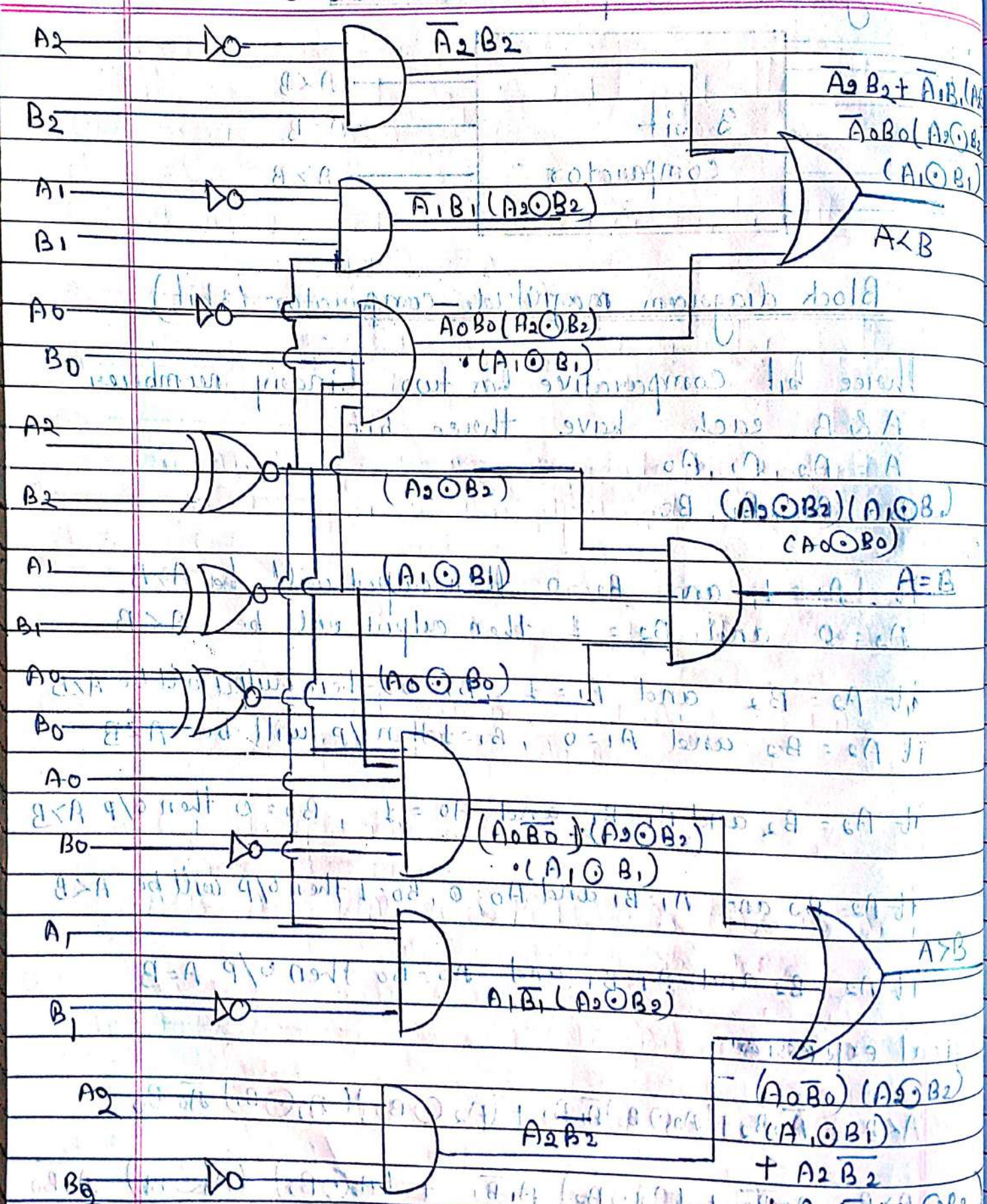
$$A>B = A_2 \overline{B_2} + (A_2 \odot B_2) A_1 \overline{B_1} + (A_2 \odot B_2)(A_1 \odot B_1) A_0 \overline{B_0}$$

$$A=B = (A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$$

Logical Diagram of magnitude comparator (3 bit)

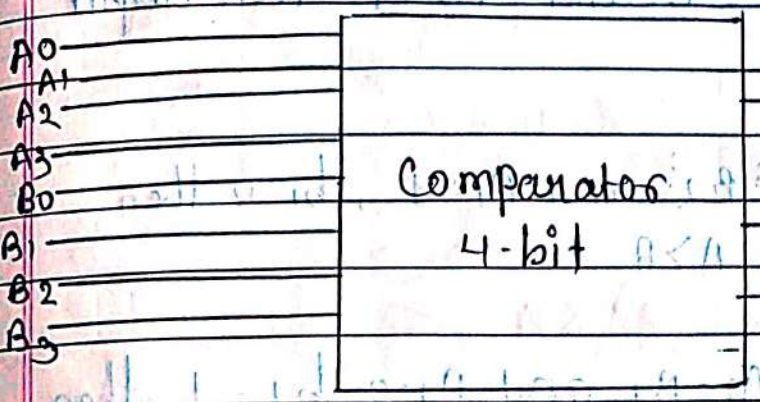


Logical Diagram of magnitude comparator (3 bit)



Page:
Date: / /

Magnitude Comperator (4-bit)



Block diagram of magnitude comperator (4-bit)

Four bit comparative has two binary number A & B each have four bit

A = A_3, A_2, A_1, A_0

B = B_3, B_2, B_1, B_0

Case :-

if $A_3 = 1$ and $B_3 = 0$ then output will be A > B

if $A_3 = 0$ and $B_3 = 1$ then output will be A < B

if $A_3 = B_3$ and $A_2 = 1$ and $B_2 = 0$ then output will be A < B

if $A_3 = B_3$ and $A_2 = 0$ and $B_2 = 1$ then output will be $A < B$

if $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = 1$, $b_1 = 0$ then output will be $A > B$

if $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = 0$ $b_1 = 1$ then output will be $A < B$

if $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 = 1$, i.e $B_1 = 0$ then output will be $A > B$

if $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 = 0$, $B_1 = 1$ then output will be $A < B$

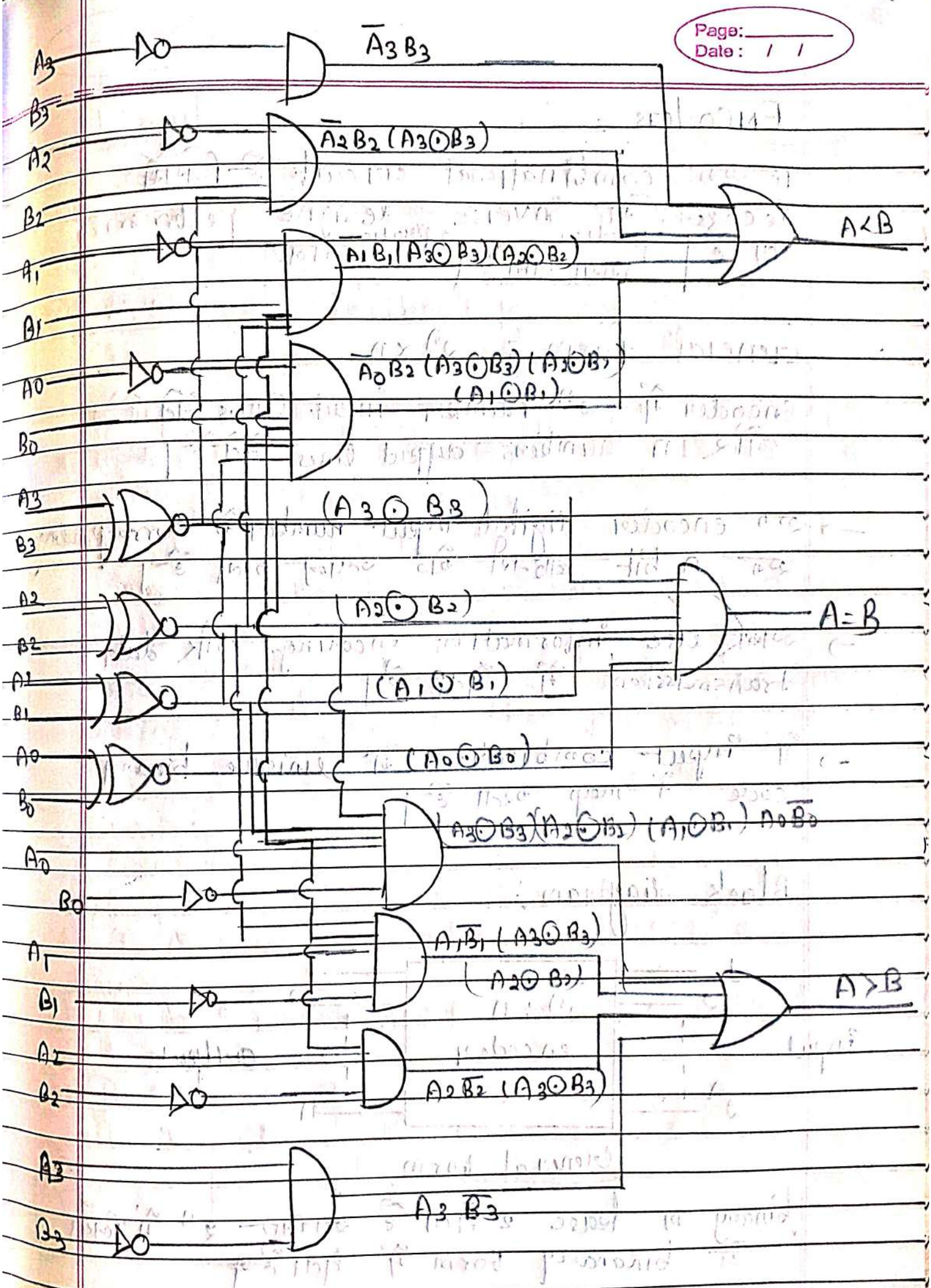
if $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 = B_0$ then output will be $A_0 = B_0$

Logical expression:

$$A < B = \overline{A_3} B_3 + (A_3 \odot B_3)\overline{A_2} B_2 + (A_3 \odot B_3)(A_2 \odot B_2)\overline{A_1} B_1 + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)\overline{A_0} B_0$$

$$A = B = (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$$

$$A > B = A_3 \overline{B_3} + (A_3 \odot B_3) A_2 \overline{B_2} + (A_3 \odot B_3)(A_2 \odot B_2) A_1 \overline{B_1} + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)\overline{A_0} B_0$$


A3
B3
A2
B2
A1
B1
A0
B0
$\overline{A_3}B_3$
$\overline{A_2}B_2(A_3 \odot B_3)$
$\overline{A_1}B_1(A_3 \odot B_3)(A_2 \odot B_2)$
$\overline{A_0}B_2(A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)$
A<B
$(A_3 \odot B_3)$
$(A_2 \odot B_2)$
$(A_1 \odot B_1)$
$(A_0 \odot B_0)$
A=B
$(A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)A_0\overline{B_0}$
$A_1\overline{B_1}(A_3 \odot B_3)$
$(A_2 \odot B_2)$
A>B
$A_2\overline{B_2}(A_3 \odot B_3)$
$A_3\overline{B_3}$

Encoders :

यह एक combinational circuit है जिसमें decoder का inverse operation perform होता है। design किया जाता है।

General form : $2^n \times n$

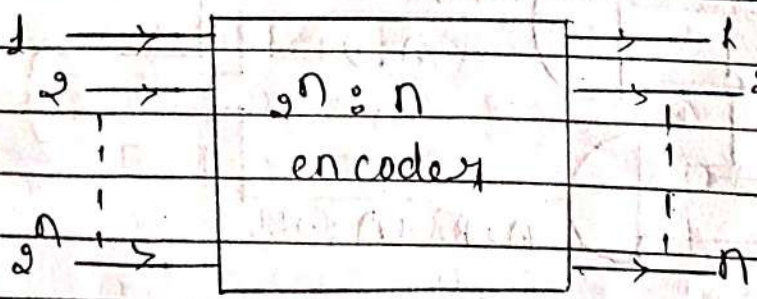
encoder में 2^n number input lines होते है और n numbers output lines होते है।

→ एक encoder digital input number के corresponding एक n bit बाइनरी कोड उत्पन्न करता है।

→ इसका use information encoding और data transmission में होता है।

→ ये input combination को unique binary code में map करता है।

Block diagram :



General form

Binary का base 2 होता है इसलिए 2^n में लिखे ये binary form में होता है।

→ Encoder एक input digital word को accept करता है और उसे एक n bit दूसरे digital word में परिवर्तित करता है।

→ The internal combinational circuit of the encoder is designed accordingly.

Encoder किसी भी original information को encrypt (~~code~~ original information को एक code के form में convert करना) करता है।

Binary information यानी n input lines को n output lines में बदलता है।

Encoder में एक ही समय पर एक ही input line activate (high) रहती है और उसी आधार पर output produce होता है।

Encoder के प्रकार

(1) 4 to 2 line Encoder or $4:2$ / $2^2:2$

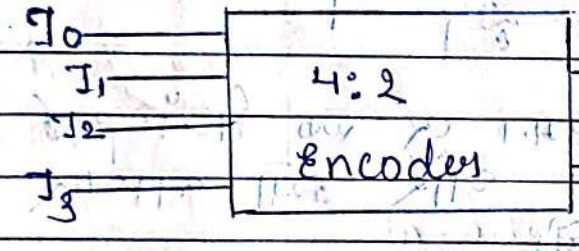
(2) 8 to 3 line Encoder or $8:3$ / $2^3:3$

(3) Decimal to BCD Encoder

(1) 4:2 Encoder

इस encoder में 4 input lines (I_0, I_1, I_2, I_3) और 2 output lines (Y_0, Y_1) होती है। जिसमें एक time पर एक ही input line active यानी high होती है।

इसका basic काम चार input lines में से active (high) वाली line को detect करना और उसके corresponding binary code को generate करना है।



Block diagram of 4:2 Encoder

Truth table

Input				Output		
I_3	I_2	I_1	I_0	Y_0	Y_1	Decimal
1	0	0	0	0	0	0
0	1	0	0	0	1	1
0	0	1	0	1	0	2
0	0	0	1	1	1	3

$Y_0 = 1$ तभी आएगा जब I_2 और I_3 input lines 1 पर है।

Y_0 का equation $\boxed{Y_0 = I_2 + I_3}$

$Y_1 = 1$ तभी होगा जब I_1 और I_3 input lines 1 पर है

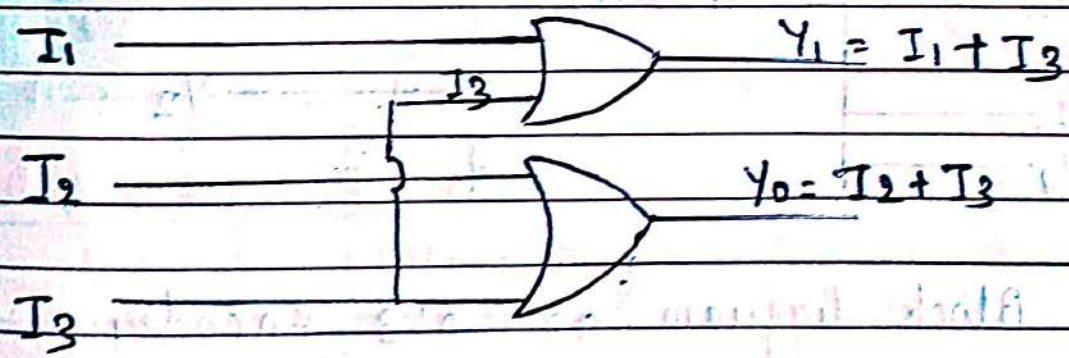
$\boxed{Y_1 = I_1 + I_3}$

Logical Expression :

$Y_0 = I_2 + I_3$

$Y_1 = I_1 + I_3$

Y_0 और Y_1 के equation को OR logic gate मदद से implement करते है।



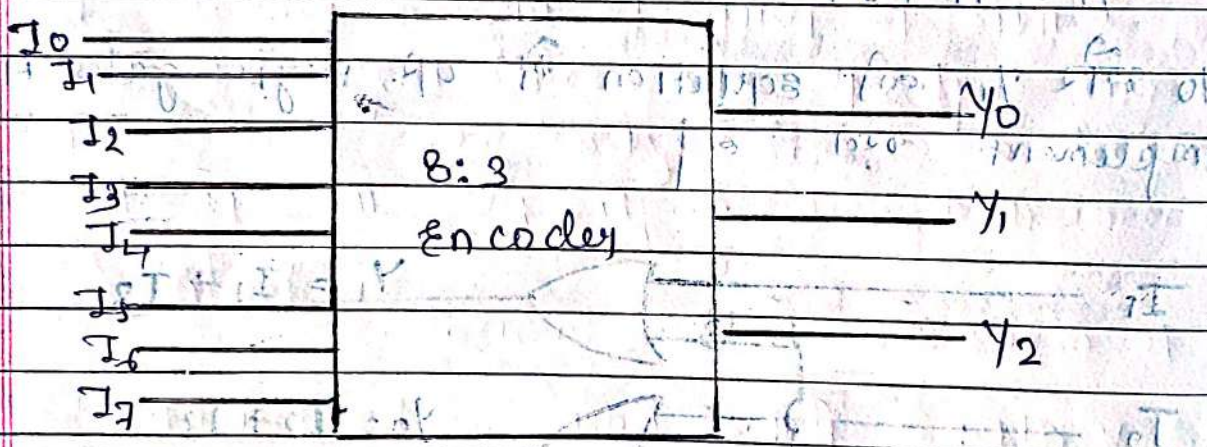
Logical Diagram of 4:2 Encoder

(2) 8:3 Encoder या Octal to Binary Encoder

→ इस encoder को octal to Binary encoder भी कहा जाता है। इसमें

→ इसमें 8 input lines ($I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7$) और 3 output lines ($Y_0, Y_1, Y_2$) होती है।

→ इसका basic काम यह है कि 8 input lines में active (high) वाली line को detect करना और उसके corresponding binary code को generate करना।



Block diagram of 8:3 encoder

ये भी एक truth table के through काम करता है जिस में हर possible input combination के लिए output define होता है।

इस encoder में भी एक समय पर एक ही कनेक्ट लाइन active या फी high होती है।

Page:
Date: / /

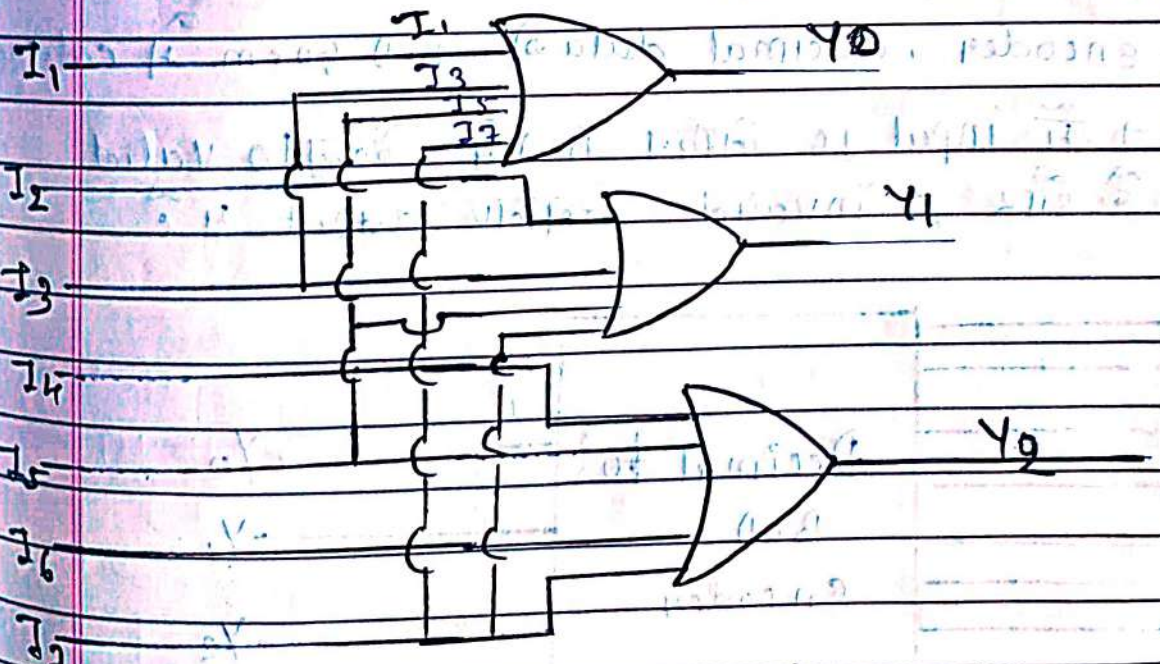
Truth table

Input								Y_2	Y_1	Y_0
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7			
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

Logical Expression : $Y_2 = I_4 + I_5 + I_6 + I_7$

$Y_1 = I_2 + I_3 + I_6 + I_7$

$Y_0 = I_1 + I_3 + I_5 + I_7$



Logical diagram of 8:2 encoder

Encoder

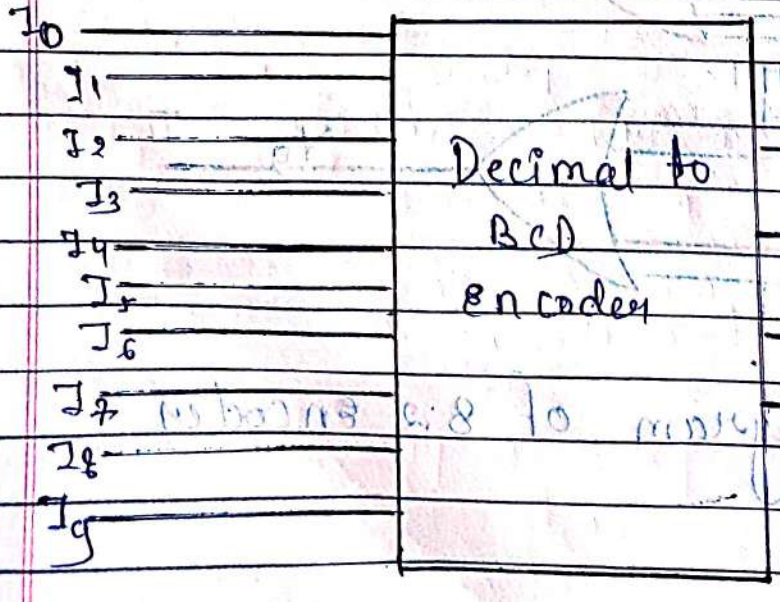
- An encoder is a digital device whose input can be decimal digit or alphabetic character and whose output are coded representation of those input (Binary)
- General form of encoder is $(2^n : n)$ where
  2^n is equal to number of output
  n is equal to number of input

(3) Decimal to BCD Encoder

इन encoder के नाम से समझ में आता है कि इसमें 10 input lines और 4 output lines होती है हर एक इनपुट line एक decimal digit को और 4 output line BCD code को represent करती है एक

→ यह encoder, decimal data को BCD form में code करता है

→ $2^4 : 4$ में input 16 लेकिन BCD में केवल 10 Valid होते है और 6 invalid इसलिए output 4 है



Page:
Date: / /

Truth table

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	Y_3	Y_2	Y_1	Y_0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

logical expression :

इस output lines Y_0, Y_1, Y_2, Y_3 के expression को truth table की मदद से लिखते है

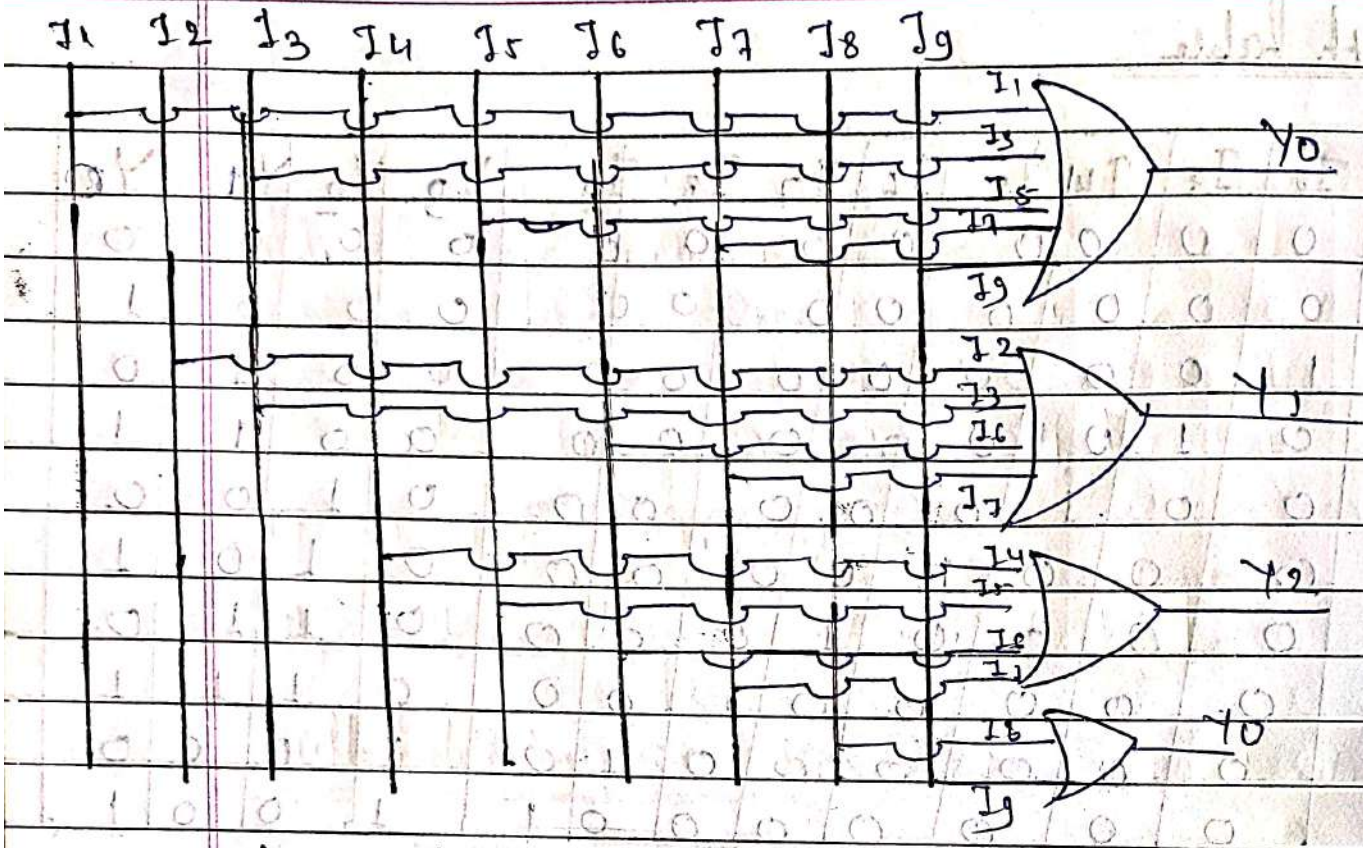
हम Y_0, Y_1, Y_2 और Y_3 output lines को active यानी high(1) होगा जब $E_0(I_1, I_3, I_5, I_7, I_9)$, $E_1(I_2, I_3, I_6, I_7)$, $E_2(I_4, I_5, I_6, I_7)$ $E_3(I_8, I_9)$ input lines active यानी 1 हो

$Y_0 = I_1 + I_3 + I_5 + I_7 + I_9$

$Y_1 = I_2 + I_3 + I_6 + I_7$

$Y_2 = I_4 + I_5 + I_6 + I_7$

$Y_3 = I_8 + I_9$



logical diagram of Decimal to BCD Encoder

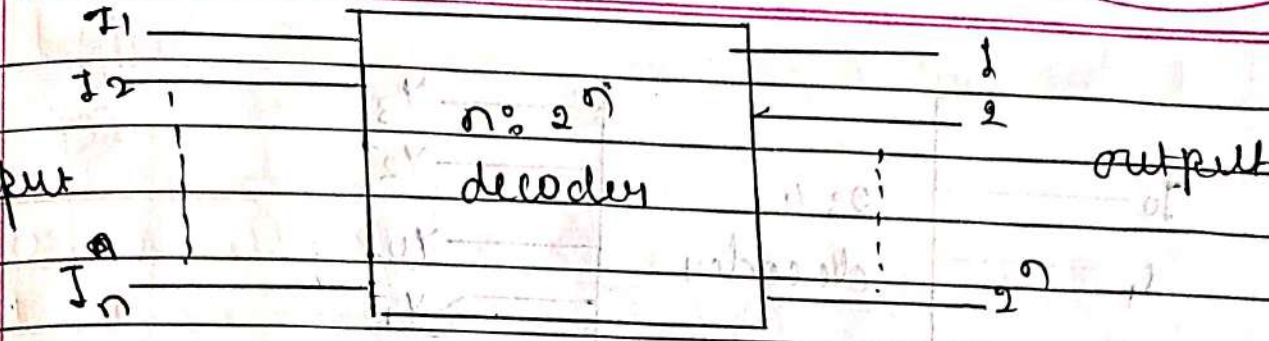
Decoder

Decoder is a logical circuit that convert n bit input code into 2^n number of output line.

→ Decoder, Encoder के समान ही code परिवर्तक है जो प्रोसेसिंग यूनिट के आउटपुट बाइनरी code को decimal में परिवर्तित करता है।

→ यह एक combinational circuit है।

→ इसमें ही binary decoding होती है जिसमे हर एक Binary combination के लिए एक specific output line होती है।



Block diagram of $n:2^n$

इसमें n input है 2^n output है।

Type of Decoder

(1) 2:4 Decoder

(2) 3:8 Decoder 8 to 4 decoder

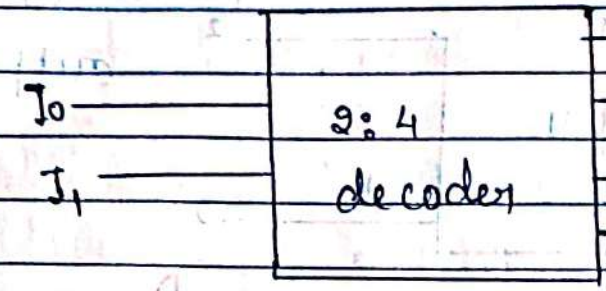
(3) BCD to Decimal decoder

(1) 2:4 Decoder :-

→ यह एक electronic circuit है जो दो input lines को use करते है और उनके combination के basis पर 4 output lines produce करता है।

→ ये decoder specifically 2 input lines को 4 output lines को handle करता है।

जब हम इसमें दो input line पर input देते है तो ये एक input combination के लिए एक corresponding output line activate होती है।



Block diagram of 2:4 decoder

इसमें दो input (I_1, I_0) है

और चार output (Y_3, Y_2, Y_1, Y_0) है

Truth table :

input		output			
I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

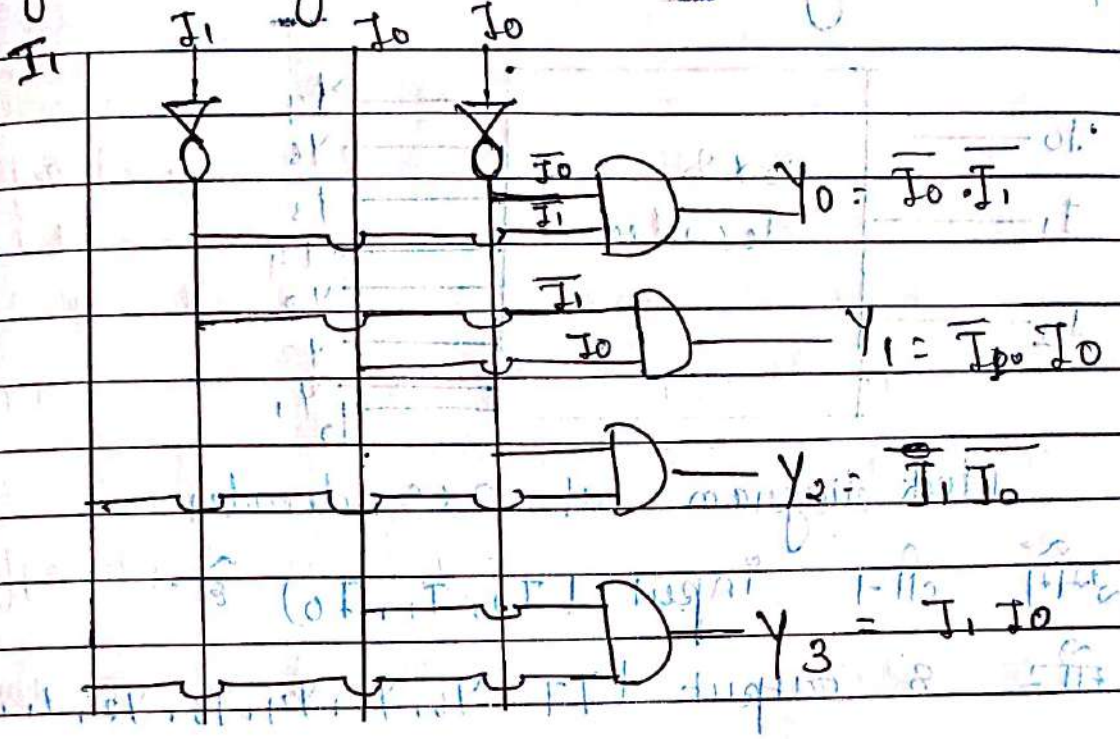
Logical expression :

$Y_3 = I_1 \cdot I_0$ $\quad Y_2 = I_1 \overline{I_0}$

$Y_1 = \overline{I_1} I_0$ $\quad Y_0 = \overline{I_1} \cdot \overline{I_0}$

The Boolean expression for the four output is

Logical Diagram of 2:4 Decoded



Logical Diagram

(2) 3:8 Decoder :-

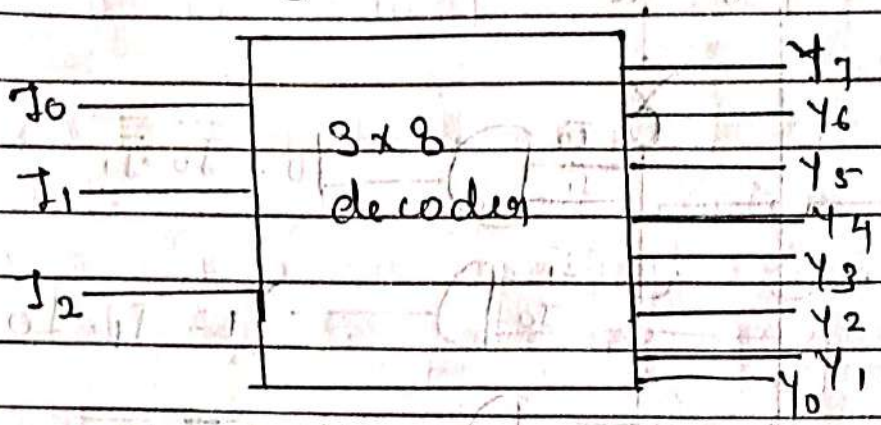
3:8 decoder एक electronic circuit है जो 3 input lines को use करता है और उनके combinations के आधार पर 8 output lines produce करता है।

→ इसका basic काम है 3 bit binary input combinations को decode करके उनके corresponding output line को activate करना

This decoder acts as a min term generator where each output corresponds to a specific min term.

Page:
Date: / /

Block Diagram :-



Block diagram of 3 x 8 decoder

उपर्युक्त तीन input (I2, I1, I0) के साथ 8 output (Y7, Y6, Y5, Y4, Y3, Y2, Y1, Y0)

Truth table

Input			Output						
I2	I1	I0	Y7	Y6	Y5	Y4	Y3	Y2	Y1
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	1	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

Logical expression :-

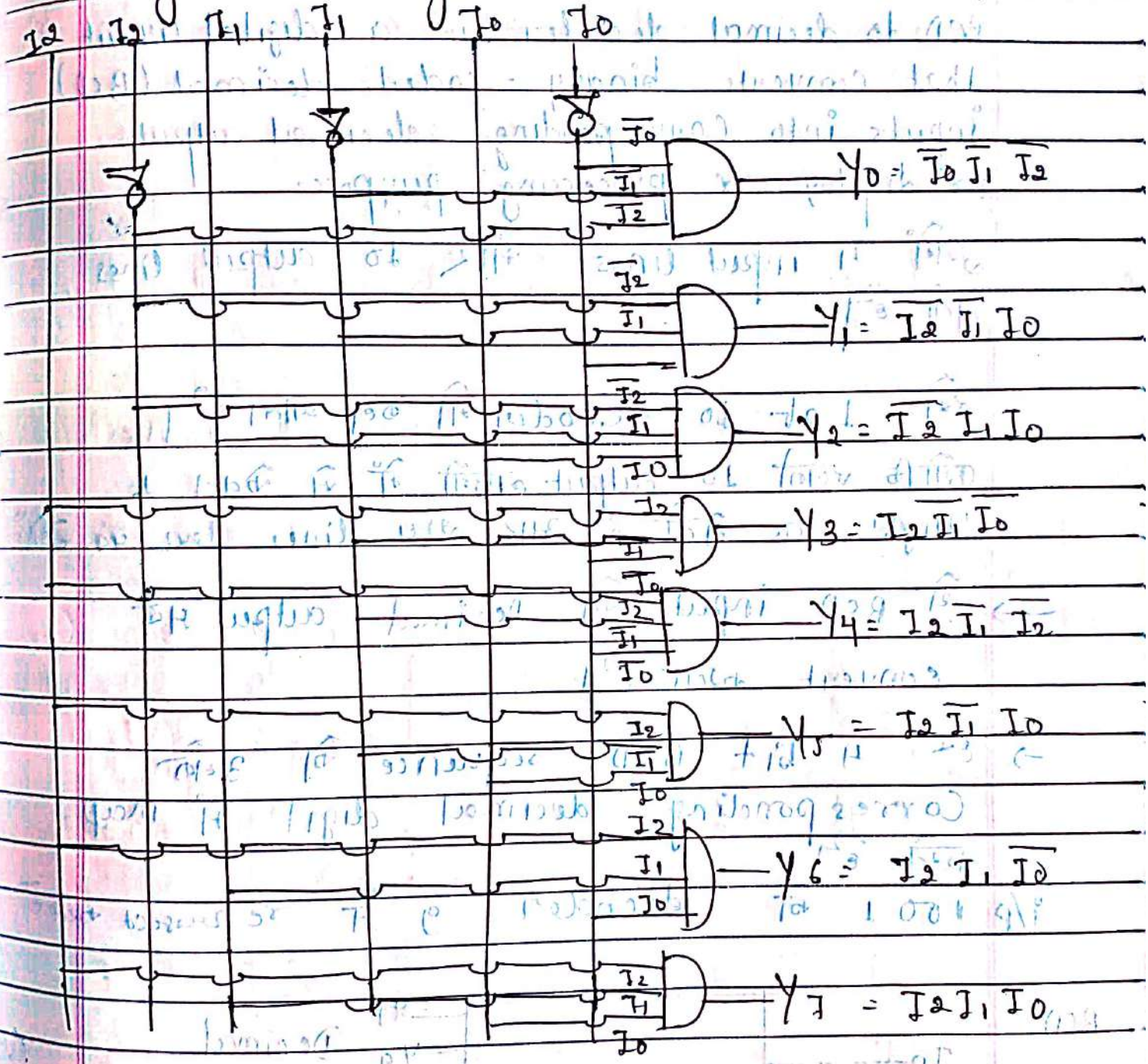
$Y_0 = \overline{I_2}\,\overline{I_1}\,\overline{I_0}$ $Y_1 = \overline{I_2}\,\overline{I_1}\,I_0$ $Y_2 = \overline{I_2}\,I_1\,\overline{I_0}$

$Y_3 = \overline{I_2}\,I_1\,I_0$ $Y_4 = I_2\,\overline{I_1}\,\overline{I_0}$ $Y_5 = I_2\,\overline{I_1}\,I_0$

Page:
Date: / /

$Y_6 = I_2 I_1 \overline{I_0}$ $Y_7 = I_2 I_1 I_0$

Logical Diagram of 3×8 Decoder



Logical Diagram 3×8 Decoder

(3) BCD to Decimal Decoder :-

BCD to decimal decoder is a digital circuit that converts binary-coded decimal (BCD) inputs into corresponding decimal outputs for display or processing purpose.

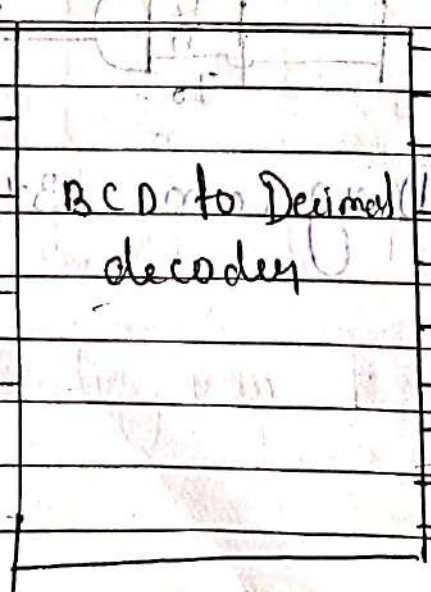
इसमें 4 input lines और 10 output lines होती है।

इसे 1 of 10 decoder भी कहा जाता है।
क्योंकि इसकी 10 output लाइनों में से केवल 1 'High' पर होती है और शेष lines low पर होती

→ ये BCD input को Decimal output में convert करता है।

→ ये 4 bit BCD sequence को उसके Corresponding decimal digit में map करते है।

i/p 1001 को decoder 9 में represent करते है



Block diagram of BCD to Decimal decoder.

Page:
Date: / /

ये decoder digital electronics में use होता है। especially display devices को control करने के लिए जहाँ decimal representation required होती है।

Truth table

Input				Output									
I_3	I_2	I_1	I_0	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0

Logical Expression :

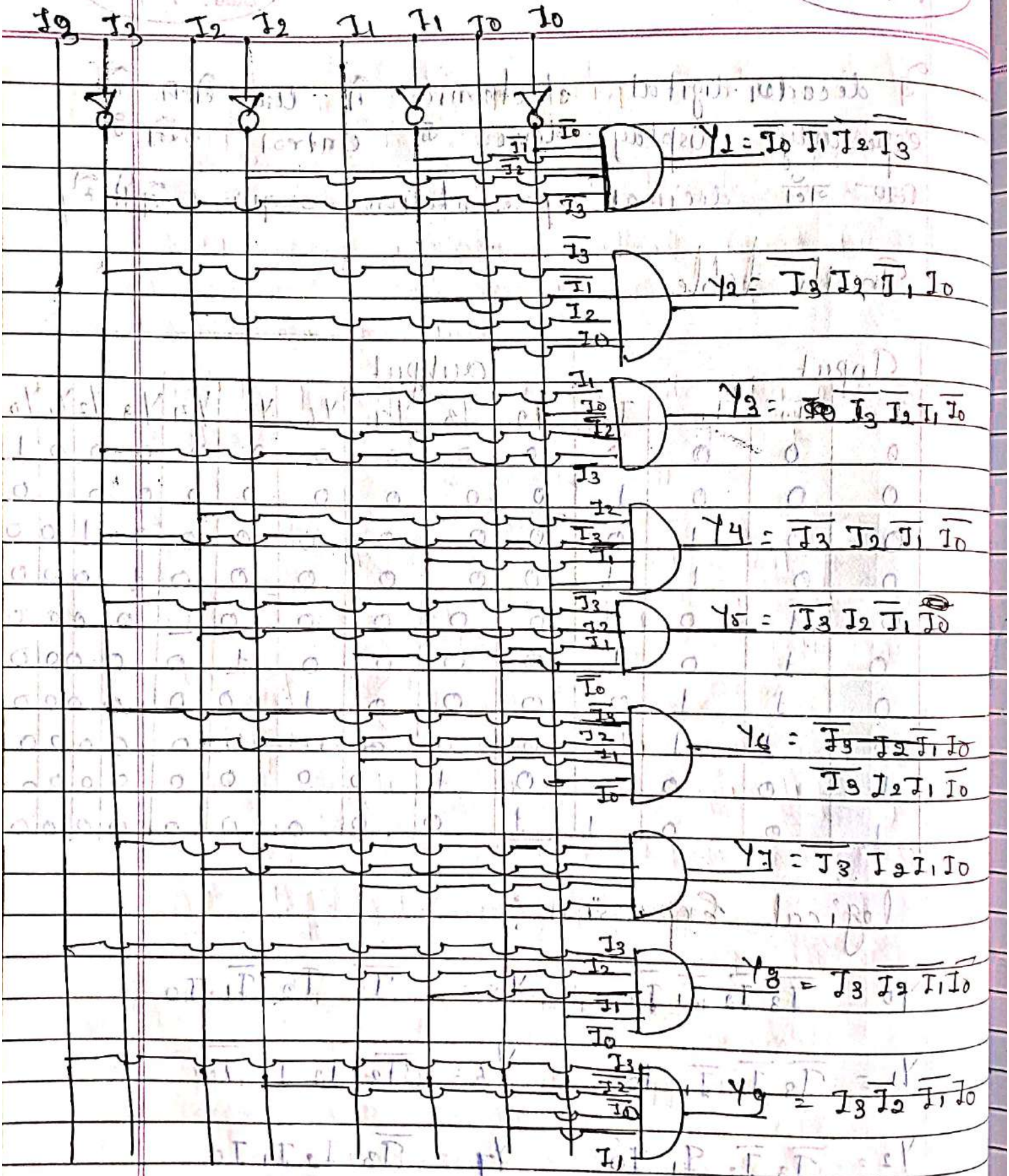
$Y_0 = \overline{I_3}\,\overline{I_2}\,\overline{I_1}\,\overline{I_0}$ $Y_5 = \overline{I_3}\,I_2\,\overline{I_1}\,I_0$

$Y_1 = \overline{I_3}\,\overline{I_2}\,\overline{I_1}\,I_0$ $Y_6 = \overline{I_3}\,I_2\,I_1\,\overline{I_0}$

$Y_2 = \overline{I_3}\,\overline{I_2}\,I_1\,\overline{I_0}$ $Y_7 = \overline{I_3}\,I_2\,I_1\,I_0$

$Y_3 = \overline{I_3}\,\overline{I_2}\,I_1\,I_0$ $Y_8 = I_3\,\overline{I_2}\,\overline{I_1}\,\overline{I_0}$

$Y_4 = \overline{I_3}\,I_2\,\overline{I_1}\,\overline{I_0}$ $Y_9 = I_3\,\overline{I_2}\,\overline{I_1}\,I_0$



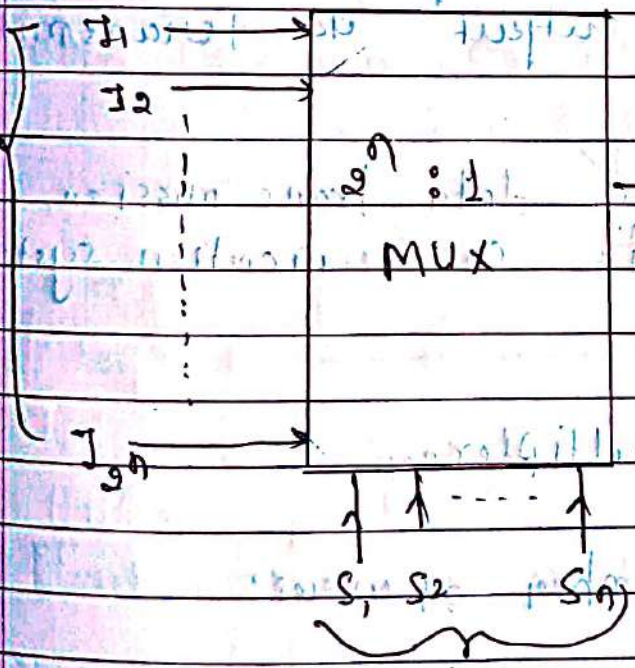
Logical diagram of BCD to Decimal Decoder

Multiplexer (MUX) / Data selector

→ multiplexer is a special type of combination circuit.

→ Multiplexer or Data selector is a logic circuit that accept multiple data input (2^n) allow only one of them to get through output.

इसमें 2^n input होता है और 1 output होता है और इसमें n selection lines / select input होते है



Block diagram of 2^n:1 multiplexer

A multiplexer is a digital circuit which select one of the n data inputs and routes it to the output. The selection of one of the n inputs is done by the select inputs

Page:
Date: / /

→ multiplexer एक ऐसा परिपथ है जिसमें अनेक input तथा केवल एक output होती है।

→ यह एक data selector परिपथ है।

→ इसमें select input / control signal की प्रयुक्त कर किसी input को output प्राप्त किया जा सकता है।

→ परिपथ का output (जो एक selected input होती है) control terminal पर प्रयोग किये गये digital code पर निर्भर करती है।

→ ये एक switch की तरह काम करता है, जिसमें एक specific input line को select करके उसका data output पर transmit होता है।

→ multiplexer का use data transmission, signal routing और communication system में होता है।

Advantages of Multiplexer

1) It reduce the number of wires.

2) so it reduce the circuit complexity and cost

3) we can implement many combinational circuits using multiplexer

4) It simplifies the logic design

5) It does not need the k-map and simplification.

Page: _____
Date: / /

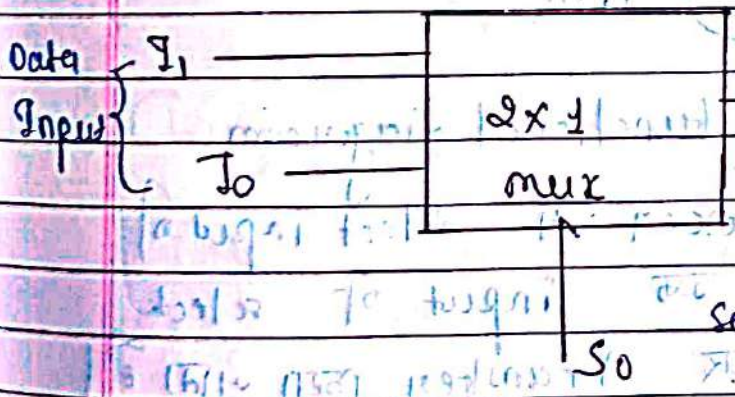
Types of Multiplexer :

1. 2 : 1 mux
2. 4 : 1 mux
3) 8 : 1 mux

(1) 2 : 1 multiplexer

इसमें 2 input signal होते है, इस circuit में किसी एक input single को select किया जाता है तथा उसी selected input को, output पर transfer किया जाता है। input selection का कार्य control signal / select input करता है

Block diagram of 2 : 1 mux



Truth table of 2 : 1 mux

i/p	o/p
S_0	Y
0	I_0
1	I_1

इसमें 2^n यानी 2 है इसलिए input 2 और select input n होगा तो $n = 1$ है।

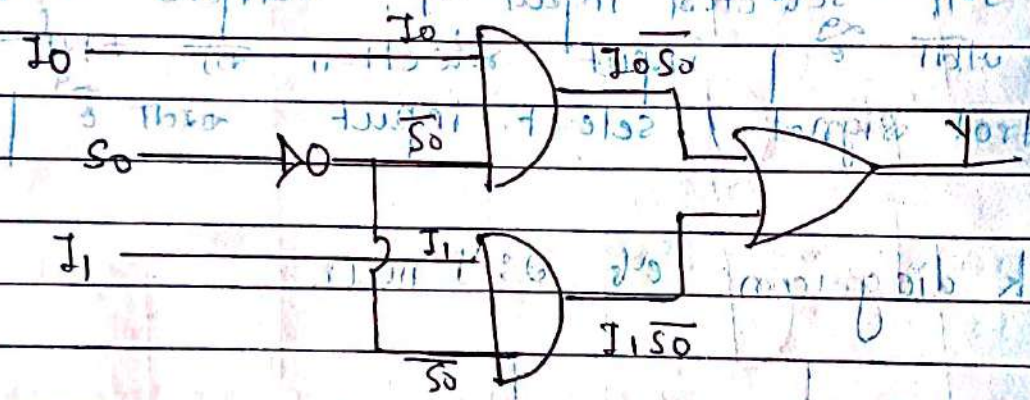
Logical expression :

$Y = I_0$ when $S_0 = 0$ then $Y = I_0 \overline{S_0}$

$Y = I_1$ when $S_0 = 1$ then $Y = I_1 S_0$

$$\boxed{Y = I_0 \overline{S_0} + I_1 S_0}$$

Logical diagram of 2x1 multiplexer :



Logical diagram या Functional diagram

→ इस प्रकार multiplexer में select input की सहायता से केवल एक input को select कर उसे output पर transfer किया जाता है।

(2) 4-to-1 multiplexer (4:1 MUX)

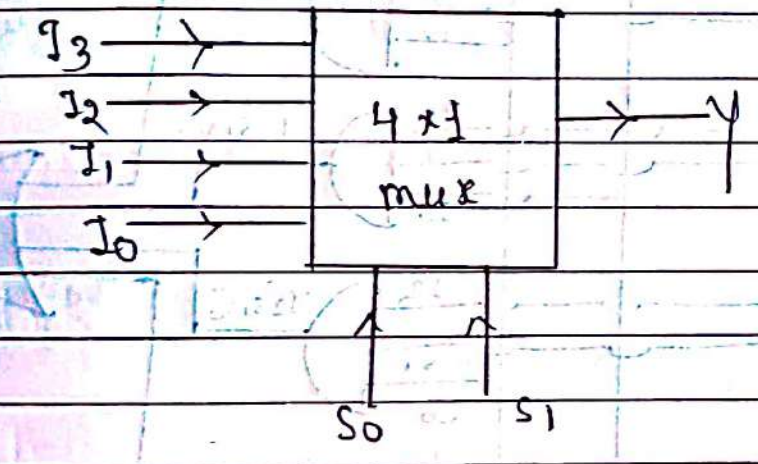
इसमें 4 input signals होते हैं जो I_0, I_1, I_2, I_3 तथा ये $2^2 : 1$ है अर्थात् $n = 2$ है तो इसमें 4 input और $n = 2$ यानि दो select input / control singal S_0, S_1 होगा और एक output Y होगा।

Page:
Date: / /

→ 4:1 multiplexer, 4 input lines से एक output line पर data transmit करने का digital circuit है।

→ इसमें control signal / select input से किसी भी input line को choose किया जा सकता है, जिसका output main line पर आता है।

Block diagram of 4x1 mux :-



Truth table of 4x1 mux

input		output
S1	S0	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

$Y = I_0$ when $S_0 = 0, S_1 = 0$ = then $Y = I_0 \overline{S_0}\,\overline{S_1}$

$Y = I_1$ when $S_1 = 0, S_0 = 1$ then $Y = I_1 \overline{S_1} S_0$

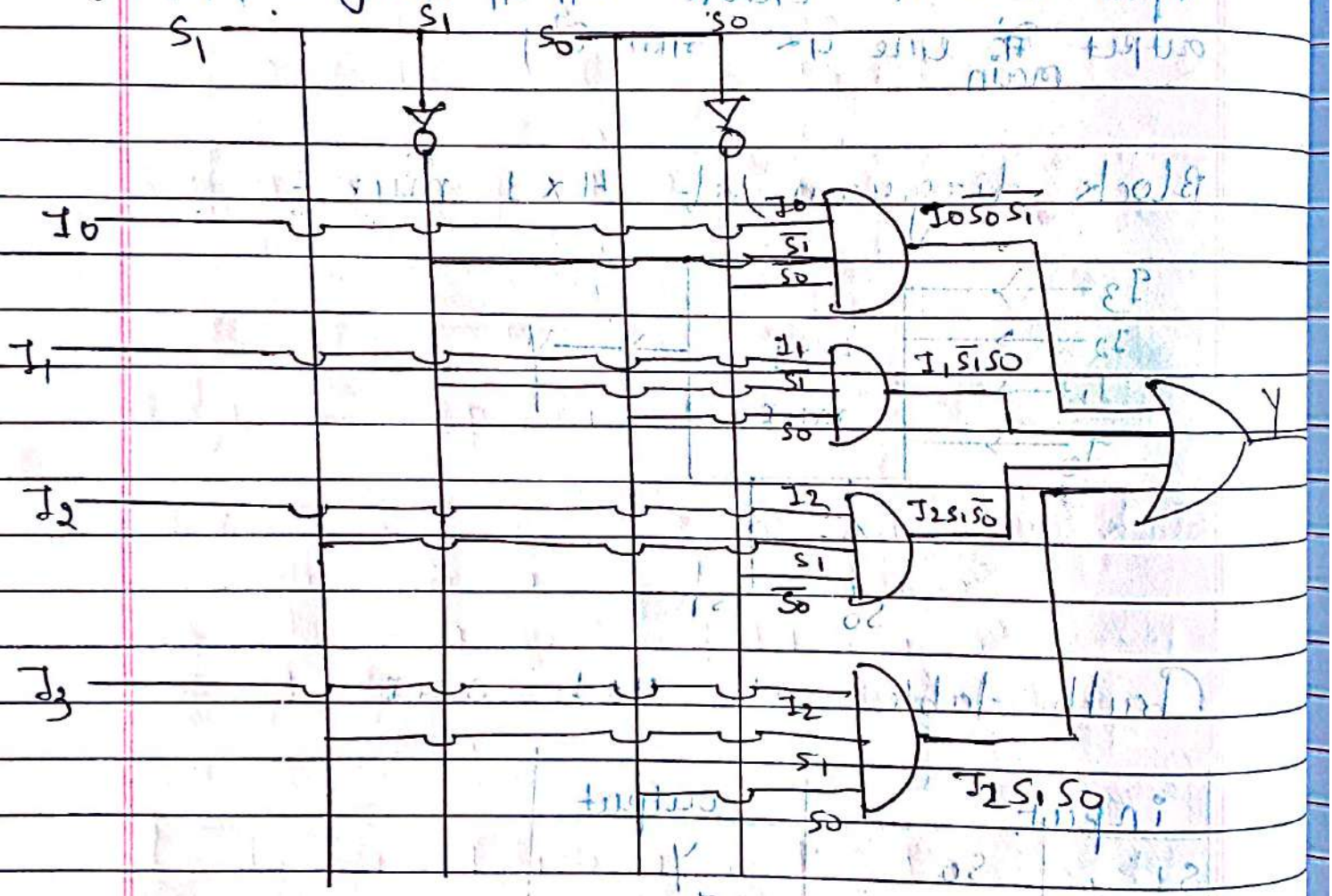
$Y = I_2$ when $S_1 = 1, S_0 = 0$ then $Y = I_2 S_1 \overline{S_0}$

$Y = I_3$ when $S_1 = 1, S_0 = 0$ then $Y = I_3 S_1 S_0$

Logical expression :

$$Y = I_0 \bar{S_1} \bar{S_0} + I_1 \bar{S_1} S_0 + I_2 S_1 \bar{S_0} + I_3 S_1 S_0$$

Logic block diagram of 4 x 1 multiplexer



Logical Diagram of 4x1 mux

Page:
Date: / /

(3) 8 x 1 multiplexer

Def :- An 8:1 mux is a digital circuit that combines 8 input lines into a single output line.

⇒ ये 8 input lines को एक single output line पर switch करने वाला digital circuit है।

⇒ ये mux, एक binary control signal के through specific input line को choose करता है और data को output पर transmit करता है।

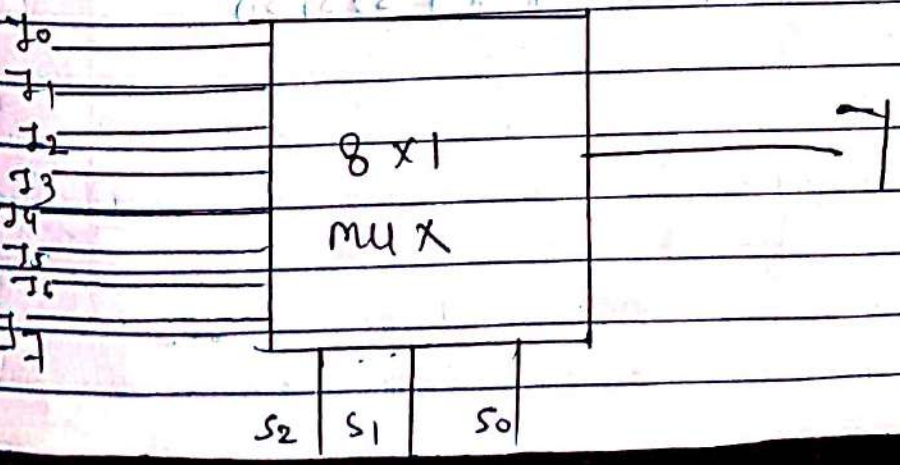
⇒ input lines :- 8 input line हैं जिनसे data आता है।

⇒ output lines :- एक ही output line है जहाँ से selected input का data pass होता है।

⇒ Control signal :- इसे इस्तेमाल करके एक particular input line को select किया जाता है।

⇒ Selection :- Control signal के अनुसार किसी भी input line का data output पर transmit होता है।

Block diagram of 8 x 1 multiplexer



Page:
Date: / /

Truth table :

S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Logical Expression :

$Y = I_0 \overline{S_2}\ \overline{S_1}\ \overline{S_0}$ $\quad Y = I_5 S_2 \overline{S_1} S_0$

$Y = I_1 \overline{S_2}\ \overline{S_1} S_0$ $\quad Y = I_6 S_2 S_1 \overline{S_0}$

$Y = I_2 \overline{S_2} S_1 \overline{S_0}$ $\quad Y = I_7 S_2 S_1 S_0$

$Y = I_3 \overline{S_2} S_1 S_0$

$Y = I_4 S_2 \overline{S_1}\ \overline{S_0}$

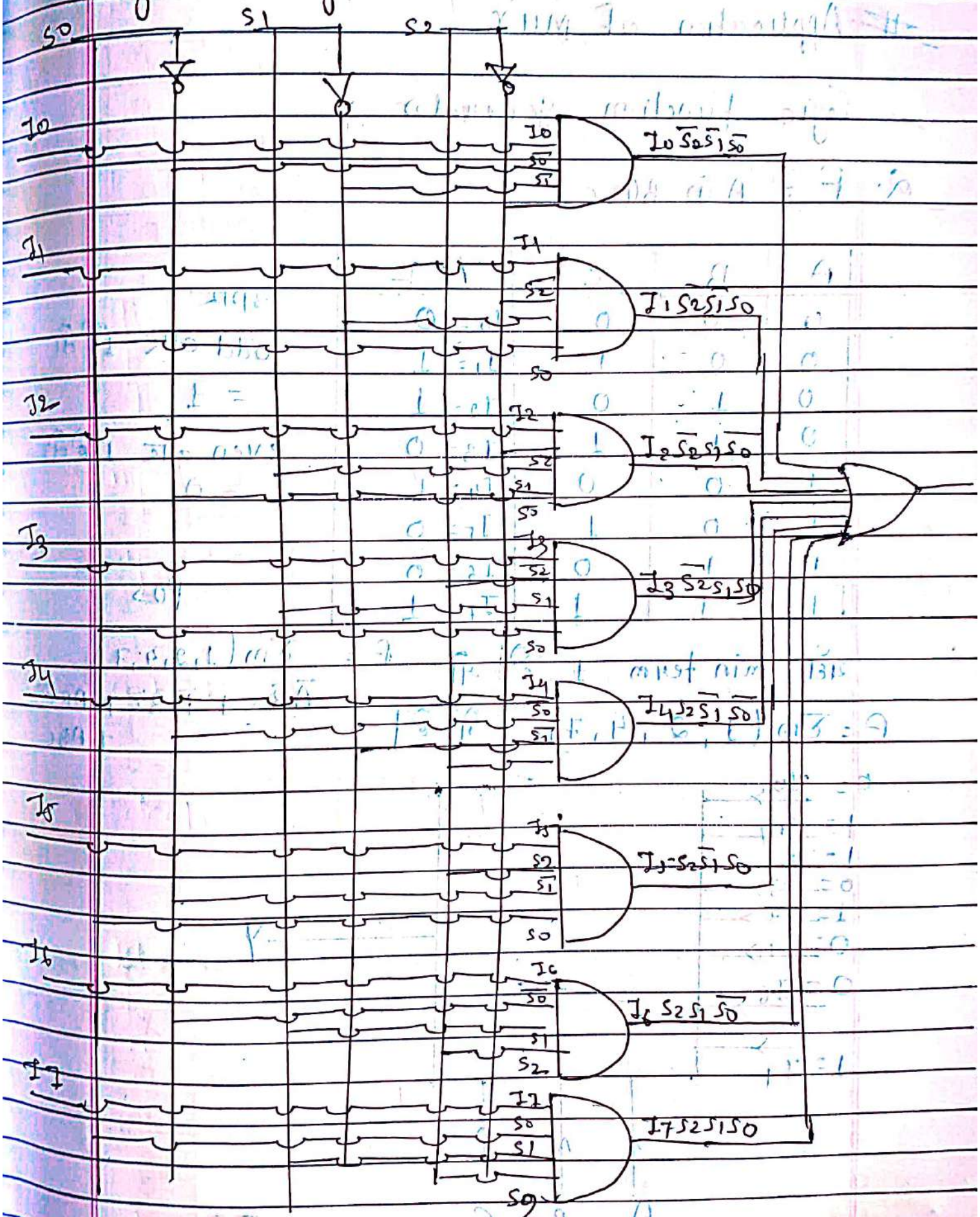
$Y = I_0 \overline{S_2}\ \overline{S_1}\ \overline{S_0} + I_1 \overline{S_2}\ \overline{S_1} S_0 + I_2 \overline{S_2} S_1 \overline{S_0}$

$I_3 \overline{S_2} S_1 S_0 + I_4 S_2 \overline{S_1}\ \overline{S_0} + I_5 S_2 \overline{S_1} S_0$

$+ I_6 S_2 S_1 \overline{S_0} + I_7 S_2 S_1 S_0$

Page:
Date: / /

logical diagram :



Page:
Date: / /

Question

Application of MUX

Logic function generator

Q. $F = A \oplus B \oplus C$

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

अगर
odd वाले 1 दो
= 1
even वाले 1 दो
= 0

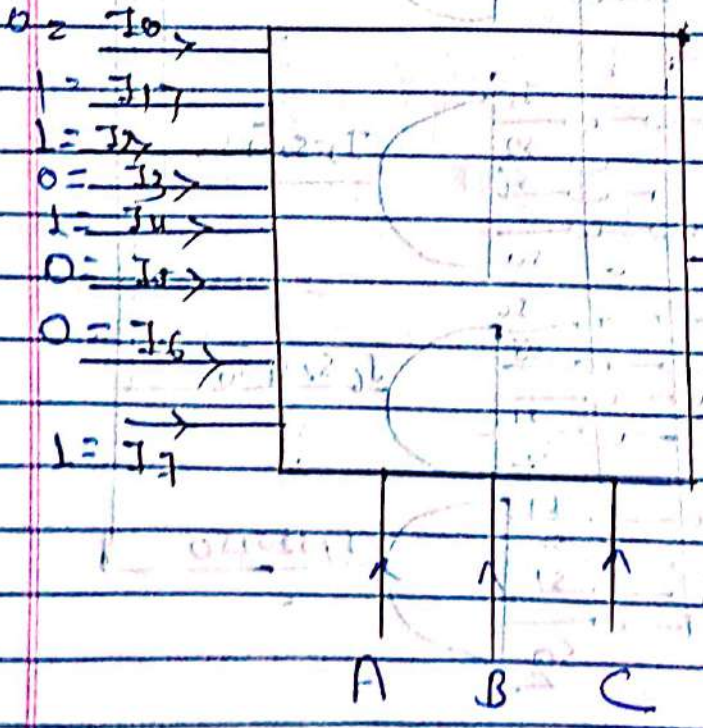
POS

यहाँ min term 1, 2 को $\overline{A}$ पर $\overline{B}$ पर $\overline{C}$ पर

$F = \Sigma m(1,2,4,7)$

$= \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC$

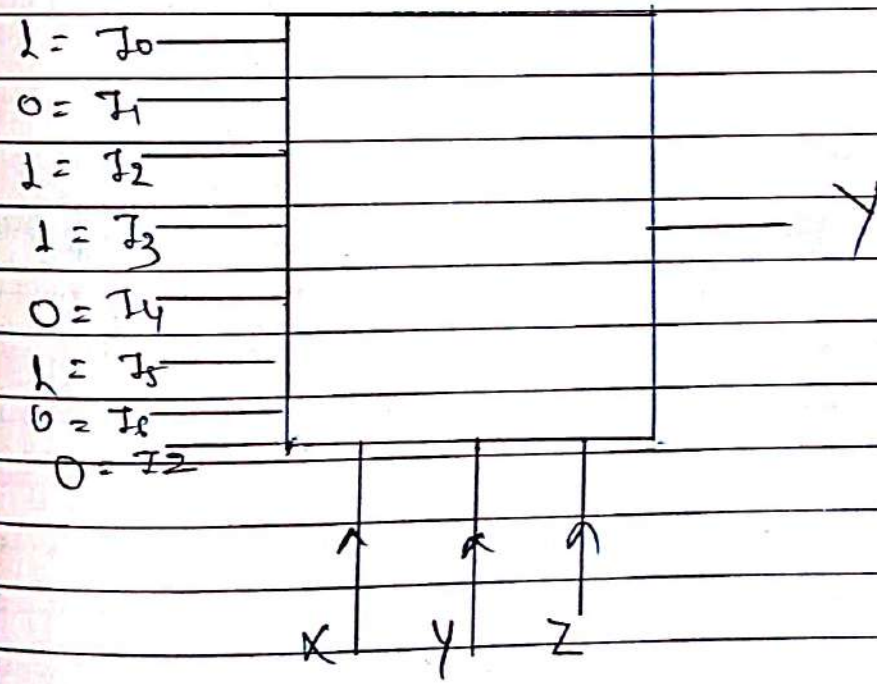
$P = \Sigma m(1,2,4,7)$



Page:
Date: / /

Question: $f(\Sigma m(0,2,3,5))$

x	y	z	F
0	0	0	$I_0 = 1$
0	0	1	$I_1 = 0$
0	1	0	$I_2 = 1$
0	1	1	$I_3 = 1$
1	0	0	$I_4 = 0$
1	0	1	$I_5 = 1$
1	1	0	$I_6 = 0$
1	1	1	I_7 0



Q. $F = \Pi M(4,7,9,10)$

x	y	z	M	F
0	0	0	0	$I_0 = 0$
0	0	0	1	$I_1 = 1$
0	0	1	0	$I_2 = 1$
0	0	1	1	$I_3 = 1$
0	1	0	0	$I_4 = 0$
0	1	0	1	$I_5 = 1$
0	1	1	0	$I_6 = 1$
0	1	1	1	$I_7 = 0$
1	0	0	0	$I_8 = 1$
1	0	0	1	$I_9 = 0$
1	0	1	0	$I_{10} = 0$
1	0	1	1	$I_{11} = 1$
1	1	0	0	$I_{12} = 1$

प्रश्न 17

Page:
Date: / /

Gray कोड → यह कोड binary number system का एक non-weighted कोड है जिसमें दो लगातार मान (दो) केवल एक बिट में भिन्न होते हैं।
It is not suitable for ~~mat~~ arithmetic operation

बाइनरी कोड ग्रे कोड में परिवर्तन

Ex 1 - 001 ; 2 = 010

Ex = 001 2 = 011

इस प्रकार 1 से 2 तक बढ़ने के लिए दो के बजाय केवल एक बिट को बदलने की आवश्यकता होती है।

Electromechanical switch से नकली output को रोकने और Digital टेलिविजन television और Telemetry system जैसे डिजिटल संचार में त्रुटि सुधार की सुविधा

Demultiplexer :-

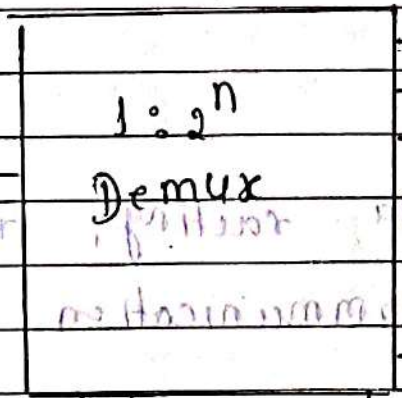
→ यह एक ऐसा logic circuit होता है जिसमें केवल एक इनपुट तथा एक से अधिक आउटपुट लाइन होती है।

→ कंट्रोल सिग्नल / select input को किसी एक output लाइन पर डाला जा सकता है।

→ A demultiplexer has perform the reverse operation of multiplexer.
that means it receives one input and distributes it over several outputs.
multiple

Hence demultiplexer is called as Data Distribution.

→ At a time only one output line is selected by select lines and the input is transmitted to the selected output line.



Block Diagram of 1 : 2^n demux

Page:
Date: / /

1) single input, multiple outputs

Demux में एक input line से data को multiple output lines पर distribute करता है।

इसमें एक input होता है, लेकिन कई output होते हैं।

2) Selection lines :

→ Demux के पास एक या multiple selection line होती है।

→ इन lines के through हम decide करते हैं कि data कौन-सी specific output line पर जायेगा।

3) Binary Decoder Configuration :

→ एक common use case में, demux को binary decoder के रूप में इस्तेमाल किया जाता है। Selection lines binary format में होते हैं, और हर combination के लिए एक specific output line select करते हैं।

4) Application :

इनका इस्तेमाल : data routing, address decoding and digital communication systems में किया जाता है।

Types of Demultiplexers

1) 1 : 2 demultiplexer
2) 1 : 4 demux
3) 1 : 8 demux

1) 1 : 2 demultiplexer

1 : 2 demux एक ऐसा digital circuit है जो एक input से 2 output lines पर data को distribute करता है।

Key point :

(i) Input & output :

Input line : एक ही input line होती है जिस demux से multiple O/P lines पर distribute किया जाता है।

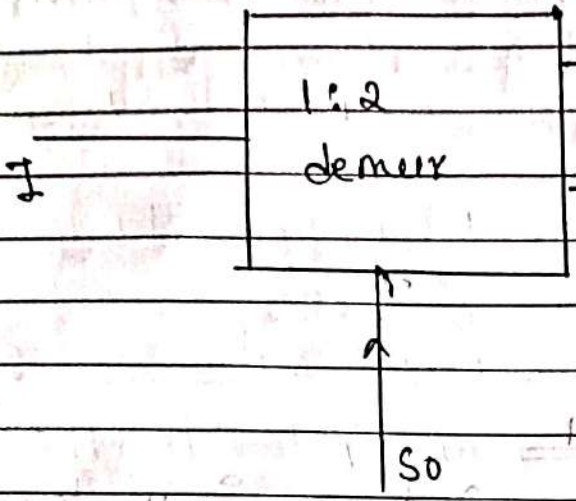
output line : दो output line होती हैं जिनमें से हर एक पर input data को एक बार भेजा जाता है।

(ii) Sebetion line : 1 : 2 Demux में, एक selection line होती है यह line decide करता है कि input data किस output line पर जायेगा।

(3) implementation : 1 : 2 Demux को AND और 2 OR gates का combination use करके implement किया जा सकता है।

Page:
Date: / /

Block diagram of 1:2 demux



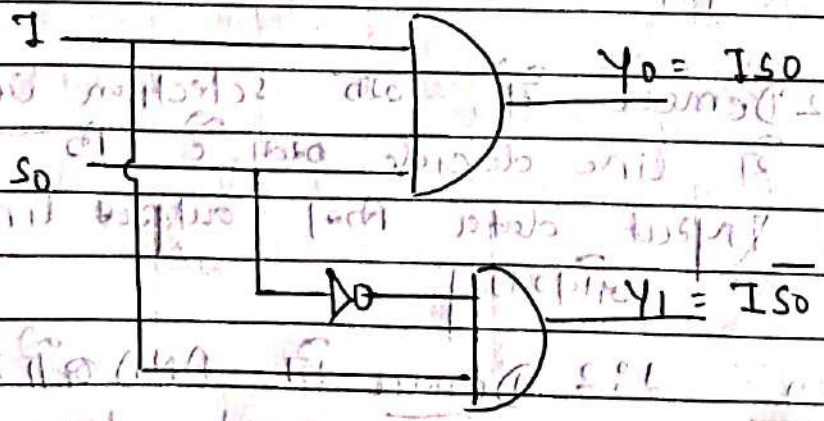
Logical Truth table of 1:2 demux

Input	output	
S_0	Y_1	Y_0
0	0	I
1	I	0

logical expression :

$Y_0 = I$ when $S_0 = 0$ $\Rightarrow Y_0 = \bar{I} S_0$

$Y_1 = I$ when $S_0 = 1$, $Y_0 = I S_0$

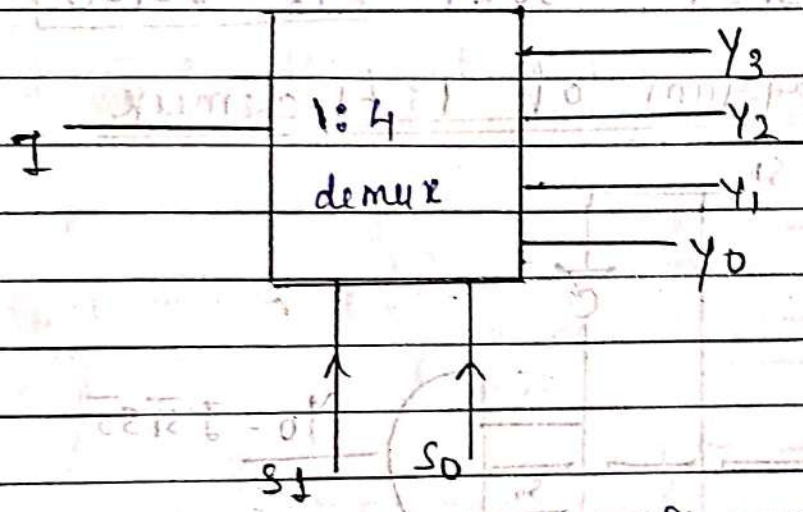


logical diagram of 1:2 demux

② 1 X 4 Demux

1:4 Demux एक digital circuit है जो एक input line से चार output lines पर data को distribute करता है।

Block diagram of 1:4 Demux



यहाँ input line एक है जिसे demux से 4 output lines पर distribute किया जाता है।

Truth table of 1:4 demux

Input		output			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

Logical expression

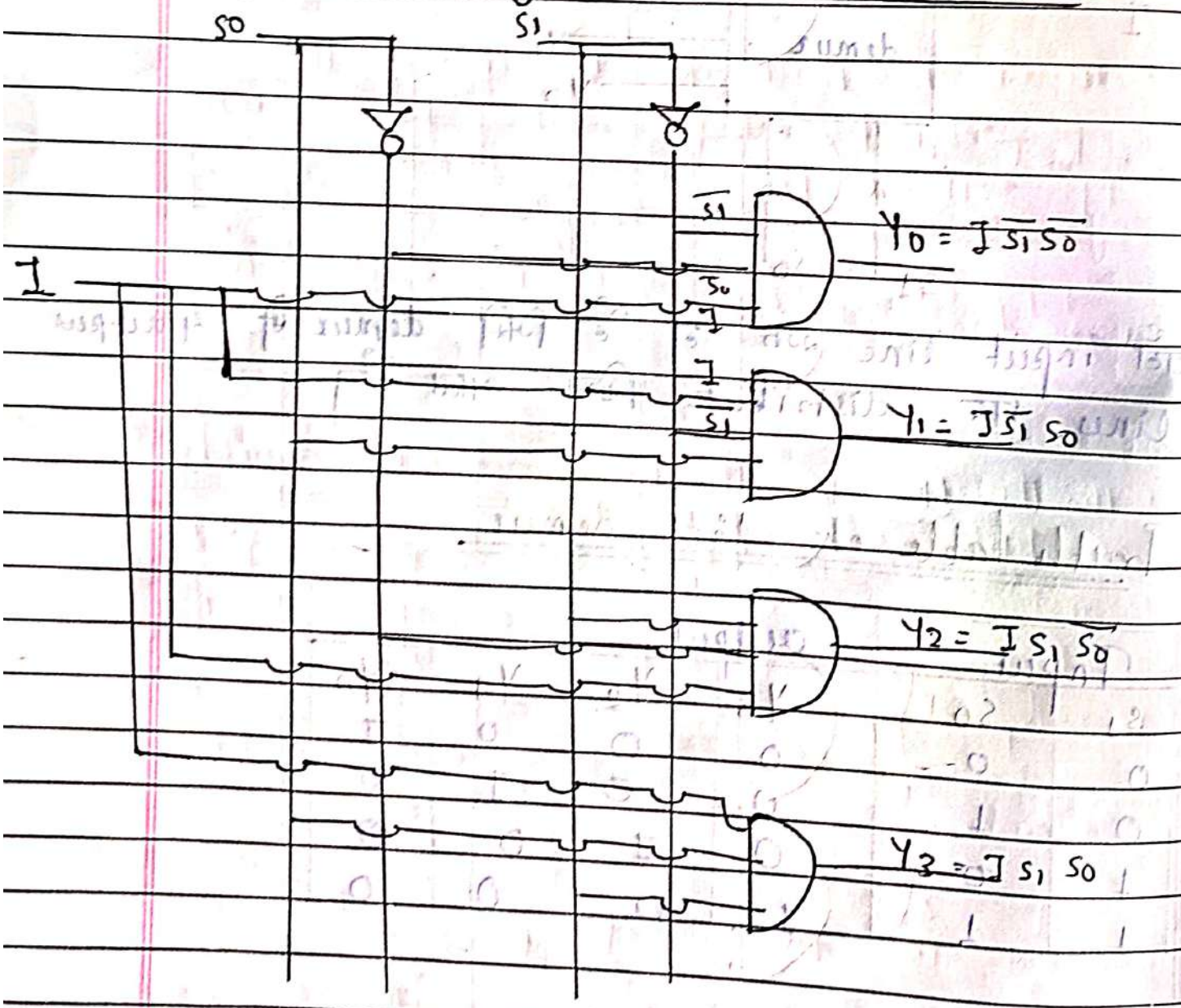
$Y_0 = I$ when $S_0 = 0$, $S_1 = 0$ $= \boxed{Y_0 = I\bar{S_0}\bar{S_1}}$

$Y_1 = I$ when $S_1 = 0$, $S_0 = 1$ $= \boxed{Y_1 = I\bar{S_1}S_0}$

$Y_2 = I$ when $S_1 = 1$, $S_0 = 0$ $\boxed{Y_2 = IS_1\bar{S_0}}$

$Y_3 = I$ when $S_1 = 1$, $S_0 = 1$ $\boxed{Y_3 = IS_1S_0}$

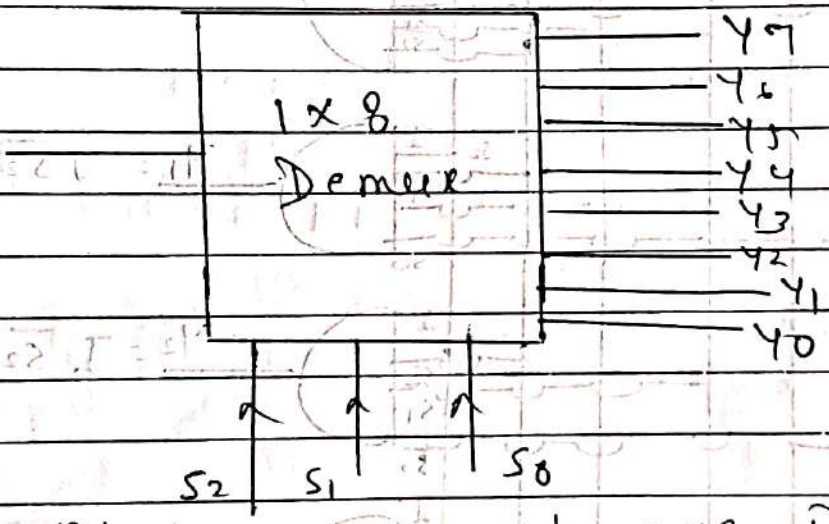
Logical Diagram of 1:4 demux



8) 1:8 Demux

→ 1:8 demux is a digital circuit.
जो एक input को लेकर उसे बाइनरी code के आधार पर 8 संभावित output में
8 possible outputs में direct करता है।

इसका मुख्य काम एक input को 8 अलग-अलग output में distribute करना होता है।



Block diagram of 1x8 Demux

Truth table

S_2	S_1	S_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0

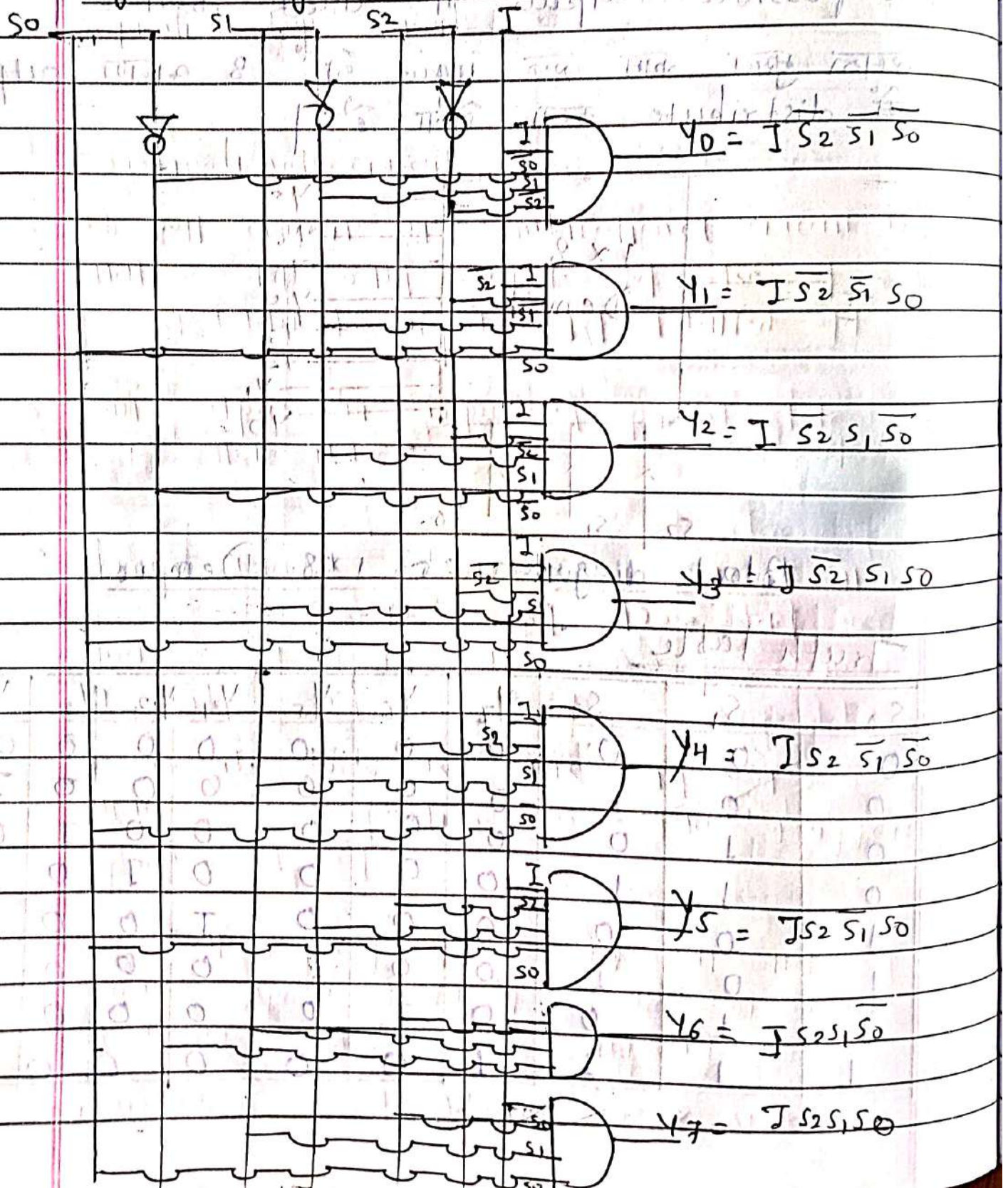
Page:
Date: / /

Logical expression

$Y_0 = I\bar{S_2}\bar{S_1}\bar{S_0}$ $Y_1 = I\bar{S_2}\bar{S_1}S_0$ $Y_2 = I\bar{S_2}S_1\bar{S_0}$ $Y_3 = I\bar{S_2}S_1S_0$

$Y_4 = IS_2\bar{S_1}\bar{S_0}$ $Y_5 = IS_2\bar{S_1}S_0$ $Y_6 = IS_2S_1\bar{S_0}$ $Y_7 = IS_2S_1S_0$

logical diagram of 1x8 Demux



CHAPTER : 4

Sequential Circuits

The digital systems in general are classified into two categories names :

① Combinational logic circuits

② Sequential logic circuits

① Combinational Circuits

→ एक ऐसा circuit को combinational circuit कहा जाता है जब उसका output पूरी तरह से उसके present (वर्तमान) input द्वारा निर्धारित होते है।

→ इसमें output को input की present state के द्वारा खोजा जाता है।

→ Storage Efficient : Combinational circuit जो है वह Data को store नहीं कर सकते है ये किसी भी memory element को contain नहीं करते है।

→ ये circuit clock पर निर्भर नहीं रहते है इसलिए इनके operations के लिए triggering महत्वपूर्ण नहीं है।

इसमें किसी memory का use नहीं किया जाता है। इसलिए input की पिछली स्थिति का circuit की वर्तमान स्थिति पर कोई प्रभाव नहीं पड़ता।

Ex → HALF Adder, Full Adder, Demux, Mux

(2) Sequential circuit

→ इसमें present input और past state output दोनों का उपयोग output की पहचान करने के लिए किया जाता है।

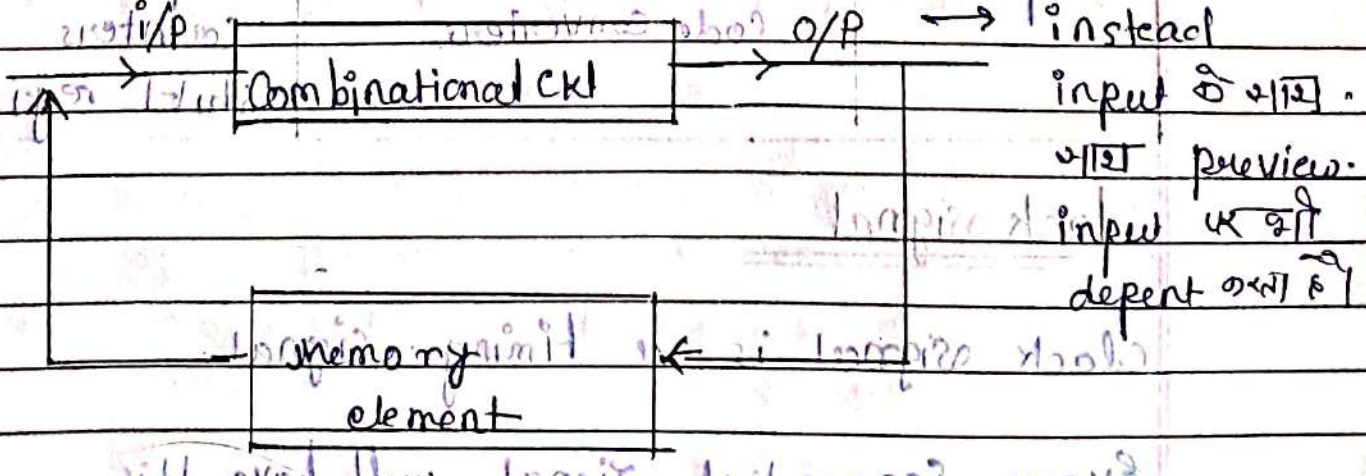
→ ये बहुत छोटी मात्रा में data को स्टोर कर सकते हैं, जिनमें memory element होते हैं जो उन्हें digital circuit में संगृहीत करते हैं।

→ इसमें triggering को करने के लिए clock का उपयोग किया जाता है।

→ ये sequential circuit का output वर्तमान समय के input, पिछले output और उस sequence पर निर्भर करता है जिसमें input लागू होते हैं।

→ पिछला input या output प्रदान करने के लिए एक memory तत्व का उपयोग करना आवश्यक है। जब प्रकार sequently circuit को एक memory तत्व की आवश्यकता होती है।

→ output को memory में store करता है।



→ instead input के साथ. साथ previous input पर भी depent करता है।

Block diagram of Sequential circuit

Page:
Date: / /

① Synchronous : एक साथ काम / एकसाथ काम

Ex : Flip-flop एक synchronization है। उसमें edge triggering

② Asynchronization Circuit : अलग-अलग time पर काम

Ex : Latch उसमें level triggering का use होता है

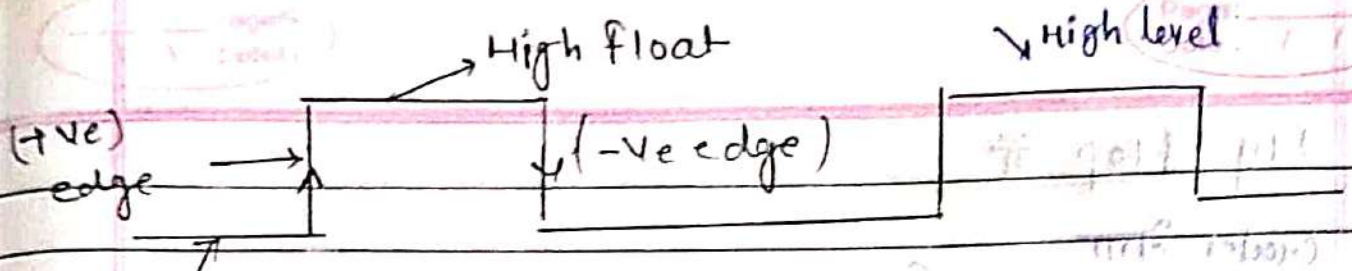
Comparision of Combinational and Sequential circuit.

Parameter	Combinational circuit	Sequential circuit
output depend on	Inputs present at that instant of time	Present inputs and past i/p / o/p
Memory	Not necessary	Necessary
Clock i/p	Not nessary	Necessary
Example	Adder, Subtractor Code converters	Flip-flops, Counters Shift registers

Clock signal

Clock signal is a timing signal.

Every sequential signal will have this timing signal



Clock signal

Flip-Flop :

Flip-flop is the basically the bistable elements

Flip-Flop is a sequential circuit which generally samples its inputs and changes its outputs only at paticular instants of time and not continously.

यह sequential circuit है जो memory cell के रूप में काम करता है यह bistable multivibrator के रूप में भी जाना जाता है

Flip-Flop दो स्थितियों में रह सकता है

(1) SET और (2) RESET

ये digital circuits में binary information को store करने और transfer करने में उपयोग होता है।

Flip flop एक clock signal और एक या दो input signals से संचालित करता है, जिनसे उसकी स्थिति बदलती है

Flip Flop का प्रमुख उपयोग memory units + registers और sequential logic circuits में होता है

Page:
Date: / /

Flip flop में

→ clocks होगा
→ clock signal भी होगा। और

There are two types of edge (level) triggered flip-flop:

(1) positive edge triggered (2) Negative edge triggered

Flip Flop के types:

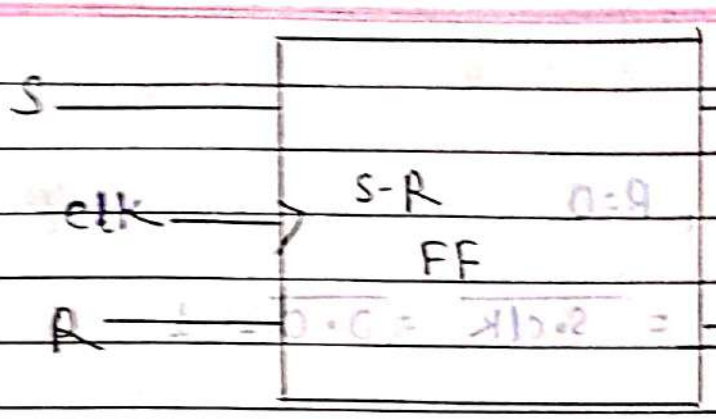
(1) S-R FF
(2) D-FF
(3) T-FF
(4) JK-FF

(1) S-R Flip-flop : (Set - Reset flip-flop)

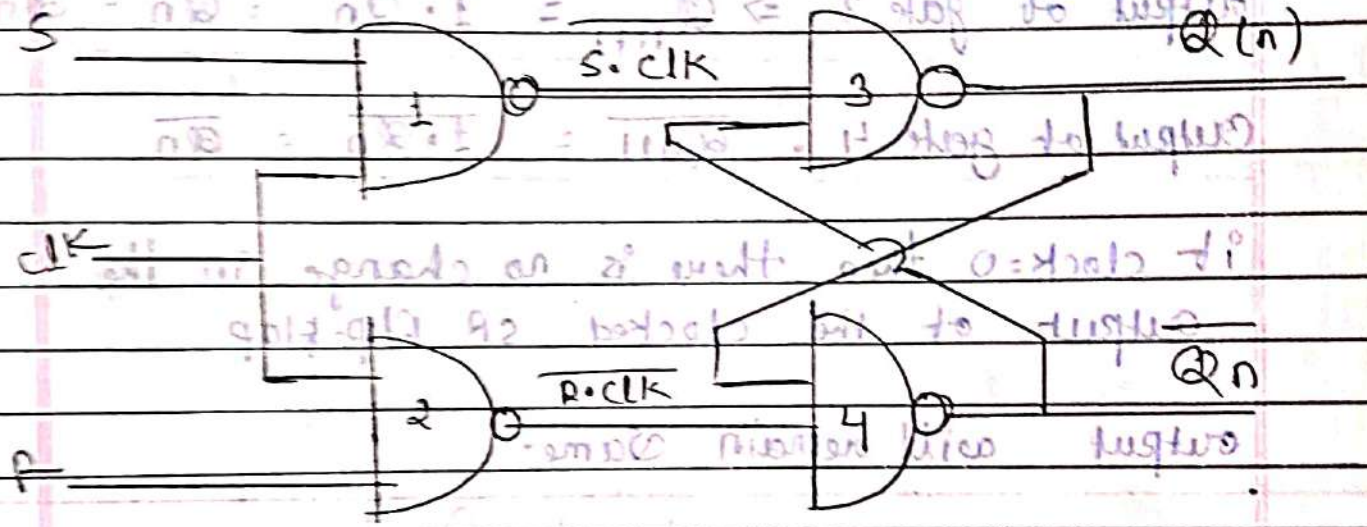
S-R FF एक प्रकार का astable multivibrator है जिसका उपयोग binary information storage में होता है। इसमें 2 input होते हैं SET और RESET

ये Flip-Flop दो स्थितियों में होता है SET और RESET स्थिति

SR Flip-Flop के अन्तर्गत sequential logic circuit और memory cells में होता है। ये circuits designing में उपयोगी होता है जब आवश्यक होती है कि किसी particular स्थिति में flip flop को set करना है या use reset करना है।



Block diagram of S-R FF



Logical Diagram of clocked S-R Flip-Flop

इसमें तीन input S, R, clk है और दो output Q(n) और $\overline{Q}$n है।

previous output next stage के लिए input का काम करेगा।

Q(n) present stage का output

next stage Q(n+1) input के जैसे काम करेगा

★ clk = 0 होगा तो inactive होगा। clock

Case : 1

S=0 , clk=0 , R=0

output of gate 1 = $\overline{S \cdot clk}$ = $\overline{0 \cdot 0}$ = 1

output of gate 2 = $\overline{R \cdot clk}$ = $\overline{0 \cdot 0}$ = 1

output of gate 3 $\Rightarrow Q_{n+1} = \overline{1 \cdot Q_n} = \overline{\overline{Q_n}} = Q_n$

output of gate 4 : $\overline{Q_{n+1}} = \overline{1 \cdot Q_n} = \overline{Q_n}$

if clock=0, then there is no change in the output of the clocked SR flip-flop

output will remain same.

★ output S, R पर depent नही होता जब clock inactive है

Case : 2 S=0 , R=0 , clk=1

output gate 1 = $\overline{S \cdot clk}$ = $\overline{0 \cdot 1}$ = 1

output gate 2 = $\overline{R \cdot clk}$ = $\overline{0 \cdot 1}$ = 1

output gate 3 $Q_{n+1} = \overline{1 \cdot \overline{Q_n}} = \overline{\overline{Q_n}} = Q_n$

output gate 4 = $\overline{Q_{n+1}}$ = $\overline{1 \cdot Q_n}$ = $\overline{Q_n}$

→ output not change

Page
Date

Case 3 : S = 0, R = 1, clk = 1

output gate 1 : $\overline{S \cdot clk} = \overline{0 \cdot 1} = \overline{0} = 1$ Q_n

Output gate 2 : $\overline{R \cdot clk} = \overline{1 \cdot 1} = \overline{1} = 0$ $\overline{Q_n}$

Output gate 3 $Q_{n+1} = \overline{0 \cdot \overline{Q_n} \cdot 1 \cdot 1} = 0$

Output gate 4 $\overline{Q_{n+1}} = \overline{0 \cdot Q_n} = \overline{0} = 1$

output will be Reset stage.

Case 4 : S = 1, R = 0, clk = 1

output gate 1 : $\overline{S \cdot clk} = \overline{1 \cdot 1} = 0$ Q_n

output gate 2 : $\overline{R \cdot clk} = \overline{0 \cdot 1} = 1$ $\overline{Q_n}$

output gate 3 $Q_{n+1} = \overline{Q \cdot \overline{Q_n}} = 1$ [0 है तो input तो output 1]

output gate 4 $\overline{Q_{n+1}} = \overline{1 \cdot 1} = 0$

output will be SET stage.

Case -5 : $S_1 = 1$ hold R = 1, clk = 1

output gate 1 $\overline{S \cdot clk} = \overline{1 \cdot 1} = 0$

output gate 2 $\overline{R \cdot clk} = \overline{1 \cdot 1} = 0$

Output gate 3 $Q_{n+1} = \overline{0 \cdot \overline{Q_n}} = 1$ [0 है तो NAND gate O/P 1 होगा]

output gate 4 $\overline{Q_{n+1}} = \overline{0 \cdot Q_n} = 1$

This can not be possible Invalid stage.

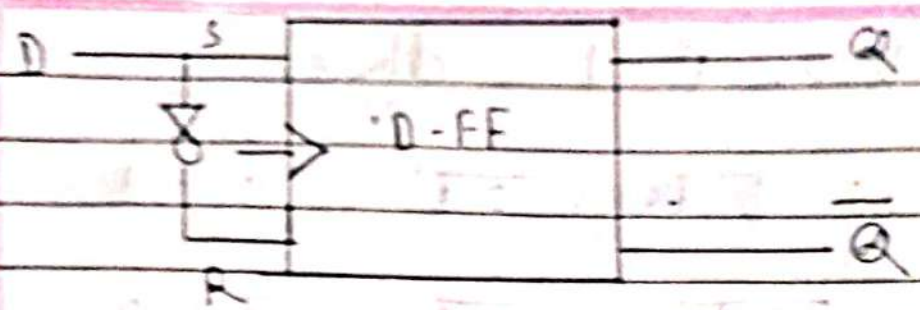
Truth table / characteristic table

clk	S	R	Qn	Qn+1	stage
0	X	X	0	0	] No change
0	X	X	1	1	
↑	0	0	0	0	] No change
↑	0	0	1	1	
↑	0	1	0	0	] RESET stage
↑	0	1	1	0	
↑	1	0	0	1	] SET stage
↑	1	0	1	1	
↑	1	1	0	X	] invalid stage
↑	1	1	1	X	

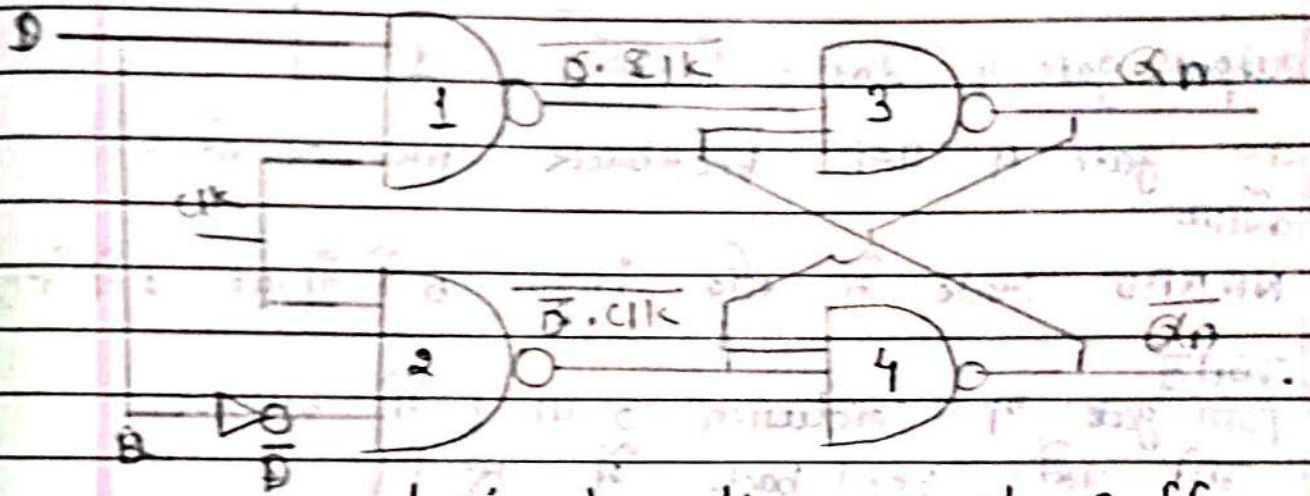
D - FF - Data FF

R-S flip-flop में बिट 1 को store करने के लिए S input high रखा जाता है तथा बिट 0 को store करने के लिए R input high रखा जाता है

इसके अतिरिक्त R तथा S दोनों input कभी भी एक साथ high के सकते है जो R-S flipflop में permitted नहीं है इन दोनों दोषों को दूर करने के लिए D flip flop use करते है।



Block diagram of D-FF



Logical diagram of D-FF

Case 1 : $D=0$, $\overline{D}=1$, $clk=0$

output gate 1 : $\overline{D \cdot clk} = \overline{0 \cdot 0} = \overline{0} = 1$

output gate 2 : $\overline{D \cdot clk} = \overline{1 \cdot 0} = \overline{0} = 1$

output gate 3 : $Q_{n+1} = \overline{1 \cdot \overline{Q_n}} = Q_n$

output gate 4 : $\overline{Q_{n+1}} = \overline{1 \cdot Q_n} = \overline{Q_n}$

Output 3 & 4 remains same
No change state.

Case 2 :- D = 0, $\overline{D}$ = 1 clk = 1

output gate 1 : $\overline{D \cdot clk} = \overline{0 \cdot 1} = \overline{0} = 1$

output gate 2 : $\overline{D \, clk} = \overline{1 \cdot 1} = \overline{1} = 0$

output gate 3 : $Q_{n+1} = \overline{1 \cdot 1} = \overline{1} = 0$

output gate 4 : $\overline{Q_{n+1}} = \overline{0 \cdot Q_n} = 1$

* यहाँ gate 4 पहले feedback provide करेगा क्योंकि NAND Gate में यदि input 0 है तो out 1 ही आएगा अर्थात जिस gate का conclusion 0 या 1 में आएगा वही पहले feed back में जाएगा।

∴ output 3 & 4 में will be reset stage.

Case 3 :- D = 1, $\overline{D}$ = 0, clk = 1

output gate 1 : $\overline{D \cdot clk} = \overline{1 \cdot 1} = \overline{1} = 0$

output gate 2 : $\overline{\overline{D} \cdot clk} = \overline{0 \cdot 1} = \overline{0} = 1$

output gate 3 : $Q_{n+1} = \overline{0 \cdot \overline{Q_n}} = 1$

output gate 4 : $\overline{Q_{n+1}} = \overline{1 \cdot 1} = \overline{1} = 0$

Output will be set stage.

Truth table :

Input		output		
clk	D	Qn	Qn+1	state
0	x	0	0	] No change
0	x	1	1	
↑	0	0	0	] Reset state
↑	0	1	0	
↑	1	0	1	] set state
↑	1	1	1	

D - FF का use S-R FF के drawback को over कम करने के लिए किया जाता है।

S-R FF में 4 state से चलने पर यदि A = 1, B = 1 तो output भी 1, 1 आता है।

★ इसमें S-R FF के invalid / state को over कम करने के लिए use किया जाता है।

Output output :

(1) No change : Qn जो है वही Qn+1 भी होगा।

(2) Reset state : Qn पर निर्भर नहीं करता। D पर करता है।

(3) Set state : Qn पर निर्भर नहीं करता है।

इसे data FF इसलिए कहते हैं कि जो input दे रहे हैं वही output प्राप्त होता है।

important

Page:
Date: / /

JK-FF

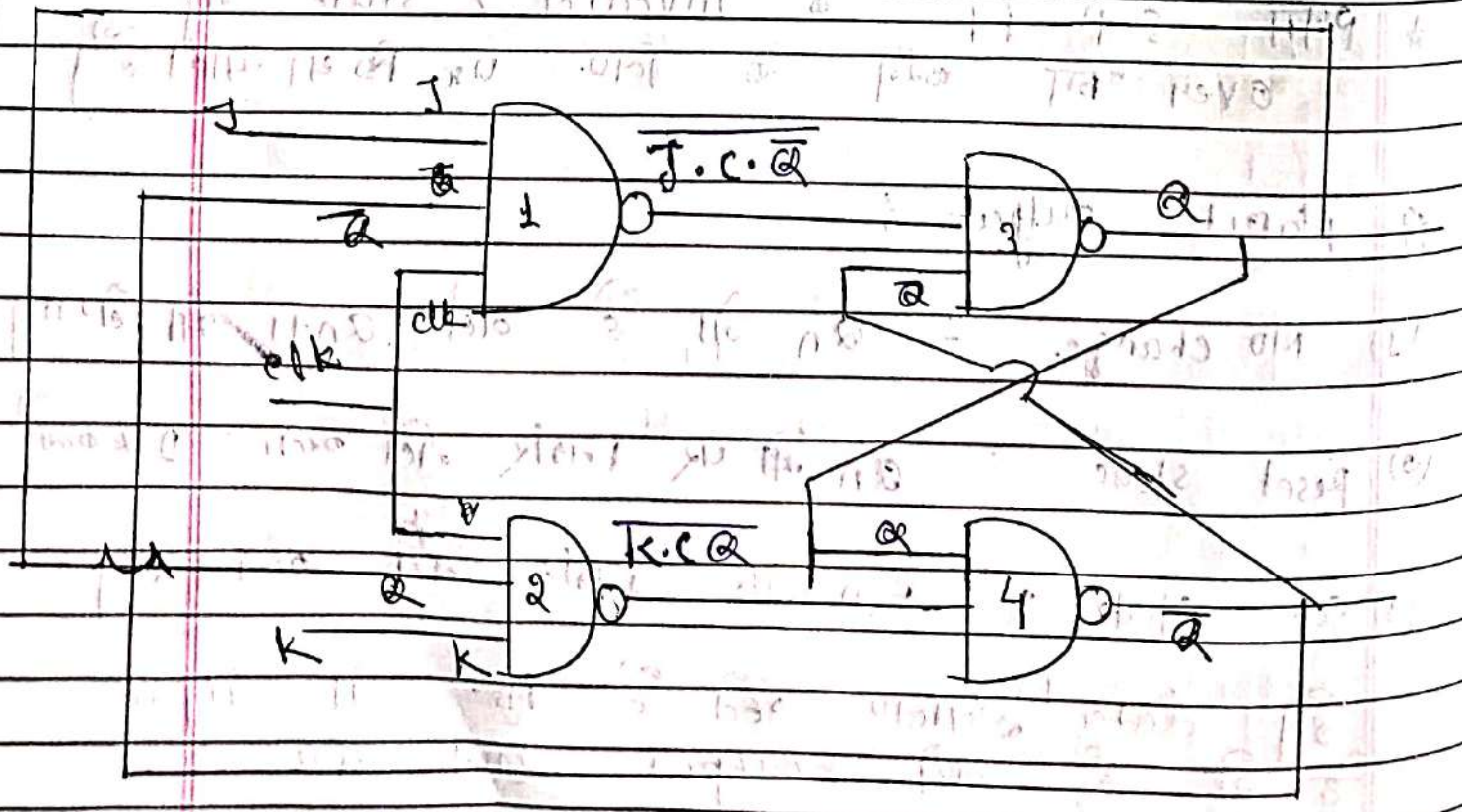
J.K-FF एक digital electronic circuit है जो कि binary information को store करने के लिए use होता है।

★ यह S-R FF के invalid state को Useful बनाता है।

★ D-FF में Invalid state को हटा कर S-R perform करता है।

JK Flip-flop, D Flip-flop के एक प्रकार का extension है, जिसमें J(set) और K(reset) के अलावा एक और संभव यानी toggle input होता है।

C = clk



logical diagram of JK-FF

JK Flip-Flop का उपयोग mainly sequential logic circuits, frequency division circuit, और memory units में होता है।

→ Toggle input से आप Flip-flop के स्थिति को toggle (बदल) सकते है।
जिससे Versatile Sequential Logic design किया जा सकता है।

Case ÷ I

clk = 0, J = X (Logic की है), K = X

gate 1 output $\overline{J \cdot C \cdot \bar{Q}} = \overline{X \cdot 0 \cdot \bar{Q}} = 1$

gate 2 output $\overline{KCQ} = \overline{X \cdot 0 \cdot Q} = 1$

gate 3 output $Q_{n+1} = \overline{J \cdot C \cdot \bar{Q} \cdot \bar{Q}} = \overline{1 \cdot \bar{Q}} = Q$

gate 4 output $\overline{Q_{n+1}} = \overline{J \cdot C \cdot Q \cdot Q} = \overline{1 \cdot Q} = \bar{Q}$

output remains say no change state

Case ÷ II clk = 1, J = 0, K = 0

gate 1 output $\overline{J \cdot C \cdot \bar{Q}} = \overline{0 \cdot 1 \cdot \bar{Q}} = 1$

gate 2 output $\overline{KCQ} = \overline{0 \cdot 1 \cdot Q} = 1$

gate 3 Output $Q_{n+1} = \overline{1 \cdot \bar{Q}} = Q$

gate 4 output $\overline{Q_{n+1}} = \overline{1 \cdot Q} = \bar{Q}$

In Case II output will be no change.

Case III

$clk = 1, J = 0, k = 1$

gate 1 output $= \overline{JC\overline{Q}} = \overline{0 \cdot 1 \cdot \overline{Q}} = 1$

gate 2 output $= \overline{KCQ} = \overline{1 \cdot 1 \cdot Q} = \overline{Q}$

gate 3 output $= Q_{n+1} = \overline{\overline{JC\overline{Q}} \cdot \overline{Q}} = \overline{1 \cdot 1} = 0$

gate 4 output $= \overline{Q_{n+1}} = \overline{\overline{KCQ} \cdot Q} = \overline{\overline{Q} \cdot Q} = \overline{0} = 1$

output will be Reset state.

इसमें gate no. 4 से पहले feedback लेंगे क्योंकि जिस gate का conclusion 0 या 1 हो।

gate 4 में $\overline{KCQ} = \overline{Q}$ है और दूसरा input Q है तो $\overline{\overline{Q}Q} = 1$, feedback को दूसरे के लिए input की तरह लिखेंगे।

Case III : $Q_{n+1} = 0$, $\overline{Q_{n+1}} = 1$
इसलिए reset state है।

Case : 4

$clk = 1, J = 1, k = 0$

gate 1 output $= \overline{J \cdot C\overline{Q}} = \overline{1 \cdot 1 \cdot \overline{Q}} = Q$

gate 2 output $= \overline{KCQ} = \overline{0 \cdot 1 \cdot Q} = 1$

Gate 3 output : $Q_{n+1} = \overline{J \cdot C \cdot \bar{Q}} \cdot \bar{Q} = \overline{Q \cdot \bar{Q}} = \bar{0} = 1$

Gate 4 output : $\overline{Q_{n+1}} = \overline{K \cdot C \cdot Q} \cdot Q = \overline{1 \cdot 1} = 0$

$Q_{n+1} = 1 \quad , \quad \overline{Q_{n+1}} = 0$

हमें यहाँ input से 0 प्राप्त नहीं हो रहा है इसलिए gate 3 & 4 से ही जिसमें Conclusion पर 1 में आएगा वही पहले feedback में जाएगा।

Case : S

clk = 1 , J = 1 , K = 1 | 0

Case S में दो और case होंगे

→ माना $Q_n = 0$, $\overline{Q_n} = 1$

Gate 1 output : $\overline{J \cdot C \cdot \bar{Q}} = \overline{1 \cdot 1 \cdot \bar{Q}} = \bar{Q} = 0$

Gate 2 output : $\overline{K \cdot C \cdot Q} = \overline{1 \cdot 1 \cdot Q} = \bar{Q} = 1$

Gate 3 output $Q_{n+1} = \overline{J \cdot C \cdot \bar{Q}_n \cdot Q_n} = \overline{0 \cdot 0} = 1$ Q_n

Gate 4 output $\overline{Q_{n+1}} = \overline{K \cdot C \cdot Q_n \cdot Q_n} = \overline{1 \cdot 1} = 0$ $\overline{Q_n}$

अर्थात् जब

$Q_n = 0 \Rightarrow Q_{n+1} = 1$

$\overline{Q_n} = 1 \Rightarrow \overline{Q_{n+1}} = 0$

Page:
Date: / /

Case 5 $\frac{2^0}{1}$ Case 2

clk = ↓, J = 1, K = 1

माना $Q_n = 0$, $\overline{Q_n} = 1$

gate 1 o/p $\overline{Jc \cdot Q} = \overline{1 \cdot \overline{Q}} = \overline{\overline{Q}} = Q = 1$

gate 2 o/p $\overline{Kc\overline{Q}} = \overline{1 \cdot 1 \cdot Q} = \overline{Q} = 0$

gate 3 o/p $\overline{Jc\overline{Q} \cdot \overline{Q}} = Q_{n+1} = \overline{1 \cdot 1} = 0$

gate 4 o/p $\overline{K \cdot cQ \cdot Q} = \overline{Q_{n+1}} \Rightarrow \overline{0 \cdot Q} = 1$

जब $Q_n = 0$ Then $Q_{n+1} = 0$

$\overline{Q_n} = 1$ then $\overline{Q_{n+1}} = 1$

Truth table :

clk	J	K	Q_n	Q_{n+1}	state
0	x	x	0	0	No change
0	x	x	1	1	
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	set
1	1	0	1	1	
1	1	1	0	1	toggle
1	1	1	1	0	

→ Counter बनाने के लिये J K flip flop की आवश्यकता होती है।

→ R-S flip-flop में Invalid अवस्था (R=S=1) को दूर करने के लिए उसमें दो AND gate लगाकर उसे संशोधित करते हुए एक edge triggered J-K flip-flop बनाया जाता है।

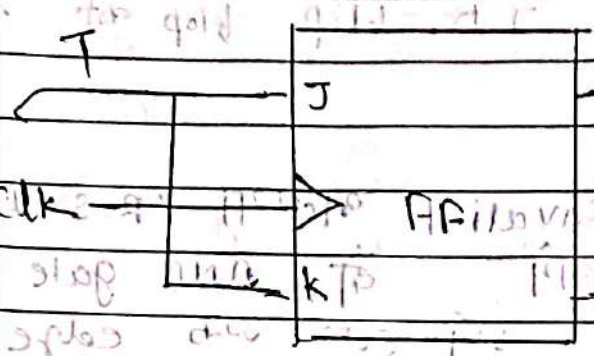
→ यहाँ J तथा K दो control inputs है। clock को एक R-C परिपथ से जोड़ा जाता है। clock के धनात्मक edge पर flip-flop के output में परिवर्तन होता है।

T - Flip - Flop

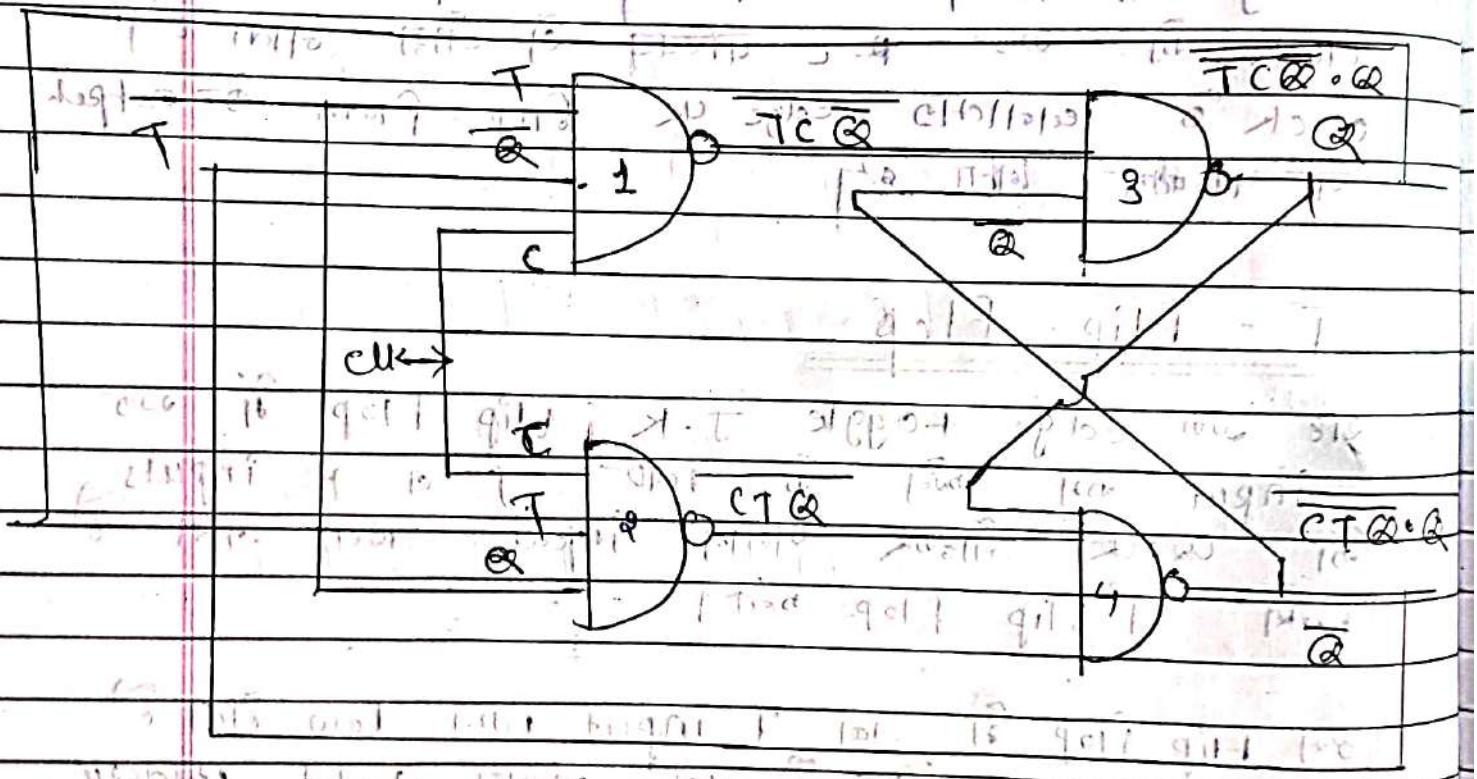
→ यह एक edge toggle J-K flip flop में एक input कम करने के लिए J व K inputs को परस्पर जोड़कर समान input दिया जाता है जिसे T flip flop कहते।

इस flip flop में जब T input निम्न low होता है अर्थात J=K=0 होती है तब परिपथ hold अवस्था में रहता है और output में कोई change नहीं होता है।

जब T input high होता अर्थात J=K=1 होता है तब clock + के धनात्मक edge पर flip-flop toggle करता है और पूर्ण output complement प्राप्त होता है।



Block diagram of T-FF



logical diagram of T-FF

Case 1 :

C = 0, T = X

(1) Gate 1 output $\overline{TC\overline{Q}} = \overline{X \cdot 0 \cdot \overline{Q}} = 1$

(2) gate 2 output $\overline{CTQ} = \overline{0 \cdot T \cdot Q} = 1$

(3) gate 3 output: $Q_{n+1} = \overline{T C \bar{Q} \cdot \bar{Q}} = \overline{1 \cdot \bar{Q}} = Q$

(4) gate 4 output: $\overline{Q_{n+1}} = \overline{C T Q \cdot Q} = \overline{1 \cdot Q} = \bar{Q}$

$Q_{n+1} = Q$, $\overline{Q_{n+1}} = \bar{Q}$

Output will be NO change.

Case - II :

$C = 1$, $T = 0$

1) gate 1 output $\overline{T C \bar{Q}} = \overline{1 \cdot 0 \bar{Q}} = 1$

2) gate 2 output $\overline{C T Q} = \overline{1 \cdot 0 \cdot Q} = 1$

3) gate 3 output $Q_{n+1} = \overline{T C \bar{Q}_n \bar{Q}_n} = \overline{1 \cdot \bar{Q}_n} = Q_n$

gate 4 output $\overline{Q_{n+1}} = \overline{T C Q \cdot Q} = \overline{1 \cdot Q} = \bar{Q}_n$

$Q_{n+1} = Q$, $\overline{Q_{n+1}} = \bar{Q}$ Not change

Case III $C = 1$, $T = 1$

(1) gate 1 output $\overline{T C \bar{Q}} = \overline{1 \cdot 1 \cdot \bar{Q}}$

(2) gate 2 output $\overline{C T Q} = \overline{1 \cdot 1 Q} = \bar{Q}$

(3) gate 3 output $Q_{n+1} = \overline{T C \bar{Q} \bar{Q}} = \overline{Q \cdot \bar{Q}} = \bar{0} = 1$

(4) gate 4 output $\overline{Q_{n+1}} = \overline{T C Q \cdot Q} = \overline{\bar{Q} \cdot Q} = \bar{0} = 1$

Page:
Date: / /

Case :- III C|k = 1 , T = 1

$Q_n = 0$ $\overline{Q_n} = 1$

1) gate 1 output $\overline{TC\bar{Q}} = \overline{1 \cdot 1 \cdot \bar{Q}} = Q = 0$

2) gate 2 output $\overline{CTQ} = \overline{1 \cdot 1 \cdot Q} = \bar{Q} = 1$

3) gate 3 output $Q_{n+1} = \overline{TC\bar{Q} \cdot \bar{Q}_n} = \overline{0 \cdot \bar{Q}_n} = 1 = Q_{n+1}$

4) gate 4 output $\overline{Q_{n+1}} = \overline{CTQ_n \cdot Q_n} = \overline{1 \cdot Q_n} = \overline{1 \cdot 1} = 0$

उसमें, gate 3 से पहले feedback लेंगे क्योंकि उसका output 1 है।

$\boxed{Q_{n+1} = 1}$ $\boxed{\overline{Q_{n+1}} = 0}$

Ex. $Q_n = 1$ $\overline{Q_n} = 0$

1) gate 1 output $\overline{TC\bar{Q}} = \overline{1 \cdot 1 \cdot \bar{Q}} = Q = 1$

2) gate 2 output $\overline{CTQ} = \overline{1 \cdot 1 \cdot Q} = \bar{Q} = 0$

3) gate 3 output $Q_{n+1} = \overline{TC\bar{Q}\bar{Q}} = \overline{0 \cdot Q_n} = \overline{1 \cdot 1} = 0$

4) gate 4 output $Q_{n+1} = \overline{CTQ_n \cdot Q_n} = \overline{0 \cdot Q_n} = 1$

$\because Q_n = 1$ then $Q_{n+1} = 0$

$\because \overline{Q_n} = 0$ then $Q_{n+1} = 1$

Truth table of T Flip flop

Input		output		
clk	T	Qn	Qn+1	state
0	x	0	0	no change
0	x	1	1	
1	0	0	0	no change
1	0	1	1	
1	1	0	1	toggle
1	1	1	0	

Page:
Date: / /

Excitation table : यह truth table से ही बनता है।

Excitation table is used for desinging or sequential circuit or flip flop design करना।

It Indicate the input required to be a p at flip flop to desier output.

S-R FF :

Q_n → present state

Q_{n+1} → Next state

Ps	NS	Required i/p	
Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Excitation table :

Excitation table digital circuits में use होता है और ये बताता है कि give input condition पर किस state में flip-flop को excite किया जाए। इसमें flip-flop के current state और inputs के basis पर next state को determine करने के लिए ...

information होती है।

ये table design और analysis में मदद करता है।

→ हमने अभी तक Flip-Flops के Truth table लिखी है जिसे characteristic tables भी कहते है।

लेकिन sequential circuit के design करते समय, कभी-कभी circuit की वर्तमान और next स्थिति दी जाती है और हमें संबंधित input स्थिति खोजने की उम्मीद की जाती है।

Present state & clock pulse लगाने का पहले कि स्थिति है और Next state clock pulse लगने के बाद की स्थिति है।

S-R FF का excitation table: इसमें हम invalid और inactive को छोड़ते है।

Truth table of SR Flip-Flop

clk	S	R	Qn	Qn+1
↑	0	0	0	0
↑	0	0	1	1
↑	0	1	0	0
↑	0	1	1	0
↑	1	0	0	1
↑	1	0	1	1

excitation table S-R FF के truth table से बनेगा जैसे

1) → Qn = 0 , Qn+1 = 0 दो condition पर है

S	R
0	0
0	1

यानी S = 0 और R x (कुछ भी हो) output 0 → 0 होगा

Present state	Next state	Required i/p	
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(2) $Q_n = 0$, $Q_{n+1} = 1$ होने के लिए truth table में $S = 1$ और $R = 0$ है।

(3) $Q_n = 1$ $Q_{n+1} = 0$ होने के लिए truth table में $S = 0$ और $R = 1$ है।

(4) $Q_n = 1$ $Q_{n+1} = 1$ होने के लिए truth table दो condition में है → S R → 0 0

इसमें $R = 0$ ही होगा और $S = X$ (कुछ भी हो)

Case II : Q should change from 0 to 1

This is nothing but set condition.
Hence $S_n = 1$, $R_n = 0$ should be the inputs

Case III Q should change from 1 to 0

This is nothing but set condition.
Hence $S = 0$, $R_n = 1$ should be the input

III Q should be 1 No. change

Condition 1: $S_n = S_R = 0$ No change in output
condition 2: $S_n = 1$ and $R = 0$

Hence these condition we conclude that S_n can be either 0 or 1 i.e. don't care and $R_n = 0$.
Hence we can the inputs corresponding to the case is $S_n = X$ and $R_n = 0$

(2) Excitation table of D flip flop

Truth table of D flip flop

clk	D	Q	Q_{n+1}
↑	0	0	0
↑	0	1	0
↑	1	0	1
↑	1	1	1

Excitation table of D flip flop

	Ps	Ns	Input
clk	Q	Q_{n+1}	D
	0	0	0
	0	1	1
	1	0	0
	1	1	1

Page:
Date: / /

Excitation table of JK flip flop:

~~Truth table~~

किसी J-K में क्लॉक पल्स आने के पूर्व तथा clock pulse आने के बाद प्राप्त output के संगत, आवश्यक Input मानों को दिखाने वाली table को excitation table कहते हैं।

किसी flip flop में निश्चित output प्राप्त करने के लिए J व K input के मान ज्ञात करने के लिए truth table की आवश्यकता होती है

Truth table of J-K flip flop

ans.

clk	J	k	Q_n	Q_{n+1}
0	x	x	0	0
0	x	x	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Excitation table of JK flip flop

Present state	Next state	Input	
Q	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Example

→ माना $Q = 0$ और $Q_{n+1} = 0$ है तो इसे प्राप्त करने के लिए के J और K के दो मान हो सकते हैं

J K
Condition ← 0 0 अर्थात J = 0 और K = X
0 1

→ जब $Q = 0$ और $Q_{n+1} = 1$ प्राप्त करना है तो इस स्थिति में किसी या तो सेट करना होगा या लॉजिक | J और K inputs को क्रमशः 1, 0 और 1, 1 होने चाहिये इन दोनों स्थितियों में J का मान 1 और K का मान 0 या 1 हो सकता है |

→ इस प्रकार जब $Q = 1$ और $Q_{n+1} = 0$ है तो प्राप्त करने के लिए J व K inputs की क्रमशः 0, 1 और 1, 1 होने चाहिए | इन दोनों स्थितियों में K का मान 1 और J का मान 0 या मान 1 हो सकता है अर्थात K निश्चित व J का मान परिवर्तित हो |

इस प्रकार जब $Q = 1$ और $Q_{n+1} = 1$ प्राप्त करने के लिए J व K की स्थिति 0, 0 या 1, 0 हो सकती जिसमें K = 0 और J = 0 या 1

(IV) · Excitation table of T flip-flop

present state	Next state	Input
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Truth table of T-FF

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

→ मान $Q_n = 0$ और $Q_{n+1} = 0$ प्राप्त करने के लिए T = 0 होगा

→ $Q_n = 0$ और $Q_{n+1} = 1$ प्राप्त करने के लिए T = 1 होगा।

→ $Q_n = 0$ और 1 प्राप्त करने के लिए T = 1 होगा

→ $Q_n = 1$, $Q_{n+1} = 1$ प्राप्त करने के लिए T = 0 होगा।

Page:
Date: / /

Register :

→ ये Data को store करता है।

∵ Flip-flop 1 bit के Data को store करता है।
अतः Register 4 bit Data को store करता है।

a number of flip-flop connected together such that data may be shifted into and shifted out of them is called shift Register

Digital प्रणालियों में Register एक महत्वपूर्ण घटक होता है। FF के वे समूह जो बाइनरी संख्याओं (0 व 1) को स्टोर करने के लिये प्रयुक्त किये जाते हैं, Register कहलाते हैं।

Register में बाइनरी संख्या के प्रत्येक bit को store करने के लिये अलग-अलग flip flop की आवश्यकता होती है।

ex → चार bit binary no. store करने के लिए 4 flip flop की Register का उपयोग निम्नलिखित रूपों में किया जाता है :-

(i) output Devices को transmit किये जाने वाले बाइनरी data को अणिक currently रूप से momentarily स्टोर करने के लिए

(ii) keyboard से input data प्राप्त कर microprocessor chip के input पर देने के लिये।

(iii) Computation, गुणन अथवा भाग आदि की प्रक्रियाओं के बीच प्राप्त आंशिक प्रतिफलों को store करने के लिए

Type of Registers

किसी भी Register में flip flop जिस प्रकार संयोजित किये जाते है कि बाइनरी संख्याये उनमें प्रवेश की जा सके और उनको बाहर भी shift की जा सके। Flip-flop के समूह जिनमें बाइनरी संख्यायो shift in अथवा shift out की जा सके shift registers कहलाते है।

1) (Serial in - Serial out) shift Register (SISO)

2) Serial in - parallel out shift Register (SIPO)

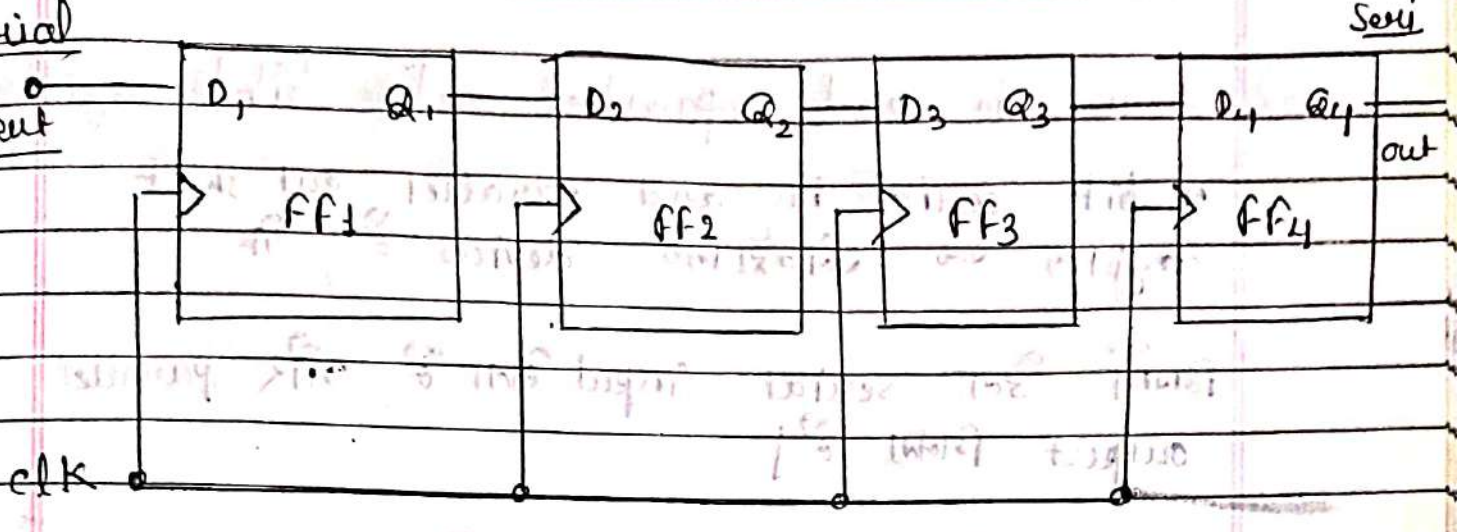
3) parallel in - serial out shift Register (PISO)

4) parallel in - parallel out shift Register (PIPO)

(1) Serial in - Serial out shift register

Data bits shift from left to right by 1 position per clock cycle.

* इसे shift left mode भी कहते है
इस register में data को in तथा out, Serial में होता है इसमें एक समय पर एक bit को left या right shift किया जाता है
इसमें यदि flip flop के output पहले flip flop के i/p में दे दी जाये तो यह एक Ring counter की तरह काम करता है



→ The logical diagram of 4-bit serial in and serial out shift register is given above

→ It has 4 b. flip-flop since it has 4 stages that is it can store upto 4 bits of data.

→ The serial data input is applied at the D input of first flip flop (FF1) (Q_1).

output of FF1 (Q_1) is connected to input of FF2 (D_2) and so on.

→ Data is outputted from Q output of (FF4) last stage.

→ When serial data is transferred into a register each new bit is clk the input FF-1.

→ The bit was previously stored in FF1 and transfered to FF2 then FF3 Hence FF4. sequentially

→ The bit stored in FF4 is shifted out.

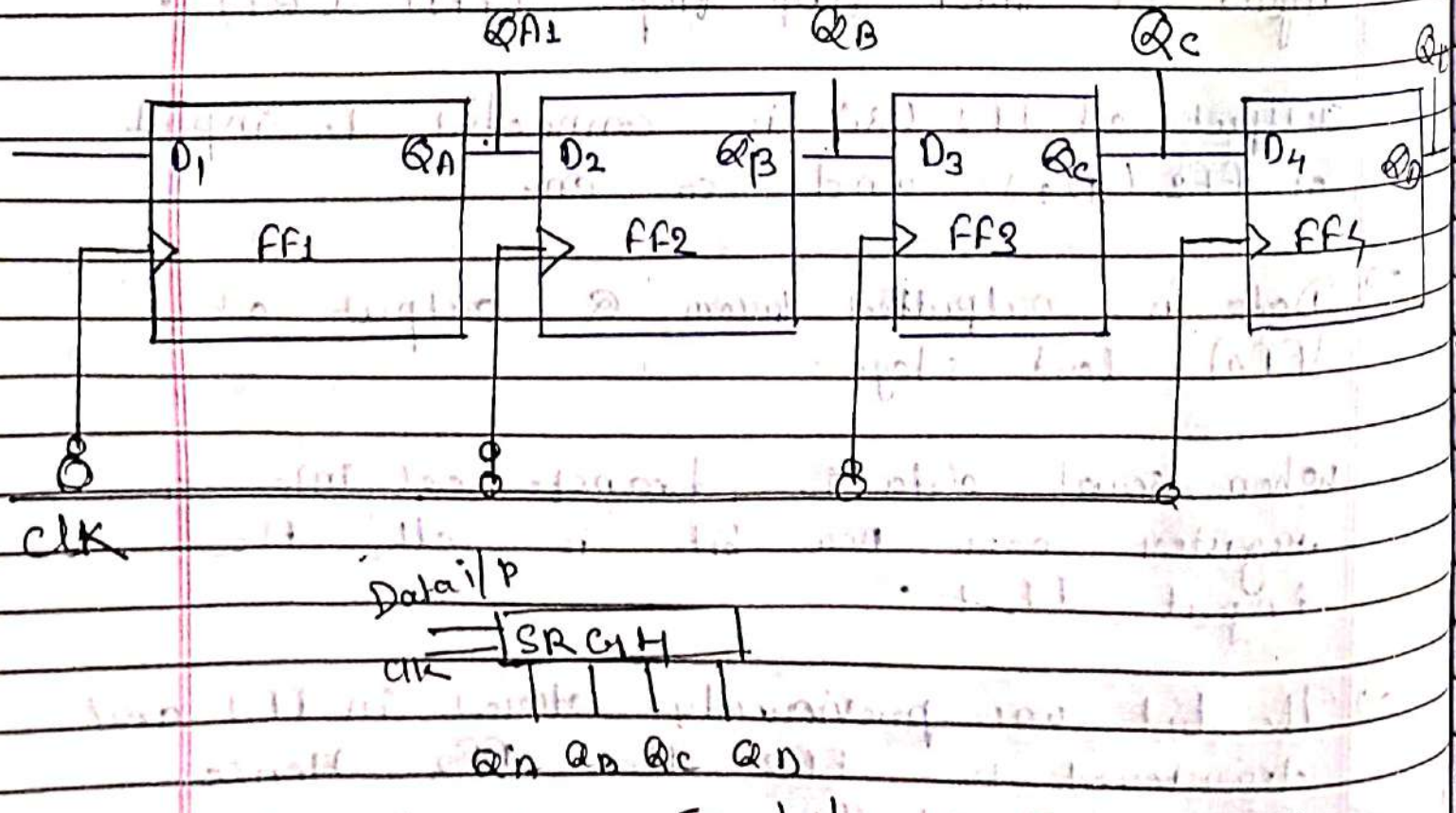
② Serial in and parallel out shift register

4 bit serial in and parallel out shift register एक इलेक्ट्रॉनिक devices है जो

जिसमें डेटा serial input होता है और parallel output मिलता है।

जब हम किसी इलेक्ट्रॉनिक device में 'serial in' कहते है तो यह तात्कालिक डेटा को bit-bit करके एक series में प्रवेश करने की जरूरत होती है। उसके बाद, जब यह डेटा एक स्थान से दूसरे स्थान पर जाता है, तो उसे पैरलल output में बाहर निकाला जाता है, जिसमें सभी bit एक साथ देखे जा सकते है।

★ यह तेजी से डेटा को transfer करता है।



Symbol

- In this operations the data is entered serially and taken out in parallel.
- That means first the data is loaded bit by bit. The outputs are disabled as long as the loading is taking place.
- As soon as the loading is complete and all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines simultaneously.
- Number of clock cycle required to load a four bit word is 4. Hence the speed of operation of SIPO mode is same as that of SISO mode.

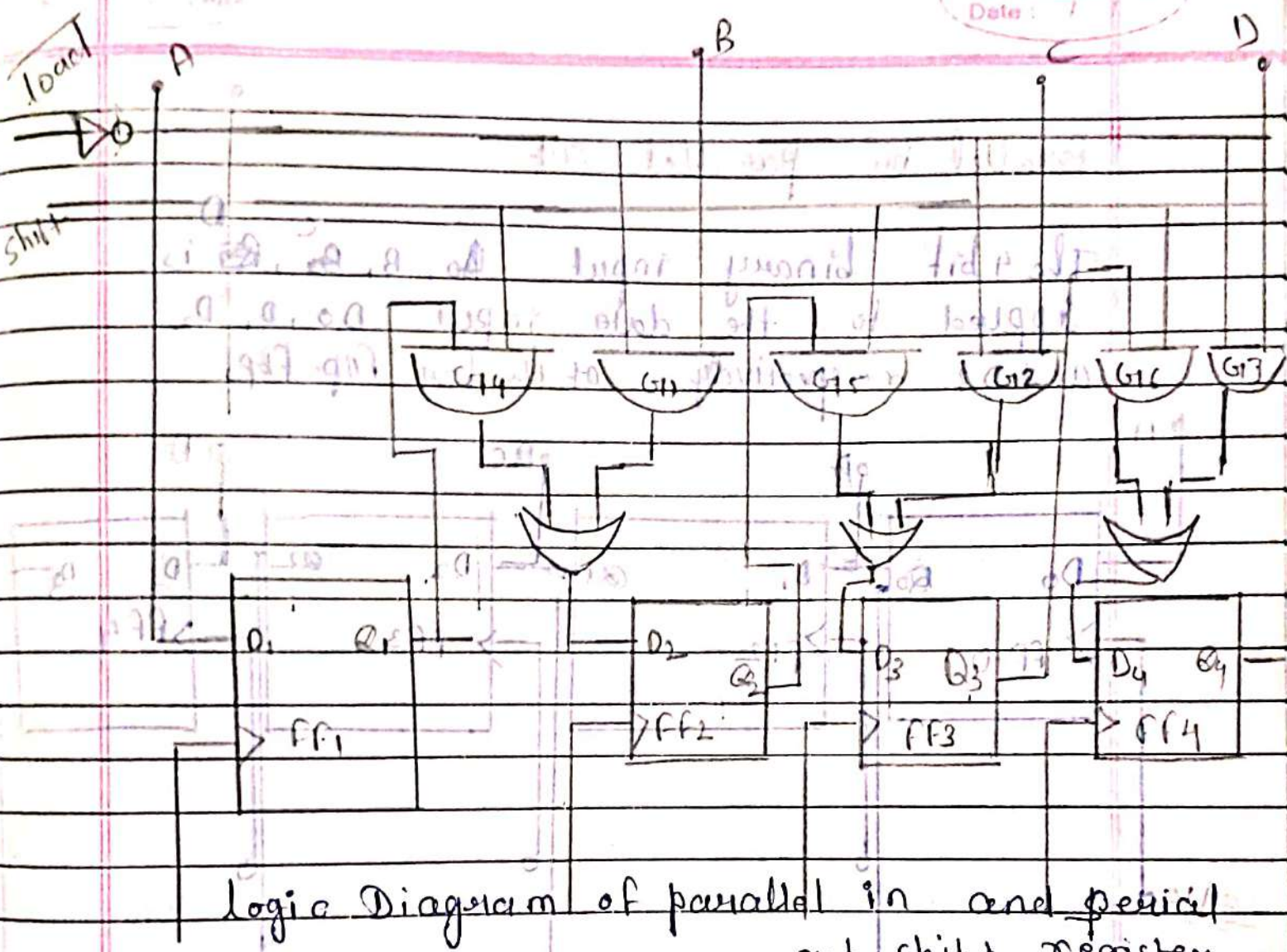
(3) Parallel In - Serial out register

- In this mode the bits are entered in parallel.
- The circuit is a four bit parallel input serial output register.
- Output of previous FF is connected to the input of the next one via a combination circuit.
- The binary input word B_0, B_1, B_2, B_3 is applied through the same combinational circuit.

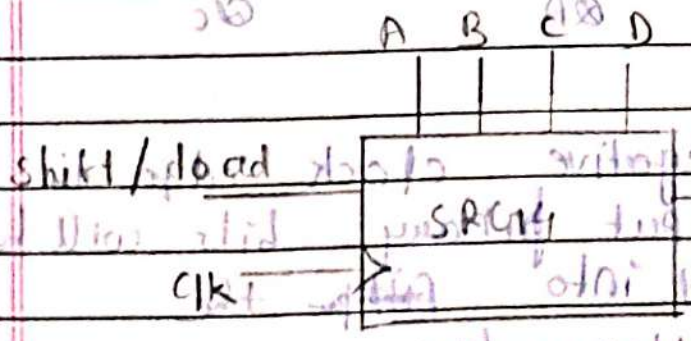
There are two modes in which this circuit can work namely shift mode or load mode.

- When the shift / load line is low (0), the AND gates 2, 4 and 6 become active. They will pass B1, B2, and B3 bits to the corresponding flip flops.
- On the low going edge of clock, the binary inputs B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading take place.

- When the shift / load line is high (1), the AND gates 2, 4, 6 become inactive. Hence the parallel of the data becomes impossible.
- But the AND gates 1, 3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses.
- Thus the parallel in serial out operation takes place.



Logic Diagram of parallel in and serial out shift register

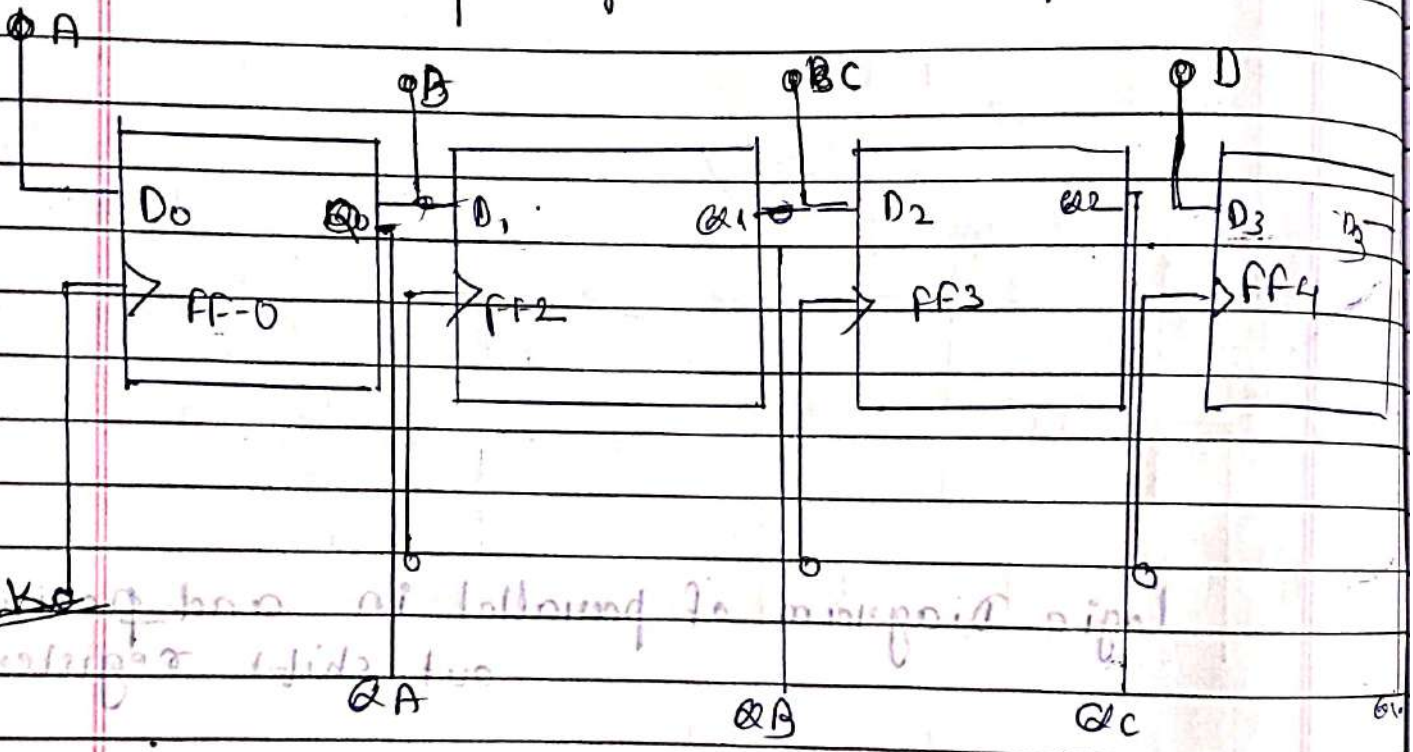


Logical Symbol

Page:
Date: / /

Parallel in- parallel out

The 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data input D_0, D_1, D_2 and D_3 respectively of the four flip-flops



As soon as a Negative clock edge is applied, the input binary bits will be loaded ~~bits~~ ~~with~~ into ~~flip~~ the flip-flops simultaneously.

- The loaded bit will appear simultaneously to the output size. only one clock pulse is essential to load all the bits.

(1) Serial-in & serial out shift Register

Serial in serial out shift register एक digital circuit है जो Data को sequentially stored और transfer करता है उसमें Data के लिए एक input line (serial in) और Data (serial out) के लिए एक output line है। Data को clock signal के basal पर register के माध्यम से bit by bit in या out transfer किया जाता है।

(2) Serial-in & parallel out shift register :

The working of a serial-in, parallel-out shift register involves the following steps :

(i) Serial input :

shift register एक input line के माध्यम से sequentially Data पाता है।

(ii) shift Operation :

प्रत्येक clock pulse आने वाले bit को register के माध्यम से shift करने का कारण बनता है जिससे ~~exist~~ existing bits एक position में shift हो जाते हैं।

Parallel out :

जब desired number में bits से shift किया जाता है तो Data output lines पर Parallel form में उपलब्ध होता है।

Shift register अपने Serial input के माध्यम से bit द्वारा Data store करता है, और प्रत्येक clock pulse के साथ, यह existing bits को shift करता है। एक बार desired Data load हो जाने पर उसे Parallel output lines से एक साथ read किया जा सकता है।

यह mechanism digital system में serial और Parallel devices के बीच interfacing के लिए उपयोगी है।

(3) Parallel in - serial out shift register.

→ इस mode में bits को Parallel में entered किया जाता है।

→ यह circuit 4 bit parallel input ; Serial output register है।

→ Parallel in serial out register एक ऐसा shift register है। जहाँ Data को Parallel में कई bits में load किया जाता है और फिर sequentially shifted किया जाता है।

→ Parallel in :

Data के सभी bits को register के FF में एक साथ load किया जाता है।

Serial out :

Data को clock signal का प्रयोग करके usually in a sequential manner एक time में एक bit shift किया जाता है।

extract desired Shifting occurs in a predefined order, सबसे important bit (MSB) से सबसे कम important Bit (LSB) तक।

exist existing इस प्रकार के shift register का use commonly इन scenarios में किया जाता है जहाँ parallel Data को serial shift किया जाना है।

Shifting एक predefine order में होता है

(4) Parallel in parallel out shift register :

Parallel in parallel out shift register एक digital circuit है जो Data को parallel loading और Data को parallel retrieval को allow करता है।

Page:
Date: / /

Parallel in :

सभी Data bits को shift register के FF में एक साथ load किया जाता है। प्रत्येक FF Data के एक bit से मेल खाता है।

Parallel out :

Data को सभी FF से एक साथ read किया जा सकता है। which means कि shift register का output किसी भी time frame में मौजूद parallel Data को दर्शाता है।

clock signal :

time signal का उपयोग renaming job को synchronize करने के लिए किया जाता है।

Flip - flop :

एक variable name जो simple building block जो कुछ Data store कर सकता है। यह या 1 state में हो सकता है।

→ shift register के design और specification के आधार पर, shifting left ward या rightward हो सकता है।

Counter :-

- The digital circuit used for counting pulse is known as counter.
- It is a sequential circuit.
- Counter is the widest application of flip-flop. It is a group of flip-flops with a clock signal pulse.
- Counter count the number of clock pulse Hence with some modifications it can be used for measuring frequency or time period
- Counter एक electronic circuit या device है जो input pulse या events को count करता है। ये digital circuits में इस्तेमाल होता है और commonly binary numbers में events को represent करने के लिए बनाया जाता है
- ~~एक~~ counter multiple flip-flops का group होता है जिसमें हर flip-flop अपने binary state को बदल कर एक count represent करता है।

Counter are basically of two type. In switching case

1) Asynchronous or ripple counter
2) synchronous ripple counter

Asynchronous counter :- ~~यह एक ऐसा~~ counter है जिसमें हर flip-flop अपने clock pulse को अपनी पुरानी ~~state~~ के flip-flop के output से प्राप्त करता है।

For these counters the external clock signal is applied to one flip-flop and then the output of preceding flip-flop is connected to the next flip-flop.

(ii) Synchronous counters :

In synchr counter एक ऐसा digital counter है जिसमें सभी flip-flop एक ही common clock pulse पर काम करते है, यानी हर flip-flop अपने state को change करने के लिए एक साथ clock pulse receive करते है।

In timing case counter is two type :

(1) up counter (2) down counter.

Full modulus counter :

इसे mod-N counter या N-modulus counter भी कहते जहाँ N modulus को represent करता है।

mod number = 2^n

यह counter एक specific modulus (N) तक गिनता है और फिर से शुरू हो जाता है।

ex → "mod-8 counter" है तो ये 0 से 7 तक count करेगा फिर से 0 से restart होगा।

A counter which goes through all possible state (o/p) before re-starting is called full modulus counter

ex : 2 bit

Variable modulus :- A counter in which maximum number of state can be change is called Variable modulus.

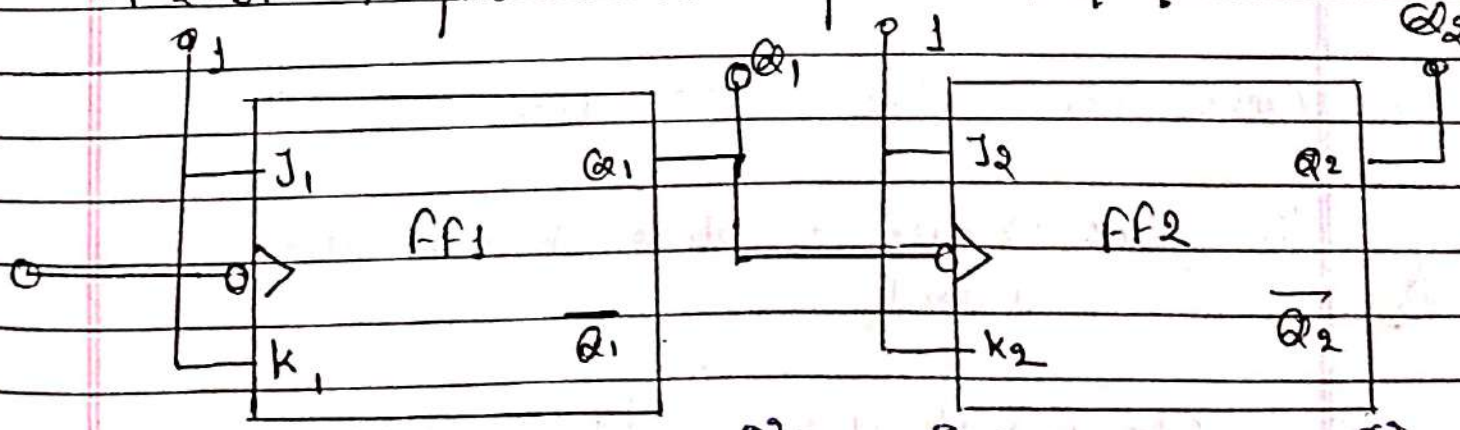
यह modulus का मतलब होता है counting range यानी कि वो maximum value तक गिनती है और फिर से शुरू से गिनती है।

Terminal counter: The final state of counter sequence is called terminal Counter.

Combination modulus : when different module counter are combine to form a new module counter is called combination modulus counter

★ n-bit ripple counter is called mod-2 counter
2-bit ripple counter / 2-bit flip flop = mod-4 counter

A 2-bit Asynchronous up counter :-



ये Negative state है क्योंकि ○ ▷ है यानी Negative triggering है।

A two bit asynchronous binary counter

Chapter – V
Logic Family

All the digital circuits are available in IC form. While producing digital ICs, different circuit configurations and manufacturing technologies are used. This results into a specific logic family. Each logic family designed in this way has identical electrical characteristics such as supply voltage range, speed of operation, power dissipation, noise margin etc.

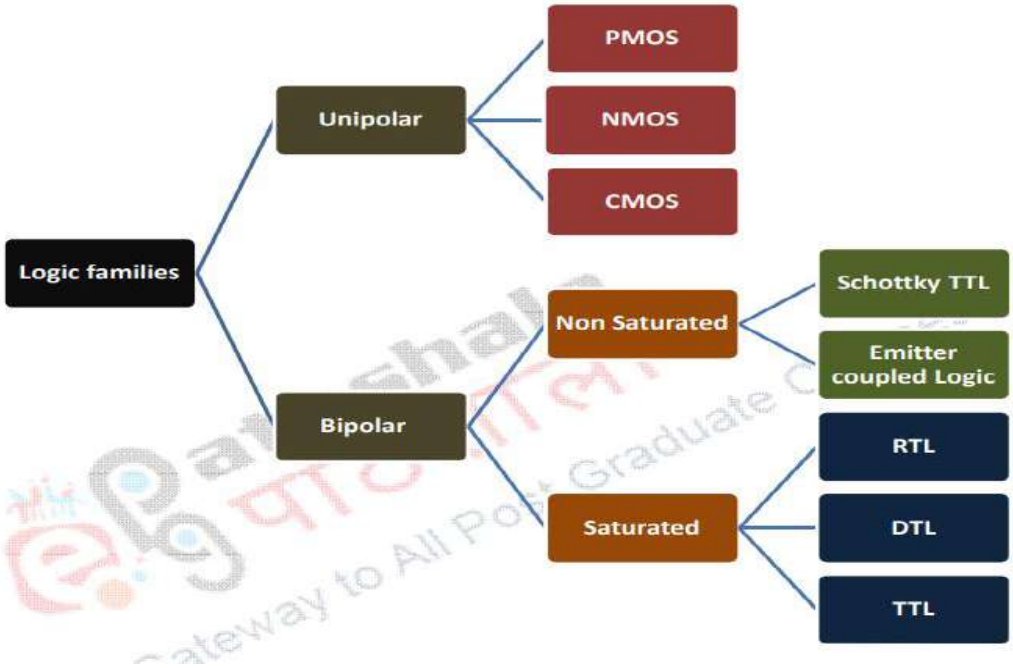
Almost all electronic gadgets make use of different digital systems for their operation. All the digital systems use some kind of digital ICs. **For the sake of simplicity in design and compatibility in constructing any complex digital system**, all digital circuits (ICs) used in the design process should be from same logic family.

For the expansion of the system, it is necessary to connect different logic circuits together. In order to connect the output of one logic circuit to the input of another logic circuit, one must have circuit with similar characteristics. If the electrical IO characteristics of these logic circuits are not similar, there is a need to design an interfacing circuit to maintain the compatibility of digital logic ICs. This interfacing circuit will match the electrical characteristics of the logic circuits. This ensures the compatibility for proper operation of the circuit.

This is the key concept behind family for logic circuits. In a logic family, the family members have similar electrical characteristics. Digital logic circuit has to be designed considering these compatibilities of different logic families in terms of different characteristics and parameters associated with the families.

A logic family refers to a group of digital logic circuits constructed using the same underlying semiconductor technology. All circuits within a particular logic family share similar characteristics in terms of speed, power consumption, input/output voltage levels, and switching behaviour.

Classification of Digital Logic Family



Logic families are the logic circuits having identical electrical parameters. It is a group of compatible ICs with the same logic levels and supply voltages for performing various logic functions. They are fabricated using a specific circuit configuration which is referred to as a Logic family. The circuit design of the basic gate of each logic family is the same. The logic family is designed by considering the basic electronic components such as resistors, diodes, transistors, and MOSFET; or combinations of any of these components. Accordingly, logic families are classified as per the construction of the basic logic circuits. Many different logic families of digital ICs have been introduced commercially are listed in

Table 1. Logic families and the components used for construction of logic family

Name of logic family	Components used
DL(Diode Logic)	Diodes
RTL(Resistor Transistor Logic)	Resistors and transistors
DTL(Diode Transistor Logic)	Diodes, transistors and resistors
TTL(Transistor Transistor Logic)	Transistors and resistors
ECL(Emitter Coupled Logic)	Transistors and diodes
PMOS(P channel Metal Oxide Semiconductor Logic)	P- MOSFETs
NMOS(N channel Metal Oxide Semiconductor Logic)	N- MOSFETs
CMOS(Complementary Metal Oxide Semiconductor Logic)	P –MOSFET and N-MOSFET

Logic families are classified according to the principle type of electronic components used in their circuitry as shown in Figure 1. They are

a. Bipolar ICs: which uses diodes and transistors (BJT)
b. Unipolar ICs: which uses MOSFETs

table 1.

Characteristics of Digital Logic Family

8.3.1 Fan Out

The fan out of a logic gate is defined as the maximum number of standard loads that the output of the gate can drive without degrading its normal operation. The term standard load is specified as the amount of current needed by an input of another gate of the same logic family.

The fan out is some times called 'loading factor' because of the fact that the output of a gate can supply a limited amount of current, above which it ceases to operate properly and is said to be overloaded. Exceeding the specified maximum number of loads may cause a malfunction.

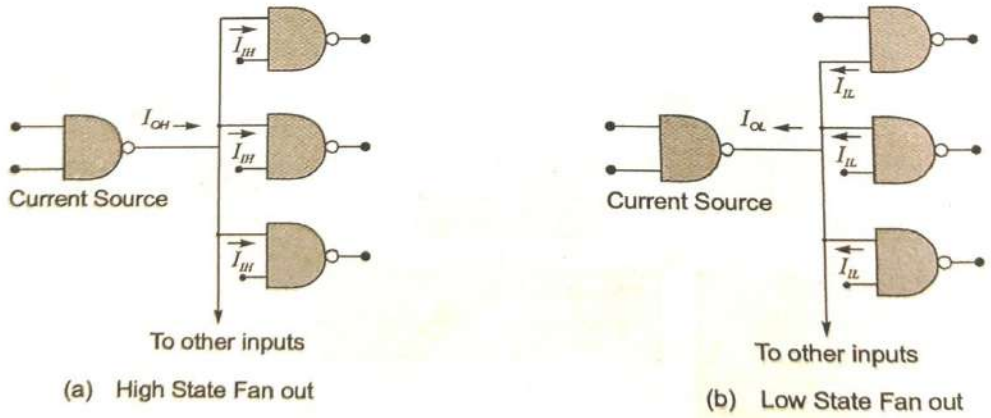
The fan out is the maximum number of inputs that can be connected to the output of a gate and it is expressed by a number. High fan out is advantageous because it reduces the need for additional drivers to drive more gates. Figure 8.10(a) shows the fan out computation of a logic circuit. Here the output of one gate is connected to one or more inputs of other gates. If the output of the gate is in high voltage level or at logic (1), then fan out is said to be HIGH state fan out,. It provides a current source I_{OH} to all the gate inputs connected to it. Each gate input requires a current I_{IH} for proper operation. Therefore HIGH state fan out is given by

$$\text{HIGH state Fan out} = \frac{I_{OH}\ (\text{min})}{I_{IH}\ (\text{min})}$$

Similarly, if the output of the logic gate is in low voltage level or at logic (0) (See Figure 8.10(b)). It provides a current Sink I_{OL} for all the gate inputs connected to it. Each gate input supplies a current I_{IL}. Thus.

$$\text{LOW state Fan out} = \frac{I_{OL}\ (\text{max})}{I_{IL}\ (\text{max})}$$

NOTE: The fan out of the gate can be calculated from any of the expression given above, whichever is smaller.



(a) High State Fan out

(b) Low State Fan out

Figure-8.10 *Fan out computation.*

8.3.2 Fan In

The Fan in of a logic gate is the number of inputs connected to the gate without impairing its normal operation. The Fan in parameter determines the functional capability of a logic circuit.

8.3.3 Power Dissipation

Every logic gate requires a certain amount of power for its operation. The power dissipation 'P_D' of a logic gate expressed in milliwatt (mW), is the power required by the gate to operate with 50% duty cycle at a specified frequency. The amount of power that is dissipated in a gate is calculated from the supply voltage V_{CC} and the current I_C drawn by the circuit. The current drain from the power supply depends on the logic state of the gate i.e. when output of the gate is in the high voltage level the current is I_{CCH} and when the output is in the low voltage level, the current is I_{CCL}. Therefore the average current is

$$I_{CC(avg)} = \frac{I_{CCH} + I_{CCL}}{2}$$

and is used to calculate the average power dissipation as

$$P_D = V_{CC} \times I_{CC(avg)} \text{ milliseconds}$$

Example - 8.2 A standard TTL NAND gate uses a supply voltage V_{CC} of 5 V and has current drains I_{CCH} = 1 mA and I_{CCL} = 3 mA. Then the average power dissipation is

Solution :

$$I_{CC(avg)} = \frac{I_{CCH} + I_{CCL}}{2} = \frac{1+3}{2} = 2\,\text{mA}$$

$\therefore$ Average power dissipation $= I_{CC(avg)} \times V_{CC}$

$$P_D = 2 \times 5 = 10\,\text{mW}$$

8.3.4 Propagation Delay

The propagation delay of a logic circuit limits the speed (frequency) at which circuit can operate and it is measured in "nanoseconds (ns)". It is defined as the average transition delay for the signal to propagate from input to output when the binary input signal changes its value. If the average transition delay is expressed by t_{pd} then

$$t_{pd} = \frac{t_{PHL} + t_{PLH}}{2} \text{ nanoseconds}$$

where t_{PHL} is the signal delay time when the output goes from a logic 0 to a logic 1 and t_{PLH} is the signal delay time when the output goes from a logic 1 to a logic 0. It is customary to measure the time between 50% point on the input and output transitions as shown in Figure 8.11.

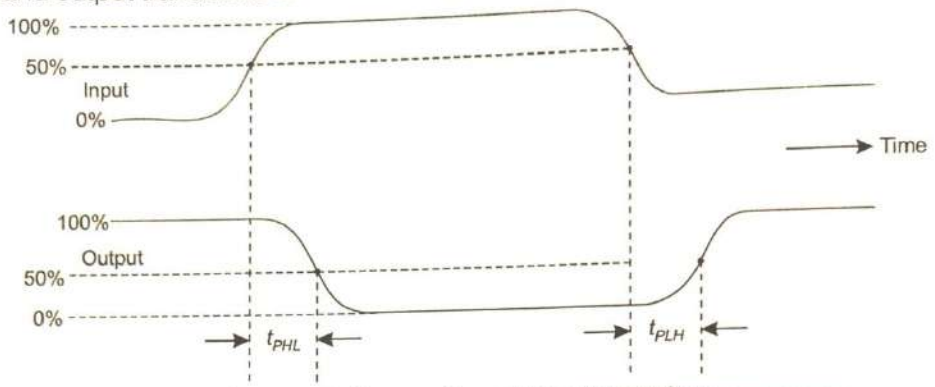


Figure-8.11 Propagation delay computation

8.3.5 Figure of Merit (FOM)

The product of "propagation delay time and power dissipation is known as figure of merit". Sometimes it is also called Speed Power Product (SPP). For best performance of a gate, minimum value of FOM is desirable. It is specified in Pico Joules (ns × mW = PJ)

$\therefore$ $\boxed{\text{FOM} = P_D \times t_{pd}}$ pico Joule

In a digital circuit, if it is desired to have high speed, i.e. low propagation delay, then there is corresponding increase in the power dissipation and vice-versa.

8.3.6 Voltage and Current Parameters

There are various currents and voltages that are useful in the design of digital systems as follows:

High level input voltage (V_{IH})

This is the minimum voltage level required at the input of a gate as logic 1.

Low Level input voltage (V_{IL})

This is the maximum voltage level that can be treated as logic 0 at the input of the gate.

High level output voltage (V_{OH})

This is the minimum voltage level required at the output of a gate as logic 1.

Low level output voltage (V_{OL})

This is the maximum voltage level that can be treated as logic 0 at the output of the gate.

High level input current (I_{IH})

This is the minimum current supplied by a driving source corresponding to HIGH level voltage.

Low level input current (I_{IL})

This is the minimum current that is supplied by a driving source corresponding to low logic level.

High level output current (I_{OH})

This is the maximum current which the gate can sink at high level.

Low level output current (I_{OL})

This is the maximum current that the gate can sink is low level (0 level).

The Figure 8.12 illustrates the currents and voltages in the HIGH and LOW states.

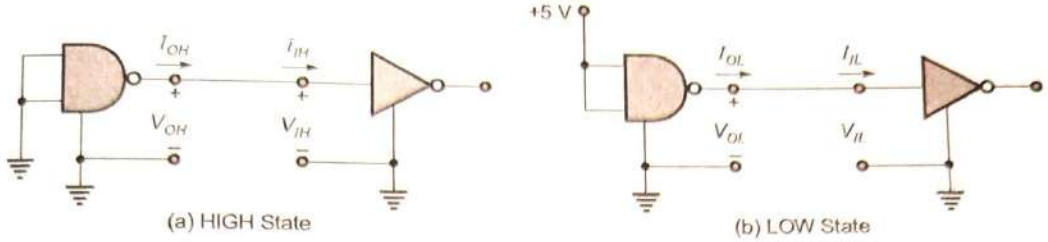


Figure-8.12 *Currents and voltages in the HIGH and LOW states*

8.3.7 Noise Margin

When the digital circuits operate in noisy environment the gates may malfunction if the noise is beyond certain limits. The noise immunity of a logic circuit is defined as the ability of a circuit to a quantitative measure of noise immunity is called noise margin.

The noise margin is the maximum noise voltage added to an input signal of a digital circuit that does not cause an undesirable change in the circuit's output. It is expressed in volts and represents the maximum noise signal that can be tolerated by the gate.

The noise margin can be calculated from the knowledge of the voltage signal available in the output of the gate and the voltage signal required in the input of the gate. Figure 8.13(a) shows the range of output voltage that can occur in a typical gate. Any voltage in the gate output between V_{CC} and V_{OH} is considered to be at high level state and between 0 to V_{OL} is considered to be at low level state. Voltages between V_{OL} and V_{OH} are indeterminate and do not appear under normal operating conditions, except during transistion between two levels.

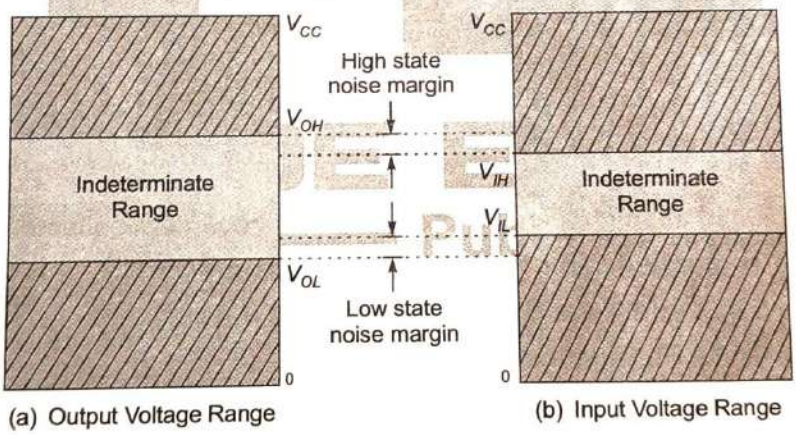


Figure-8.13 *Evaluation of Noise margin*

The corresponding ranges of voltage that are recognized by the input of the gate are shown in Figure 8.13.

In order to compensate the effect of noise, the circuit must be designed such that $V_{IL} > V_{OL}$ and $V_{IH} < V_{OH}$. Hence, the noise margin is given by

$$\boxed{(NM)_H = (V_{OH} - V_{IH}) \quad \text{or} \quad (NM)_L = (V_{IL} - V_{OL})} \text{ (which ever is smaller)}$$

NOTE

- For reliability of the system, the noise margin should be high
- From Figure 8.13, it is clear that, $V_{OH} > V_{IH} > V_{IL} > V_{OL}$

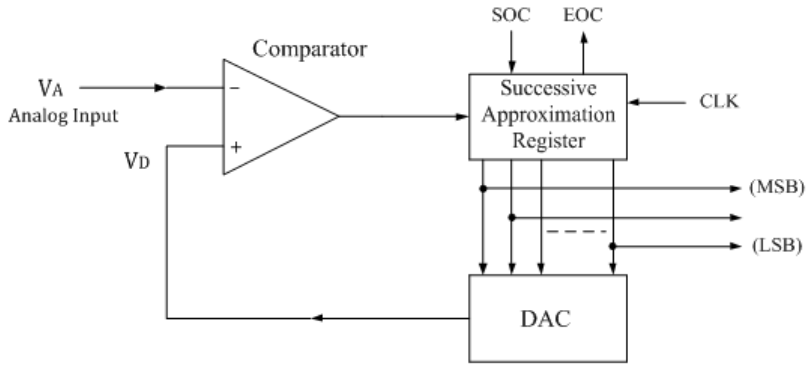
Parameter	RTL	DTL	TTL	ECL	CMOS
Components used	Resistors and transistor	Resistor, diode and transistor	Resistor, diode and transistor	Resistor and transistor	N-channel and P- channel MOSFET
Circuit	Simple	Moderate	Complex	Complex	Moderate
Noise margin [Noise immunity]	Nominal	Good	Very good	Good	Very good
Fan-out	Low (4)	Medium (8)	More (10)	High (25)	50
Power dissipation in mW per gate	12	8 - 12	10	40 - 55	0.1
Basic gate	NOR	NAND	NAND	OR-NOR	NAND/NOR
Propagation delay in ns	12	30	10	2 (ECL 10 K) 0.75 (ECL 100 K)	70
Speed power product (PJ)	144	300	100	100 (ECL 10 K) 40 (ECL 100 K)	0.7
Applications	Absolute	Absolute	Laboratory instruments.	Due to low propagation delay they are used in high speed switching applications	Due to low power consumption they are used in portable instrument where battery supply is used.
Number of functions	High	Fairly high	Very high	High	Low
Clock rate MHz	8	12 - 30	15 - 60	60 - 400	5

Successive Approximation type ADC

Successive Approximation type ADC is the most widely used and popular ADC method. The conversion time is maintained constant in successive approximation type ADC, and is proportional to the number of bits in the digital output, unlike the counter and continuous type A/D converters.

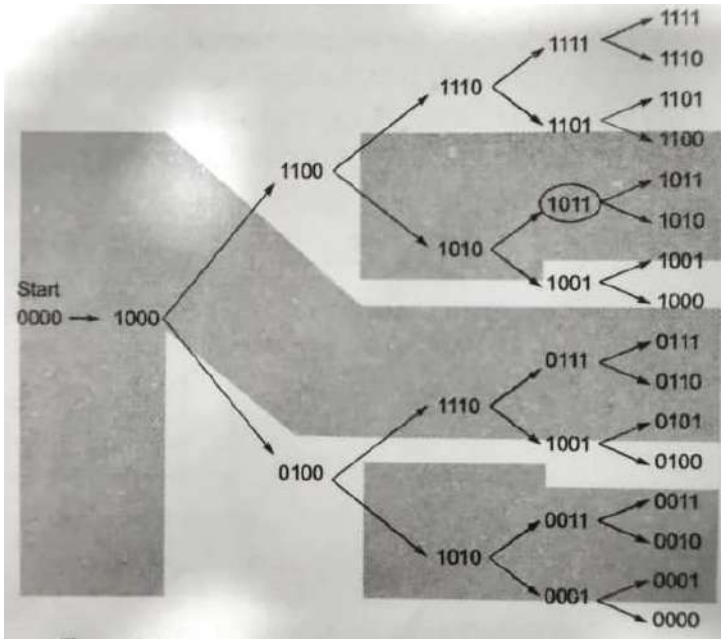
The basic principle of this type of A/D converter is that the unknown analog input voltage is approximated against an n-bit digital value by trying one bit at a time, beginning with the MSB.

The functional block diagram of successive approximation type of ADC is shown below.



It consists of a successive approximation register (SAR), DAC and comparator. The output of SAR is given to n-bit DAC. The equivalent analog output voltage of DAC, VD is applied to the non-inverting input of the comparator. The second input to the comparator is the unknown analog input voltage VA. The output of the comparator is used to activate the successive approximation logic of SAR.
When the start command is applied, the SAR sets the MSB to logic 1 and other bits are made logic 0, so that the trial code becomes 1000.

Working Principle: The principle of successive approximation process for a 4-bit conversion is explained here. This type of ADC operates by successively dividing the voltage range by half, as explained in the following steps.
(1) The MSB is initially set to 1 with the remaining three bits set as 000. The digital equivalent voltage is compared with the unknown analog input voltage.
(2) If the analog input voltage is higher than the digital equivalent voltage, the MSB is retained as 1 and the second MSB is set to 1. Otherwise, the MSB is set to 0 and the second MSB is set to 1. Comparison is made as given in step (1) to decide whether to retain or reset the second MSB.



Advantages:

1 Conversion time is very small.

2 Conversion time is constant and independent of the amplitude of the analog input signal VA.

Disadvantages:

1 Circuit is complex.

2 The conversion time is more compared to flash type ADC.

The Parallel Comparator/Flash Type ADC

The parallel comparator type ADC is also called **flash type or simultaneous ADC**. It is the fastest ADC, but it require much more circuitry than the others.

This type of converter utilizes the parallel differential comparators that compare reference voltage with the input analog voltage.

To convert an analog signal to digital signal of 'N bits, it requires **(2^N-1)** comparator, **2N** resistors, and a **(2Nx N)** priority encoder.

Working:

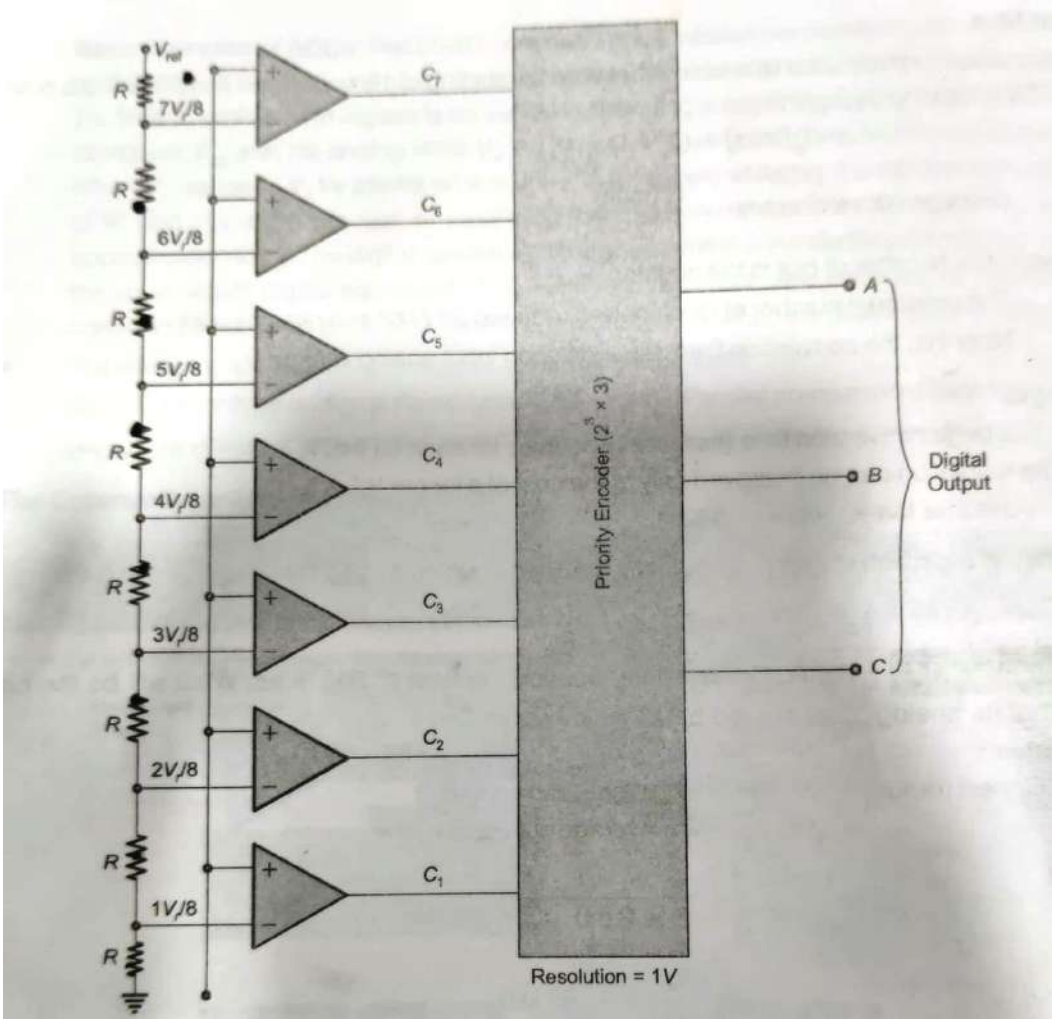


Figure shows a 3 bit flash type A/D converter which requires $(2^3-1) = 7$ comparators. The analog input which is to be converted is connected to the non-inverting terminal. Input terminal of comparator where as the inverting input terminals of Op-amps are connected to a set of reference voltage provided by voltage divider that divides it into seven equal increment levels.

Each level is compared to the analog input by a voltage comparator.

All comparator outputs are connected to a priority encoder, which produces a digital output corresponding to the input having highest priority. Thus, the digital output represents the voltage that is closest in value to the analog input.

The flash converter uses no clock signal. The conversion takes place continuously. The only delays in the conversion are in comparators and priority encoders. Thus the maximum number of clock pulse required for conversion is '1'.

The truth table of 3 bit flash ADC is shown in Table

Analog Input (V_A)	Comparator							Digital Output		
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C	B	A
$V_A < V_r/8$	0	0	0	0	0	0	0	0	0	0
$V_r/8 < V_A < 2V_r/8$	0	0	0	0	0	0	1	0	0	1
$2V_r/8 < V_A < 3V_r/8$	0	0	0	0	0	1	1	0	1	0
$3V_r/8 < V_A < 4V_r/8$	0	0	0	0	1	1	1	0	1	1
$4V_r/8 < V_A < 5V_r/8$	0	0	0	1	1	1	1	1	0	0
$5V_r/8 < V_A < 6V_r/8$	0	0	1	1	1	1	1	1	0	1
$6V_r/8 < V_A < 7V_r/8$	0	1	1	1	1	1	1	1	1	0
$V_A > 7V_r/8$	1	1	1	1	1	1	1	1	1	1

Table-9.1 Truth table of 3 bit flash ADC

It is clear that, when V_A < Vr / 8 all the comparator outputs. C1 to C7 will be HIGH, whereas with V_A > Vr/ 8 One or more comparator outputs are LOW.

For example, when V_A = 3.5V outputs C1, C2 AND C3 will be LOW and all others will be HIGH. The priority encoder will respond only to the at C3 and will produce a binary output ABC=011, which represents the digital equivalent of V_A when the resolution 1 V. When V_A is greater than 7 V, C1 TO C7 will all be LOW, and the encoder will produce ABC = 111 as the digital equivalent of V_A.