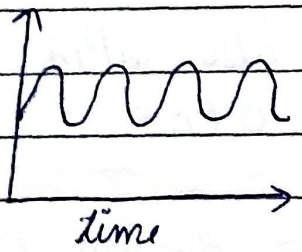
Unit-1

Date: / / Page no:

Number System

Analog & digital Signals

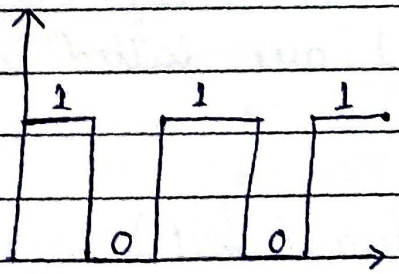
→ Analog Signals = These are continuous signals and they have values in limited range.



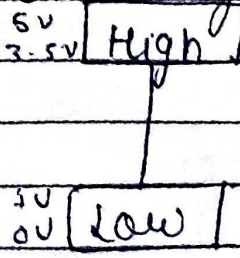
⇒ Disadvantages of Analog Signals

- Noise in channels
- When we transmit analog signals through a medium then at output we got weak signals

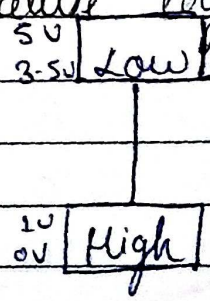
→ Digital Signals = It only changes b/w 2 discrete levels of voltage or values



→ Positive logic :-



→ Negative logic :-



Date: / / Page no:

⇒ Application & advantages of Digital System

- easier to design
- Information can be restore and retrieved easily
- Accurate
- Programmed by a set of stored instruction
- can be fabricated in chips
- fast response time
- less effected by noise

Decimal number system = It is also called arabic numerals. It has ten symbols i.e 0 to 9.
Its base or radix is 10.

NOTE = Largest digit in any number system is equals to radix-1
ex:- Largest digit in decimal no. = 10 - 1 = 9

⇒ Weight :- 45 weight of 4 is 10
weight of 5 is 1
4×10 5×1

Binary number system = It has 2 symbols i.e 0 and 1 and its radix is 2. 0 and 1 are called as bits
ex:- 0 1 0 1
MSB LSB
(Most significant bit) (Least significant bit)

⇒ 8421 System / code

8	4	2	1	Decimal
B_3	B_2	B_1	B_0	
0	0	0	0	0
0	0	0	1	1

Date: / / Page no:

B_3	B_2	B_1	B_0	Decimal
0	1	1	0	6
1	0	0	1	9

By this system we can count to 15
(vaha vaha 1 likh dena jis add karke vo decimal no. aa raha ho).
for more than 15 we use 16 bit code.

⇒ 16 8 4 2 1 System / code

16	8	4	2	1	
B_4	B_3	B_2	B_1	B_0	Decimal
1	0	0	0	0	16
1	0	1	0	0	20
1	1	0	0	1	25

here we can count to 31

⇒ formula for highest decimal no. :-
highest decimal no. = $2^n - 1$
n = bits
ex = if we are having 4 bit binary number then, highest decimal no. = $2^4 - 1$
⇒ 16 - 1 ⇒ 15

⇒ Imp facts :-
- Collection of 4 bits = a nibble
- collection of 8 bits = 1 byte
⇒ 1 byte = 2 nibble

Binary to decimal conversion
→ A no. with decimal point is represented by a series of coefficient as :- $a_4\ a_3\ a_2\ a_1\ a_0 \cdot a_{-1}\ a_{-2}\ a_{-3}$
ex:- 1 1 0 0 · 1 0
$a_3\ a_2\ a_1\ a_0$ $a_{-1}\ a_{-2}$

Date: / / Page no:

→ A no. expressed in base r system has coefficient multiplies by powers of r :-

$a_n r^n + a_{n-1} r^{n-1} + \dots a_2 r^2 + a_1 r + a_0 + a_{-1} r^{-1} + a_{-2} r^{-2} \dots + a_{-m} r^{-m}$

ex :- (1) $(11010.11)_2 = (?)_{10}$

Soln $1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 + 1\times2^{-1} + 1\times2^{-2}$

$\Rightarrow 16 + 8 + 2 + \frac{1}{2} + \frac{1}{8} \Rightarrow 26 + \frac{10}{16}$

$\Rightarrow (26.75)_{10}$ ans

(2) $(110101)_2 = (?)_{10}$

$\Rightarrow 1\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

$\Rightarrow 32 + 16 + 0 + 4 + 0 + 1 \Rightarrow (53)_{10}$

(3) $(1101101)_2 = (?)_{10}$

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

$\Rightarrow 64 + 32 + 8 + 4 + 1 \Rightarrow (109)_{10}$

(4) $(110110)_2 = (?)_{10}$

$\Rightarrow 1\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$\Rightarrow 32 + 16 + 4 + 2 \Rightarrow (54)_{10}$

(5) $(111.101)_2 = (?)_{10}$

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

$\Rightarrow 4 + 2 + 1 + \frac{1}{2} + \frac{1}{8} \Rightarrow 7 + \frac{1}{2} + \frac{1}{8}$

$\Rightarrow \frac{12 + 8 + 2}{16} \Rightarrow \frac{122}{16}\ ^{6}_{8} + 7.6 \Rightarrow (7.6)_{10}$

Decimal to Binary conversion :-

ex :- $(25.5)_{10}$

Integer part ← → fractional part

Date: / / Page no:

→ for integer part divide by 2 until the quotient becomes 0 and track the remainder

→ for fractional part multiply by 2 until it becomes 0

ex :- (1) $(25.5)_{10} = (?)_2$

⇒

2 | 25		
2 | 12	1	→ LSB
2 | 6	0	
2 | 3	0	
2 | 1	1	
	0	1 → MSB

$\Rightarrow 11001$

$\Rightarrow 0.5 \times 2 = 1.00$ 1 → MSB

$00 \times 2 = 0.00$

$\Rightarrow (25.5)_{10} = 11001.1$

(2) $(10.625)_{10} = (?)_2$

⇒

2 | 10		
2 | 5	0	→ LSB
2 | 2	1	
2 | 1	0	
	0	1 → MSB

$\Rightarrow 1010$

$\Rightarrow 0.625 \times 2 = 1.250$ 1 → MSB

$0.250 \times 2 = 0.500$ 0

$0.500 \times 2 = 1.000$ 1 → LSB

$000 \times 2 = 000$

$\Rightarrow (10.625)_{10} = (1010.101)_2$

(3) $(49.20)_{10} = (?)_2$

Date: / / Page no:

⇒

2 | 49		
2 | 24	1	→ LSB
2 | 12	0	
2 | 6	0	⇒ 110001
2 | 3	0	
2 | 1	1	
0	1	→ MSB

⇒

0.20 × 2 = 0.40	0	
[illegible]	[illegible]	
0.40 × 2 = 0.80	0	
0.80 × 2 = 1.60	1	
0.60 × 2 = 1.20	1	
0.20 × 2 = 0.40	0	

⇒ 110001.00110 ans

Octal number system:- It has digits from 0 to 7
Its radix is 8

Decimal	Binary	Octal
0	0000	0
1	0001	1
2	0010	2
⋮	⋮	⋮
7	0111	7
8	1000	10
9	1001	11

Octal to decimal conversion :-
ex:- ① $(645)_8 = (?)_{10}$
⇒ $6\times8^2 + 4\times8^1 + 5\times8^0$ ⇒ $6\times64 + 32 + 5$
⇒ $384 + 37$ ⇒ $(421)_{10}$

Date: / / Page no:

② $(75.5)_8 = (?)_{10}$
⇒ $7\times8^1 + 5\times8^0 + 5\times8^{-1}$ ⇒ $56 + 5 + \frac{5}{8}$

⇒ $\frac{448 + 40 + 5}{8}$ ⇒ $\frac{493}{8}$ 61.6 ⇒ $(61.6)_{10}$

Decimal to octal conversion
ex:- ① $(7825)_{10} = (?)_8$

8 | 7825		
8 | 978	1	→ LSB
8 | 122	2	
8 | 15	2	⇒ $(17221)_8$ ans
8 | 1	7	
0	1	→ MSB

② $(0.68)_{10} = (?)_8$

⇒

0.68 × 8 = 5.44	5	
0.44 × 8 = 3.52	3	
0.52 × 8 = 4.16	4	⇒ $(539121)_8$
0.16 × 8 = 1.28	1	
0.28 × 8 = 2.24	2	
0.24 × 8 = 1.92	1	

Octal to Binary conversion
ex:- ① $(479)_8 = (?)_2$
0100 ← 111 → 100 ⇒ $(100111100)_2$

② $(37.2)_8$
011 ← 111 → 010 ⇒ $(011111.010)_2$

Date: / / Page no:

Binary to Octal conversion

ex :- ① $(101011101.011)_2 = (?)_8 \Rightarrow 535.3$

5 3 5 3

② $(010\ 110\ 11 . 011)_2$

Hexadecimal Number System :- Base/radix = 16

It contains 10 digits from 0 to 9 and 6 character from A to F

Decimal	Binary	Hex
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

⇒ Hex to Binary Conversion :-

ex :- ① $(4F2D)_{16} = (?)_2$

0100 1111 0010 1101

$\Rightarrow (0100111100101101)_2$ ans

② $(5C4D)_{16} = (?)_2$

0101 1100 0100 1101

$\Rightarrow (0101110001001101)_2$ ans

⇒ Binary to Hex Conversion

ex :- ① $(0010\ 1001\ 1010\ 1111)_2 = (?)_{16}$

2 9 10 15

11 11

A F

$\Rightarrow (29AF)_{16}$ ans

Date: / / Page no:

② $(1011\ 0010\ 1011)_2 = (?)_{16}$

11 = B 2 11 = B $\Rightarrow (B2B)_{16}$ ans

⇒ Hex to decimal conversion

ex :- ① $(4F)_{16} = (?)_{10}$

$\Rightarrow 4 \times 16^1 + F \times 16^0 \Rightarrow 64 + 15$ $\{\because F = 15\}$

$\Rightarrow (79)_8$

② $(3A.2F)_{16} = (?)_8$

$\Rightarrow 3 \times 16^1 + A \times 16^0 + 2 \times 16^{-1} + F \times 16^{-2}$

$\Rightarrow 48 + 10 + \frac{2}{16} + \frac{15}{16^2}$ $\{\because A = 10 \text{ and } F = 15\}$

$\Rightarrow \frac{48 \times 16 \times 16 + 10 \times 16 \times 16 + 2 \times 16 + 15}{16^2 \times 16}$

$\Rightarrow \frac{12288 + 2560 + 32 + 15}{256} \Rightarrow \frac{14895}{256} \Rightarrow (58.18)_8$

⇒ Decimal to Hex Conversion

ex :- ① $(94.5)_{10} = (?)_{16}$

⇒

16 | 94		
16 | 5	14 = E → LSB	⇒ 5E
0	5 → MSB	

$\Rightarrow 0.5 \times 16 = 8.0$ 8

$0.0 \times 16 = 0$

$\Rightarrow (5E.8)_{16}$

⇒ Hex to octal conversion

$(4F.34)_{16} = (?)_8$

00100 1111 0011 0100

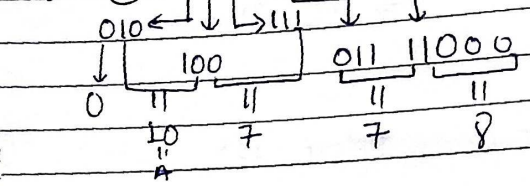
1 1 7 5

$\Rightarrow (117.15)_8$

Date: / / Page no:

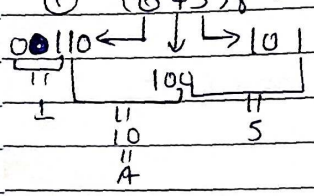
⇒ Octal to Hex conversion

ex - ① $(247.36)_8 = (?)_{16}$



$\Rightarrow (A7.38)_{16}$

② $(645)_8 = (?)_{16}$



$\Rightarrow (1A5)_{16}$ ans

Binary arithmetic

→ Addition :-

A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

ex :- ① 11001101 + 01011100 = 100010001

② 10010101 + 10100011 = 100111000

→ Subtraction :-

A	B	difference	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Date: / / Page no:

ex :- $(172)_{10} - (42)_{10}$ convert them into binary & then subtr

2	172	
2	86	0
2	43	0
2	21	1
2	10	1
2	5	0
2	2	1
2	1	0
	0	1

$\Rightarrow 10101100$

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

$\Rightarrow 101010$

$\Rightarrow$ 10101100 $\Rightarrow 1\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 0\times2^0$

− 00101010 $\Rightarrow 128 + 32 + 8 + 2$

10000010 $\Rightarrow (130)_{10}$ ans

→ Multiplication :-

A	B	ans
0	0	0
0	1	0
1	0	0
1	1	1

ex :- 1011 × 101 = 10111 + 00000X + 1011XX = 1100011 ans

111 × 01 = 111 + 000X = 0111 ans

→ Division :-

1) $0 \div 1 = 0$ 2) $1 \div 1 = 1$

ex :- 1110101 / 1001

Date: / / Page no:

⇒

```
        1 1 1
    ┌──────────
1001 | 1 1 1 0 1 0 1
     - 1 0 0 1 ↓
     ─────────
       1 0 1 1
       1 0 0 1 ↓ ↓
       ─────────
         0 1 0 0 1
           1 0 0 1
         ─────────
             0
```

ans = 1 1 1 (Quotient)
Remainder = 0

Signed & unsigned Binary Number

→ Signed Magnitude Method of Representation
It contains Sign (+/-) + Magnitude
There are 3 ways to represent Sign Binary no.
⇒ 1's compliment ⇒ 2's compliment
⇒ Signed Magnitude
In Signed Magnitude, if the left most bit is 0 then the Binary number is +ve & if 1 then -ve and other bits will tell about magnitude let us see some examples :-
① 0 1 0 1 0 1 0 ⇒ +42 ② 1 1 0 1 0 1 0 ⇒ −42

- compliment :- for base radix system there are two types of compliments
1) radix compliment (r's compliment)
2) diminished radix compliment [(r-1)'s compliment]

⇒ In binary system, radix compliment is 2's compliment & diminished radix compliment i.e (r-1)'s compliment will be (2-1)'s compliment i.e 1's compliment
⇒ In Decimal number system, r's compliment = 10's compliment

Date: / / Page no:

& (r-1)'s compliment = 9's compliment

1. one's compliment :- for finding the one's compliment of any binary number we just invert that binary bit
ex :- find one's compliment of 1 1 0 1 0 1
Sol" 0 0 1 0 1 0 ans
⇒ Using one's compliment for Subtraction
(i) Subtracting smaller no. with larger no. i.e larger - smaller
ex :- ① $(10110)_2 - (11101)_2$
step 1 :- find one's compliment of smaller no. i.e
10110 = 01001 (one's compliment)
step 2 :- Add larger no. in one's compliment of smaller no

```
  0 1 0 0 1        0 1 0 0 1
+ 1 0 1 1 0      + 1 1 1 0 1
  1 1 1 1 1        1 0 0 1 1 0
                   ↑ carry (discard this carry)
```

our result = 00110
step 3 :- add 1 in step two's result this is our ans

```
  0 0 1 1 0
        1
  0 0 1 1 1   ans//
```

② $(0011)_2 - (0111)_2$
⇒ 0011 = 1100
⇒ 1 1 0 0 0
0 1 1 1 ⇒ 0 0 1 1
1 0 0 1 1
discard this carry
⇒ adding 1 in 0 0 1 1, we get

```
  0 0 1 1
        1
  0 1 0 0
```

⇒ 0100 ans

Date: / / Page no:

two's complement -

ex:- find two's complement of 1101101

Step 1:- find one's complement of given binary no.
i.e 0010010

Step 2:- add 1 in one's complement

```
  0010010
        1
  -------
  0010011   this is two's complement
```

⇒ using two's complement for subtraction

i) Larger − Smaller

ex:- $(11101)_2 - (11010)_2$

step 1 = find two's complement of smaller no.

```
  00101
+     1
  -----
  00110   → two's complement
```

step 2 = add larger no. in this two's complement

```
  11101
  00110
  -----
 (1)00011
```

discard this carry, then the left out will be your ans
i.e 00011 ans//

ii) Smaller − larger.

ex:- $(110000)_2 - (111101)_2$

step 1 = find two's complement of larger no.

```
  000010
+      1
  -------
  000011   → two's complement
```

step 2 = add smaller no. to this two's complement

```
  110000
  000011
  ------
  110011
```

Date: / / Page no:

step 3 = If there was no carry the find two's complement of the result of addition i.e

```
  001100
+      1
  -------
 -001101   → This is the final answer.
```

just add '−' minus sign in the end because it is smaller − larger.

BCD numbers:-

It stands for Binary coded decimal in this each decimal no. is represented by 4 bit binary. It is a weighted code.

ex:- $(1234)_{10}$ = Convert into BCD

Sol" 0001 → 0010 → 0011 → 0100

⇒ (0001001000110100)BCD ans//

ex:- $(24)_8$ find Binary & BCD number

⇒ $(24)_8 \rightarrow (10100)_2$ this is Binary

⇒ (24)8 → 0010 0100 ⇒ 00100100 this is BCD

This all is decimal to BCD conversions

⇒ BCD to decimal conversions

ex:- 00110010 . 10010100
(3) (2) (9) (4) ans = 32.94

⇒ BCD addition:- If the sum of 4 bit binary is ≤9 then it is valid BCD number. But if sum is ≥9 then it is invalid BCD number. and if there is any carry, then it is also invalid BCD number.

ex:- (1) 0010
0011 It is valid because the ans is<9
0101

Date: / / Page no:

② 1001 + 0100 = 1101 + 0110 = 10011 ⇒ 0001 0011 (1 3)8

It is invalid because the ans has to make this valid we'll add 6 in this

carry

③ 0001 0110 + 0001 0101 = 0010 1011

this is <9 → this is >9

adding 6(i.e 0110) in 1011

0010 1011 + 0110 = 0011 0001 ⇒ (31)10

④ 0110 0111 + 0101 0011 = 1011 1010 + 0110 0110 = 10001 0010 0000 ⇒ (120)10

[Both are >9 then add 0110 in both]

1 2 0

Excess 3 code / XS-3 code :- It is a non weighted code it means it has no positional weighted

ex:- ① Convert 14 into XS-3 code

step 1:- add 3 in all digits i.e 1 4 +3 +3 = 0100 ← 4 7 → 0111

⇒ (01000111) this is XS 3 code

Date: / / Page no:

⑤ 25 → XS-3? 2 5 +3+3 = 5 8 ⇒ 5 = 0101, 8 = 1000 ⇒ (01011000) this is XS-3 code of 25

Gray Code :- It is also a non weighted code

Those who are non weighted code are can't be used in arithmetic, BCD. Non weighted code are cyclic code or unit distance code

In the Gray code at a time only one bit get changes

Decimal	Binary	Gray code
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110

⇒ We use Gray code in error detection techniques.

⇒ Binary to Gray Conversion

ex:- ① $(11010)_2$ → gray code

step 1:- Put the MSB as it is

step 2:- add the bits one by one & discard the carry

soln $(1+1+0+1+0)_2$

10111 this is the final ans

② $(1000)_2$ → gray code soln 1100 gray code

③ $(1011)_2$ soln 1110 gray code

⇒ Gray code to Binary :-

ex ① $(10111)_2 \rightarrow (?)_2$

step 1 = Put the MSB as it is

step 2 = add the output with bits one by one & discard the carry

1 0 1 1 1 ⇒ (11010)2

1 1 0 1 0

Date: / / Page no:

(2) 1 1 0 0 → (?)2

1 0 0 0 → (1000)2

Logic Gates = They have the ability to take decisions It is made up of no. of electronic devices (transistors, resistors etc.) It is an electric circuit It has two levels i.e High and low or True & false or on & off or 1 & 0. If we give 1 or more than 1 input signals and it produces on output signal

→ Basic Logic Gates:- It is of 3 types:-

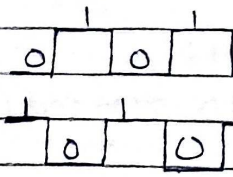
1. AND 2. OR 3. NOT

- NOT Gate = It is also known as inverter because it invert the value of input. If input is High then output will be Low & vice versa. It has only one input and produces one output

this is the Symbol

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

this is the truth table

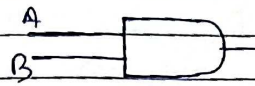


this is the timming diagram

It's IC is 7404

- AND Gate = It is simply the logical multiplication. It has two or more than two input & produces one output. If all the inputs are High then only the output is High otherwise the output will be low.

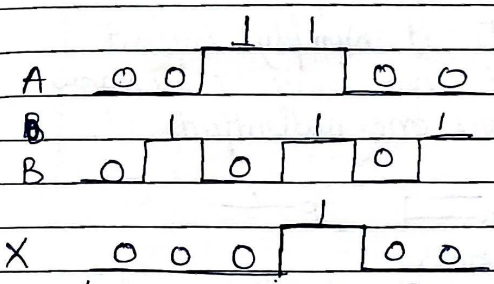
Date: / / Page no:



this is the Symbol of 2 input AND Gate

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

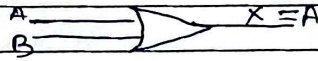
This is the truth table of 2 input And Gate



This is the timming diagram of 2 input AND Gate

It's IC is 7408

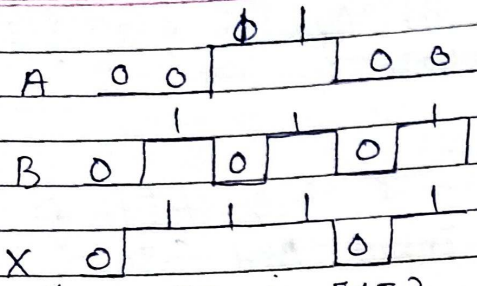
- OR Gate = It is simply the logical addition. If we give 2 or more than 2 inputs then it produces one output. If all the inputs are low then the output will be low. If any of the input is high then the output is high.



this is the Symbol for 2 input or gate

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

This is the truth table for 2 input OR Gate



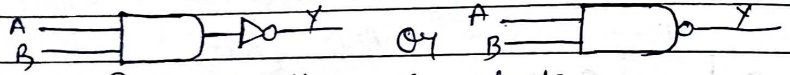
This is the timing diagram of 2 input OR Gate

It's IC is 7432

→ Universal Gates:- It is of two types :-

1. NAND 2. NOR

- NAND = It is AND + NOT Gate it simply inverted the output of AND Gate. It can have 2 or more than 2 inputs & can produce one output.



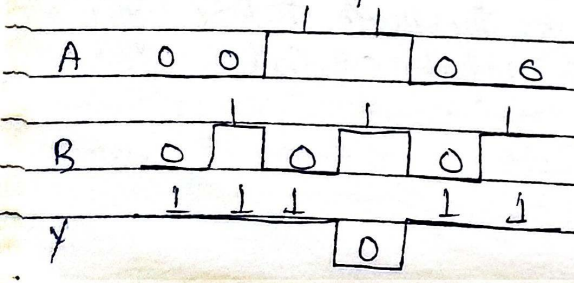
These are the symbols

If both the outputs are high then only the inputs will be low. It's IC is 7400

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

This is the truth table for 2 input NAND Gates

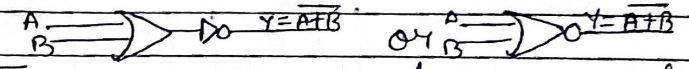
Boolean expression :- $Y = \overline{AB}$



This is the timing diagram of 2 input NAND Gate

Date: / / Page no:

- NOR Gate = It is OR + NOT. It shows inverted output of OR Gate. It can have 2 or more than 2 inputs and can produces one output. If all the inputs are low then only the output is high.

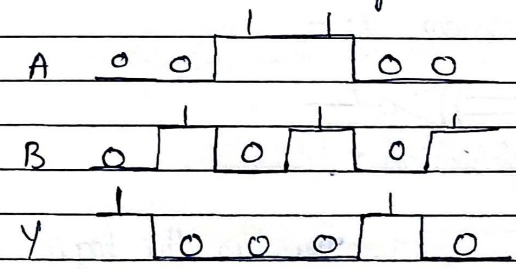


These are the symbols of 2 input NOR Gate

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

This is the Truth table for 2 input NOR Gate

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



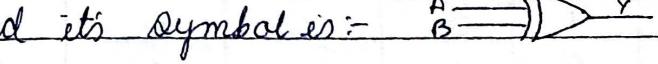
This is the timing diagram of 2 input NOR Gate

→ Special Gates :- It is of two types

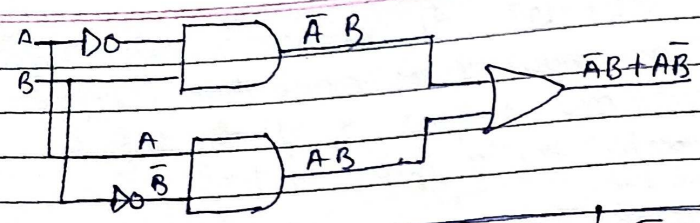
1. XOR Gate 2. XNOR

- XOR Gate = It's boolean expression is :-

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

and it's symbol is :- 

Implimenting it's Boolean expression :-



This is the logic diagram

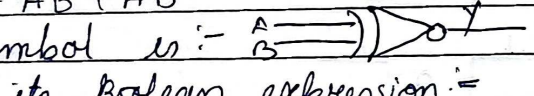
A	B	$\bar{A}$	$\bar{B}$	$\bar{A}B$	$A\bar{B}$	Y
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

$Y = A \oplus B$ this shows that we have ex XOR Gate
If both the inputs are same then the output is low otherwise the output is high

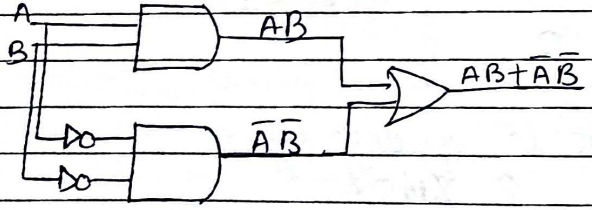
- XNOR = Its Boolean expression is :-

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

its symbol is :-



Implimenting its Boolean expression :-



This is the logic diagram

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

If both the inputs are at same then the output is high otherwise the output is low.

IC = 74266

Universal properties of Gate

firstly let us see some general expressions

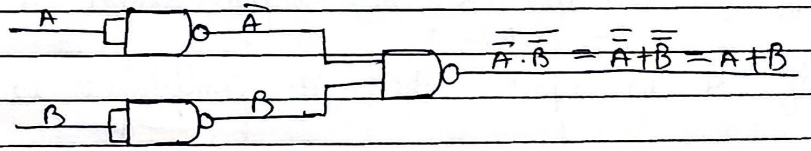
1. $\overline{X+YZ} = (X+Y)(X+Z)$
2. $\overline{X \cdot Y} = \bar{X} + \bar{Y}$
3. $\bar{\bar{X}} = X$
4. $X(X+Y) = X$
5. $\overline{X+Y} = \bar{X} \cdot \bar{Y}$

Now let use see the properties :-

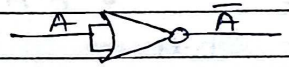
⇒ Under the universal properties NAND Gate & NOR Gate comes

⇒ designing NOT Gate using NAND Gate

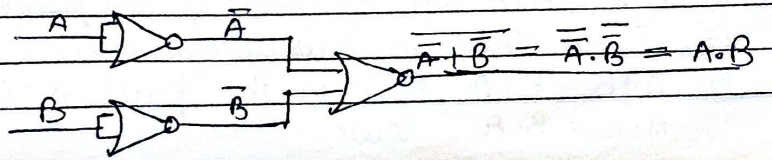
⇒ designing OR Gate using NAND Gate



⇒ designing NOT Gate using NOR Gate



⇒ designing AND Gate using NOR Gate



Date: / / Page no:

like these examples using NAND & NOR we can design any Gates.

Boolean Algebra :- It was ~~was~~ created by George Boole in 1894. It can only have Boolean constant and Boolean variable

- logical addition = OR operator
  symbol = '+'
- logical multiplication = AND operator
  symbol = '*'
- logical inversion = Not operator
  Symbol = '—' (bar), ' ' (prime)

⇒ Basic postulates of Boolean algebra

- If A≠1 then A = 0 and vice versa
- If a & b are the two inputs then
  Y = a+b
- If a and b are the two inputs then
  Y = a.b
- If a is input then its compliment is $\bar{a}$

⇒ Boolean identities :-

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

NOTE = (1) A+A & A.A are the parts of Idempotent law
(2) A.$\bar{A}$ & A+$\bar{A}$ are the parts of complimentary law
(3) $\bar{\bar{X}}$ = X this is involution law
(4) A+B = B+A this is the part of commutative law
A.B = B.A law

Date: / / Page no:

(5) X+XY = X this is absorption law
X(X+Y) = X

(6) X+$\bar{X}$Y = (X+Y)

(7) XY+YZ+$\bar{Y}$Z = XY+Z

De Morgan's Theorem :- It has two parts

➔ Theorem 1 = Compliment of a product is equal to addition of the compliment
i.e $\overline{A.B} = \bar{A}+\bar{B}$

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

Hence proved

➔ Theorem 2 = Compliment of a sum is equal to product of the compliment
i.e $\overline{A+B} = \bar{A}\bar{B}$

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

ques1) Simplify $\overline{(A+B)}+\bar{C}$

Soln By Demorgan's theorem :- $\overline{(A+B)}.\bar{\bar{C}}$

⇒ (A+B).C ans//

ques2) Simplify $\overline{(\bar{A}+B)}+\overline{CD}$

Soln ⇒ $\overline{(\bar{A}+B)}.\overline{\overline{CD}}$ ⇒ $(\bar{A}+B).CD$ ans//

Ques 3) $Y = A + \bar{A}B$ Simplify

Soln $Y = (A+\bar{A}).(A+B)$ $\{\because (A+\bar{A}) = 1 \;\&\; 1.A = A\}$

$\Rightarrow Y = 1.(A+B) \Rightarrow Y = A+B$ ans//

Ques 4) Simplify $(x+y)(x+\bar{y})(\bar{x}+y)$

Soln $(x+y)(x\bar{x}+xy+\bar{x}\bar{y}+y\bar{y})$ $\{\because x.\bar{x}=0 \;\&\; y.\bar{y}=0\}$

$\Rightarrow (x+y)(xy+\bar{x}\bar{y}) \Rightarrow (x.xy+\bar{x}\bar{y}x+xy.y+\bar{x}\bar{y}y$

$\Rightarrow xy+xy+0+0$ $\{\because x\bar{x}=0\}$

$\Rightarrow xy$ $\{\because A+A=A\}$

Ques 5) $\overline{(x\bar{y}+xyz)} + x(y+x\bar{y})$ Simplify

Soln

$\Rightarrow \overline{(x\bar{y}+xyz)} + xy + x.x\bar{y}$ $\Rightarrow \overline{(x\bar{y}+xyz)} + xy + x\bar{y}$

$\Rightarrow \overline{x\bar{y}+xyz}+xy+x(y+\bar{y})$ $\Rightarrow \overline{(x\bar{y}+xyz)}+x$

$\Rightarrow (x\bar{y}+xyz).\bar{x}$ $\Rightarrow x(\bar{y}+yz).\bar{x}$ $\{\because x.\bar{x}=0\}$

$\Rightarrow 0$//

Ques 6) $Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C}$

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

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

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

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

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

$Y = \bar{A}\bar{C} + B(\bar{A}+\bar{C}) \Rightarrow Y = \bar{A}\bar{C} + \bar{A}B + B\bar{C}$ ans//

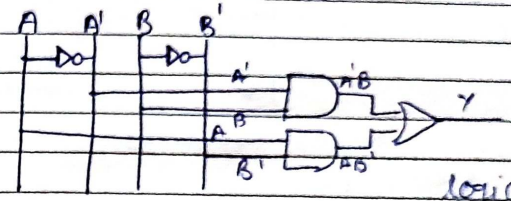
Ques 7) i) $Y = \bar{A}.B + A.\bar{B}$. draw its truth table, logical diagram

Soln

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

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

Truth table

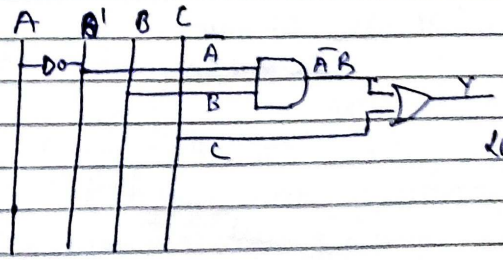


logic circuit

Ques ii) $Y = \bar{A}B + C$

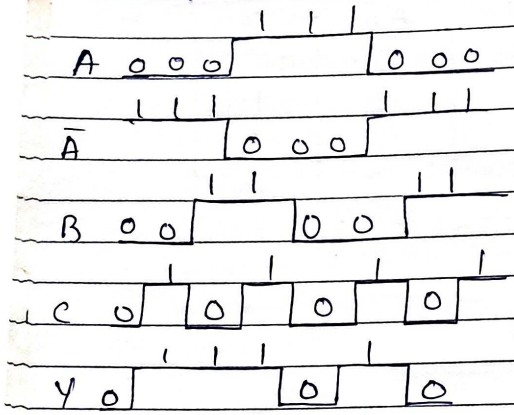
A	B	C	$\bar{A}$	$\bar{A}B$	$\bar{A}B+C$
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	1

Truth Table

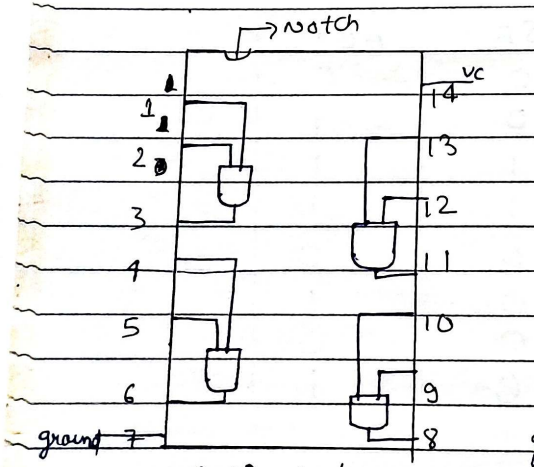


Logic diagram

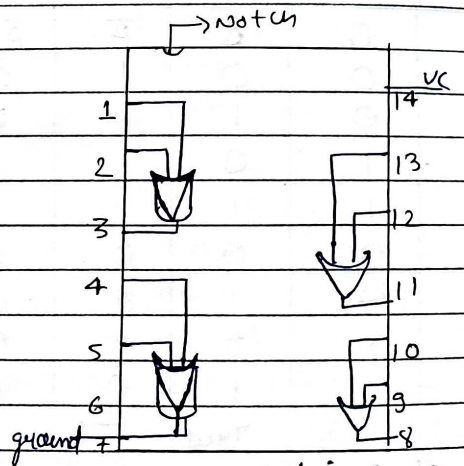
Date: / / Page no:



IC diagram of Logic Gates

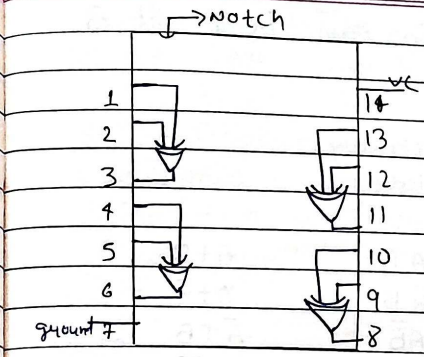


AND Gate
IC 7408



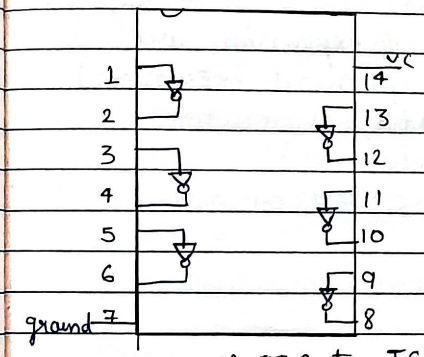
OR Gate
IC 7432

Date: / / Page no:



XOR Gate IC 7486

⇒ Application :-
- Magnitude comperator
- Binary to Gray code convertor
- adder & subtractor circuit
- Parity bit generator



NOT Gate IC 7404

⇒ Application of XNOR Gate
- even parity bit generator
- even parity bit checker
- Comperator

CMOS IC of XNOR = ~~74266~~ 4077
TTL IC of XNOR = 74266

SOP, POS, Minterm & Maxterm

→ SOP = Sum of product ex:- A·B + B·C etc.
In SOP we get minterms

→ POS = Product of sum ex:- (A+B)·(B+C) etc
In POS we get maxterms

→ Minterms = In min terms if A = 0 then it is $\bar{A}$ and if A = 1 then it'll remain same i.e A. It is represented by 'm'

→ Maxterm = In max term if A = 1 then it is $\bar{A}$

Date: / / Page no:

and if A = 0 then it'll remain same i.e A. It is represented by 'M'

The below truth table is imaginary :-

Input			output	Product terms	sum terms
A	B	C			
0	0	1	$\bar{A}\bar{B}$		$A+B$
0	1	0	$\bar{A}B$		$A+\bar{B}$
1	0	1	$A\bar{B}$		$\bar{A}+B$
1	1	1	AB		$\bar{A}+\bar{B}$

for min terms

[output K eqn K liye vahi terms aayengi jinke aage 1 hai

$\Rightarrow C = \bar{A}\bar{B} + A\bar{B} + AB$ (This type of expression are canonical expression)

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

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

$\Rightarrow C = A+\bar{B}$ (This is the reducted eqn)

Now let us see is this is correct or not

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

It shows the reduced eqn is correct

Ques) find the min terms of $BC+A$

Soln $BC+A$ → In this → In these two places possibilities are

Space three possibilities are :-

i) ABC ii) $A\bar{B}C$ iii) $AB\bar{C}$ iv) $A\bar{B}\bar{C}$

i) ABC ii) $\bar{A}BC$

Now adding all the possibilities

Date: / / Page no:

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

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

NOTE :- (1) Minterm mai output 1 mai jisme 1 hoga vahi eqn likhi jayegi.

(2) Maxterm mai output mai jisme 0 hoga vahi eqn likhi jayegi.

(3) $F(A,B,C) = \Sigma(1,4,5,6,7)$ → this shows we have to find minterm

(4) $F(X,Y,Z) = \Pi(0,2,4,5)$ → this shows we have to find maxterm (& maxterm)

(5) Conversion in minterms :- $F(A,B,C) = \Sigma(1,4,5,6,7)$ iska prime/compliment mai vo terms aayegi jo without compliment mai nahi hai. i.e $F'(A,B,C) = \Sigma(0,2,3)$ → [iska forse Prime karunga to $F = \Pi(0,2,3)$ milega]

(6) Conversion in maxterm :- Simply if we want to change min into max then, firstly change the sign and then put the missing values.

Karnaugh Map (K-Map) :- The K-Map is a systemic method for simplifying & manipulating Boolean expression

→ Two variable K-Map

A \ B	0	1
0	0 : 00	1 : 01
1	2 : 10	3 : 11

agar m likha hai to given position par 1 rakh do or agar M likha hai to given position par 0 rakh do

Q1) $\Upsilon m(0,1,2)$ find the eqn

Date:__/__/__ Page no:_____

A \ B	0	1
0	0 1	1 1
1	2 1	3 0

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

(2) $Y(0,1,2,3)$

A \ B	0	1
0	0 1	1 1
1	2 1	3 1

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

(3) $Y(0,1,3)$

A \ B	0	1
0	0 1	1 1
1	2 0	3 1

$Y = \bar{A} + B$

Three variable K-Map

A \ BC	00	01	11	10
0	0 000	1 001	3 010	2 011
1	4 100	5 101	7 110	6 111

Q1) $Y = m(0,1,3,4,5)$

A \ BC	00	01	11	10
0	0 1	1 1	3 1	2 0
1	4 1	5 1	7 0	6 0

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

(2) $Y = m(0,1,3,5,7)$

A \ BC	00	01	11	10
0	0 1	1 1	3 1	2 0
1	4 0	5 1	7 1	6 0

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

(3) $Y = M(0,1,3,5,7)$

A \ BC	00	01	11	10
0	0 0	1 0	3 0	2 1
1	4 1	5 0	7 0	6 1

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

(4) $Y = m(4,0,2,1)$

A \ BC	00	01	11	10
0	0 1	1 1	3 0	2 1
1	4 1	5 0	7 0	6 0

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

(5) $Y = m(1,2,3,6,7)$

A \ BC	00	01	11	10
0	0 0	1 1	3 1	2 1
1	4 0	5 0	7 1	6 1

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

Date:__/__/__ Page no:_____

four variable K-Map

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

This is the basic diagram for four variable K-Map

(1) $Ym(4,5,6,7,12,13,14,15)$

AB \ CD	00	01	11	10
00	0 0	1 0	3 0	2 0
01	4 1	5 1	7 1	6 1
11	12 1	13 1	15 1	14 1
10	8 0	9 0	11 0	10 0

$Y = B$ ans//

(2) $Ym(0,1,2,8,9,10,11)$

AB \ CD	00	01	11	10
00	0 1	1 1	3 1	2 1
01	4	5	7	6
11	12	13	15	14
10	8 1	9 1	11 1	10 1

$Y = \bar{B}$ ans//

Don't care in K'Map

In don't care condition we put "X" this symbole in the position that is given as don't care condition let us see some examples:-

Q1) $F(A,B,C) = \Sigma m(2,3,4,5) + \Sigma d(6,7)$

A \ BC	00	01	11	10
0	0	1	3 1	2 1
1	4 1	5 1	7 X	6 X

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

In case of minterms we put X = 1
In case of maxterm we put X = 0

$\Rightarrow$

A \ BC	00	01	11	10
0	0	1	3 1	2 1
1	4 1	5 1	7 1	6 1

$Y = A + B$ this is the final ans//

Ques2) $f(A,B,C,D) = \Sigma m(0,2,3,6,7,12,13,14) + \Sigma d(1,4,11)$

Sol:

AB \ CD	00	01	11	10
00	0 1	1 X	3 1	2 1
01	4 X	5 0	7 1	6 1
11	12 1	13 1	15 X	14 1
10	8 0	9 0	11 X	10 0

{Putting X as 1}

$\Rightarrow f = \bar{A}\bar{B} + \bar{A}C + AB$

Ques3) $f(A,B,C,D) = \pi M(5,8,9,10)\ \pi D(1,4,11,15)$

AB \ CD	00	01	11	10
00	0 X	1 X	3 1	2 1
01	4 X	5 0	7 1	6 1
11	12 1	13 1	15 X	14 1
10	8 0	9 0	11 X	10 0

{Putting X = 0}

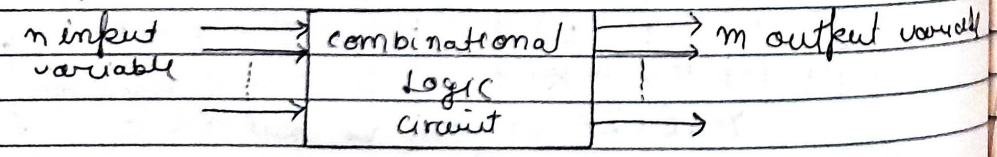
$f = (A+\bar{B}) \cdot (\bar{A}+B+\bar{C}) \cdot (\bar{A}+C+D)$

Combinational Circuit

It is divided into 4 types

① Half adder ② full adder ③ Half Subtractor ④ full Subtractor

In combinational circuit the output depends on present input only.



→ Adders = Adders are combination of logic gates that combines binary values to obtain a sum. These are classified into 2 types

① Half adder ② full adder

⇒ Half adders = A combinational circuit that performs addition of two bits is called as half adder

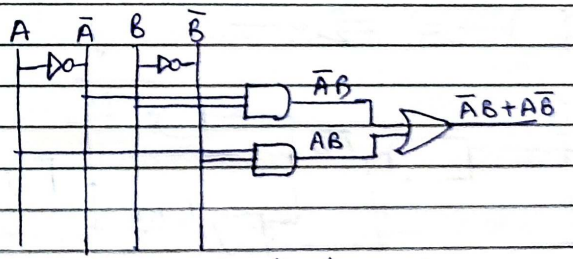
Inputs		output	
A	B	Sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth table

⇒ eqn of sum :- $S = \bar{A}B + A\bar{B} = A \oplus B$

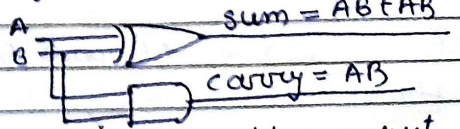
⇒ eqn of carry :- $C = AB$

implimenting eqn of sum :-



implimenting eqn of carry :-

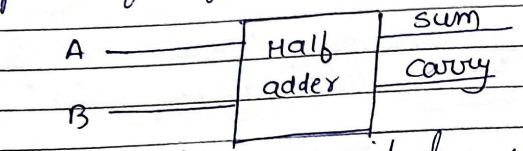
implimenting by Exor Gate



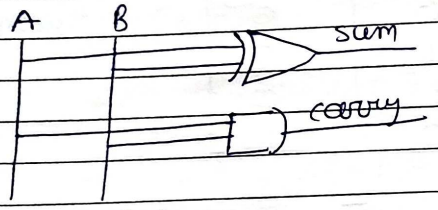
Half adder circuit

Date:__/__/__ Page no:____

logic Symbol of Half adder :-



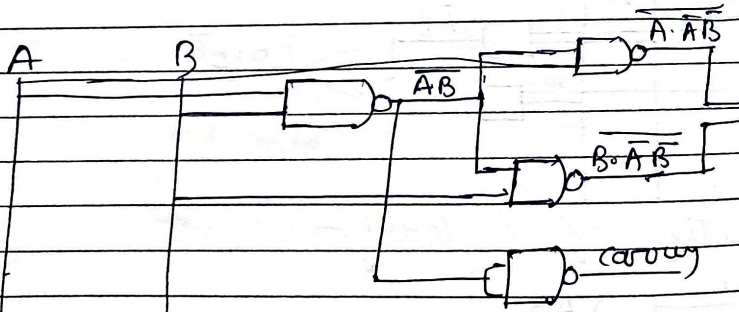
⇒ Impliment Half adder circuit by NAND Gate



This is Half adder circuit

→ By NAND Gate

{NOTE :- ① = ② = ③ =



sum = $\overline{A}B + A\overline{B}$ adding $A\overline{A}$ and $B\overline{B}$

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

Date:__/__/__ Page no:____

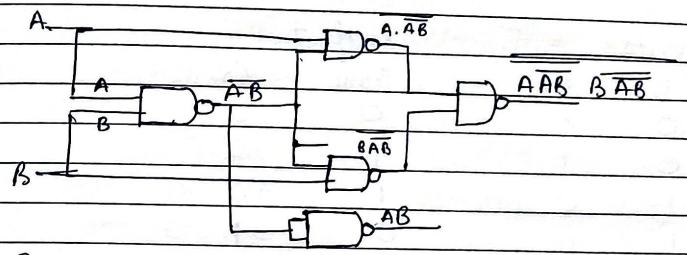
⇒ $A(\overline{A}+\overline{B}) + B(\overline{A}+\overline{B})$ ⇒ $A \cdot \overline{A\overline{B}} + B \cdot \overline{A\overline{B}}$

⇒ $\overline{A \cdot \overline{A\overline{B}} \cdot B \cdot \overline{A\overline{B}}}$ (By Demorgan's theoram)

⇒ $\overline{A \cdot \overline{A\overline{B}}}\ \overline{B\overline{A\overline{B}}}$

for carry :- $C = AB = \overline{\overline{AB}}$

Cleare logic circuit :-



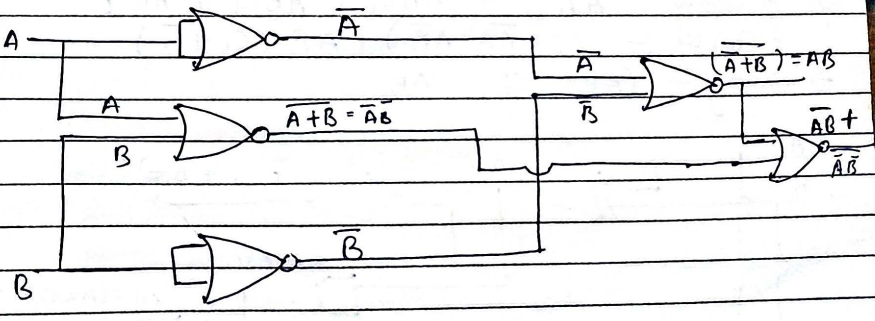
→ By NOR Gate

Carry :- $C = AB = \overline{\overline{AB}}$ {∵ $\overline{A\overline{B}} = \overline{A} + \overline{B}$ (By De. Morgan's theoram)}

⇒ $C = \overline{\overline{A}+\overline{B}}$

sum :- $sum = AB + \overline{A}\overline{B}$

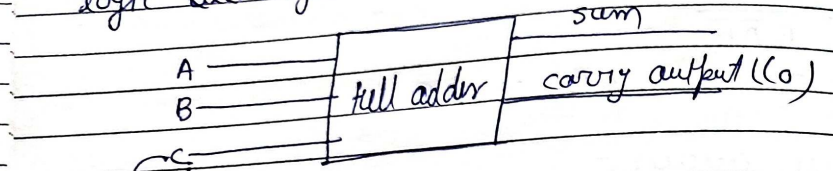
⇒ $sum = \overline{(AB + \overline{A}\overline{B})}$



Date: / / Page no:

⇒ full adder :-

logic symbol :-



Inputs			Output	
A	B	Ci	Sum	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

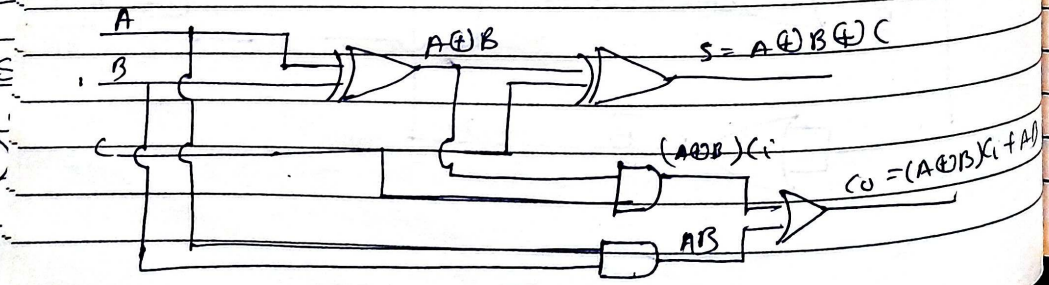
eqn of Sum :- $\bar{A}\bar{B}C_i + \bar{A}B\bar{C}_i + A\bar{B}\bar{C}_i + ABC_i$

$Sum = A \oplus B \oplus C$

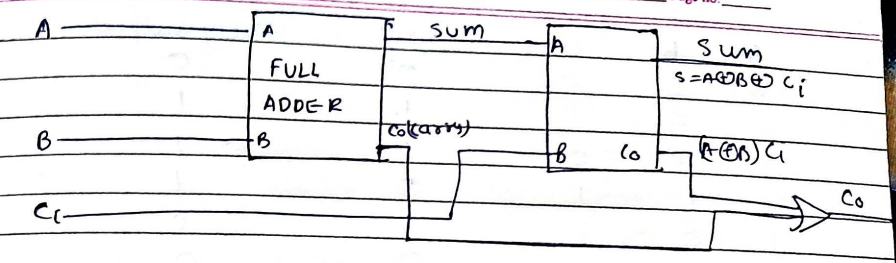
eqn of carry :- $\bar{A}BC_i + A\bar{B}C_i + AB\bar{C}_i + ABC_i$

$Carry :- C_i(\bar{A}B + A\bar{B}) + AB(C_i + \bar{C}_i)$

⇒ $Carry = (A \oplus B)C_i + AB$



Date: / / Page no:



full adder made of two Half adder

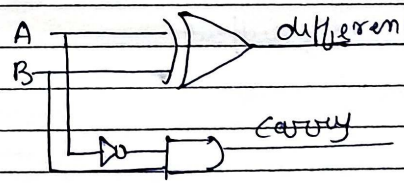
➔ Subtractor :- These are classified into 2 parts

⇒ Half Subtractor Subtracts 2 bits

Input		Output	
A	B	difference	Carry
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

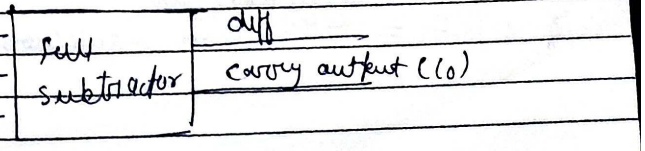
difference :- $\bar{A}B + A\bar{B} = A \oplus B$

Carry = $\bar{A}B$



⇒ full Subtractor :- Subtracts 3 bit

logic symbol



Date: / / Page no:

A	B	Ci	diff.	Carry output
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

diff :- $\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

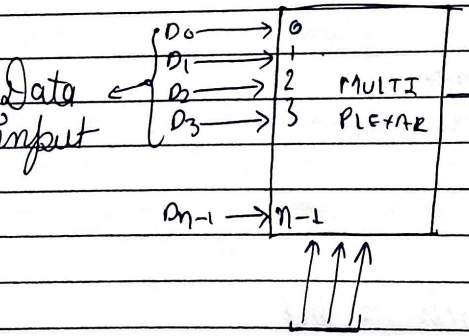
diff = A⊕B⊕C

Carry $= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C + ABC$

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

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

Multiplexers :- Many inputs, one output

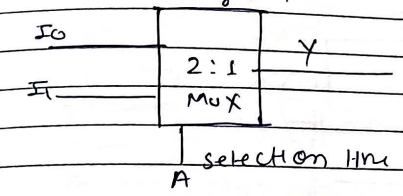


If inputs = 2^N then selection line = N

Date: / / Page no:

→ 2 : 1 Multiplexer :- $2^{(1)} = 2$ → selection line

BLOCK Diagram



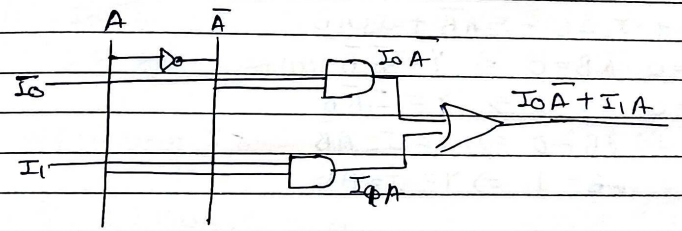
A	Y
0	I_0
1	I_1

Truth Table

Boolean expression :- $Y = I_0\bar{A} + I_1A$

Put A = 0 ⇒ $Y = I_0$

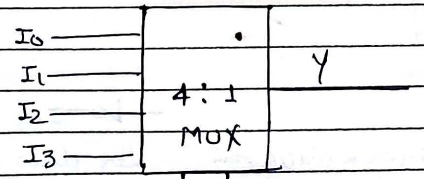
Put A = 1 ⇒ $Y = I_1$



In logic diagram we uses 2 and Gate

NOTE :- If input enable E = 0 then it is enable
If E = 1 then it is disable

→ 4 : 1 Multiplexer :- $2^{(2)} = 4$ → selection line



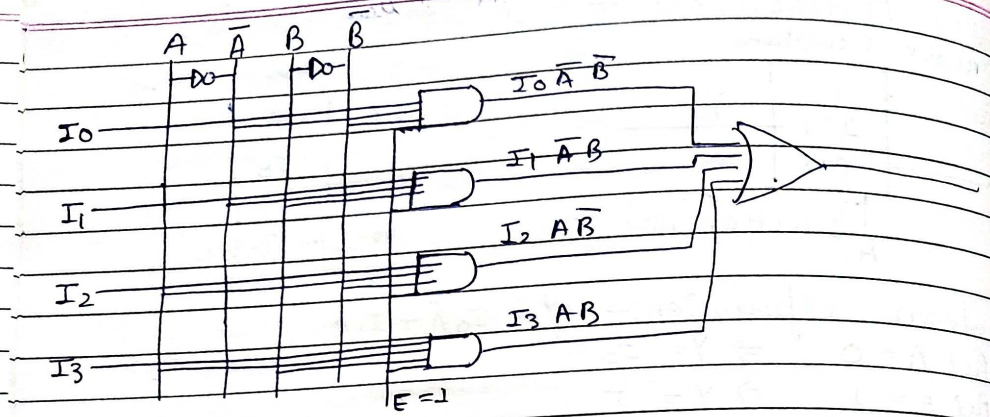
Block diagram

A	B	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Truth Table

In logic diagram we uses 4 AND Gate

Date: / / Page no:



$Y = I_0\bar{A}\bar{B} + I_1\bar{A}B + I_2A\bar{B} + I_3AB$

Put $A=0$ & $B=0 \Rightarrow Y = I_0\bar{A}\bar{B}$

Put $A=0$ & $B=1 \Rightarrow Y = I_1\bar{A}B$

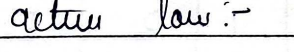
Put $A=1$ & $B=0 \Rightarrow Y = I_2A\bar{B}$

Put $A=1$ & $B=1 \Rightarrow Y = I_3AB$

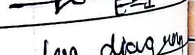
E	A	B	Y
0	X	X	Nthing
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3

This is the truth table of active high

Q ⇒ for active low :-



Block diagram

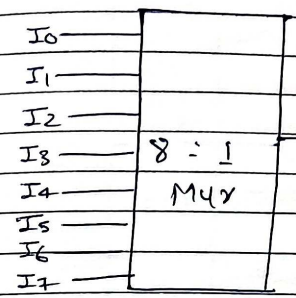


Log diagram

E	A	B	Y
1	X	X	X
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3

Date: / / Page no:

→ 8 : 1 Mux :- $2^3 = 8$ ← selection line



A	B	C	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

$Y = I_0\bar{A}\bar{B}\bar{C} + I_1\bar{A}\bar{B}C + I_2\bar{A}B\bar{C} + I_3\bar{A}BC + I_4A\bar{B}\bar{C} + I_5A\bar{B}C + I_6AB\bar{C} + I_7ABC$

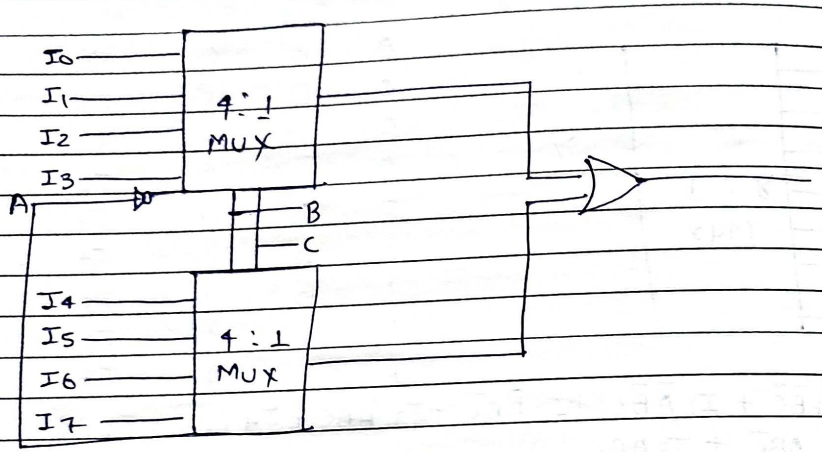
Designing 8 : 1 Mux with 4 : 1 Mux

⇒ 8 : 1 MUX :-

A	B	C	Output	
0	0	0	I_0	→ 4 : 1 Mux
0	0	1	I_1	
0	1	0	I_2	
0	1	1	I_3	
1	0	0	I_4	→ 4 : 1 MUX
1	0	1	I_5	
1	1	0	I_6	
1	1	1	I_7	

⇒ two 4 : 1 mux are required to make one 8 : 1 mux.

Date: / / Page no:

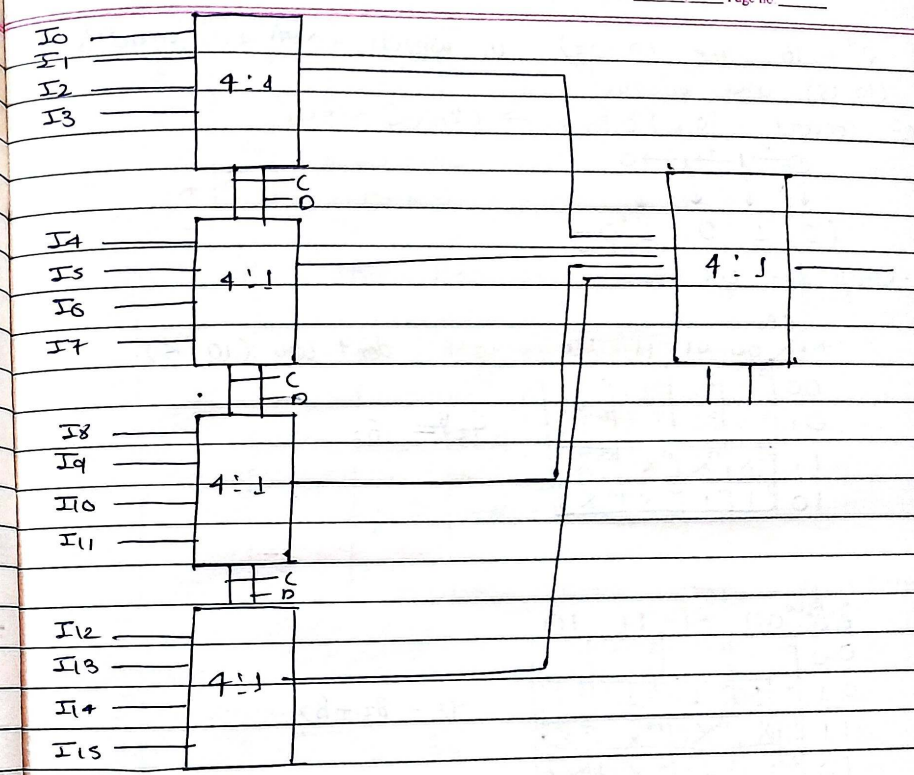


Designing 16:1 MUX with 4:1 MUX

⇒ 16:1 MUX :-

A	B	C	D	Y_1	Y_2	Y_3	Y_4	final output
0	0	0	0	I_0	I_4	I_8	I_{12}	I_0
0	0	0	1	I_1	I_5	I_9	I_{13}	I_1
0	0	1	0	I_2	I_6	I_{10}	I_{14}	I_2
0	0	1	1	I_3	I_7	I_{11}	I_{15}	I_3
0	1	0	0	I_0	I_4	I_8	I_{12}	I_4
0	1	0	1	I_1	I_5	I_9	I_{13}	I_5
0	1	1	0	I_2	I_6	I_{10}	I_{14}	I_6
0	1	1	1	I_3	I_7	I_{11}	I_{15}	I_7
1	0	0	0	I_0	I_4	I_8	I_{12}	I_8
1	0	0	1	I_1	I_5	I_9	I_{13}	I_9
1	0	1	0	I_2	I_6	I_{10}	I_{14}	I_{10}
1	0	1	1	I_3	I_7	I_{11}	I_{15}	I_{11}
1	1	0	0	I_0	I_4	I_8	I_{12}	I_{12}
1	1	0	1	I_1	I_5	I_9	I_{13}	I_{13}
1	1	1	0	I_2	I_6	I_{10}	I_{14}	I_{14}
1	1	1	1	I_3	I_7	I_{11}	I_{15}	I_{15}

Date: / / Page no:



BCD to Gray code convertor through logic & Map

	BCD				Gray			
	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1

Date: / / Page no:

$2^4 = 16$ i.e (0-15) in which (0-9) are valid & (10-15) are invalid

ex:- convert $(0\,1\,1\,0)_{BCD} \rightarrow (?)_{Gray}$

$0 \xrightarrow{+} 1 \xrightarrow{+} 1 \xrightarrow{+} 0$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

$(0\ 1\ 0\ 1)_{Gray}$

⇒ Kmap for G_3

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12 X	13 X	15 X	14 X
10	8 1	9 1	11 X	10 X

don't care (10-15)

$G_3 = B_3$

⇒ Kmap for G_2

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	4 1	5 1	7 1	6 1
11	12 X	13 X	15 X	14 X
10	8 1	9 1	11 X	10 X

$G_2 = B_3 + B_2$

⇒ Kmap for G_1

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3 1	2 1
01	4 1	5 1	7	6
11	12 X	13 X	15 X	14 X
10	8	9	11 X	10 X

$G_1 = B_2\overline{B_1} + \overline{B_2}B_1$

$G_1 = B_2 \oplus B_1$

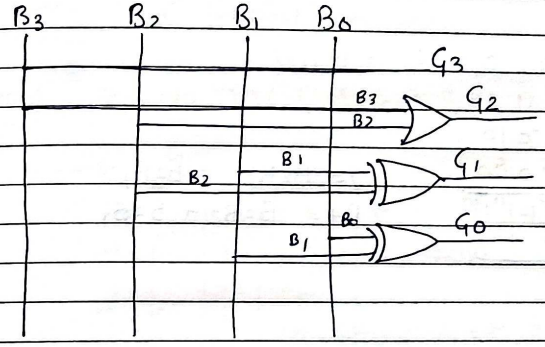
⇒ Kmap for G_0

Date: / / Page no:

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1 1	3	2 1
01	4	5 1	7	6 1
11	12 X	13 X	15 X	14 X
10	8	9 1	11 X	10 X

$G_0 = \overline{B_1}B_0 + B_1\overline{B_0}$

$G_0 = B_1 \oplus B_0$



Binary to BCD Converter

	Binary NO.				BCD NO.				
	B_3	B_2	B_1	B_0	D_4	B_3	D_2	D_1	D_0
0 -	0	0	0	0	0	0	0	0	0
1 -	0	0	0	1	0	0	0	0	1
2 -	0	0	1	0	0	0	0	1	0
3 -	0	0	1	1	0	0	0	1	1
4 -	0	1	0	0	0	0	1	0	0
5 -	0	1	0	1	0	0	1	0	1
6 -	0	1	1	0	0	0	1	1	0
7 -	0	1	1	1	0	0	1	1	1
8 -	1	0	0	0	0	1	0	0	0
9 -	1	0	0	1	0	1	0	0	1
10 -	1	0	1	0	1	0	0	0	0

Date: / / Page no:

	B_3	B_2	B_1	B_0	D_4	D_3	D_2	D_1	D_0
11 -	1	0	1	1	1	0	0	0	1
12 -	1	1	0	0	1	0	0	1	0
13 -	1	1	0	1	1	0	0	1	1
14 -	1	1	1	0	1	0	1	0	0
15 -	1	1	1	1	1	0	1	0	1

⇒ kmap for D_4 $m(10, 11, 12, 13, 14, 15)$

B_3B_2 \ B_1B_0	00	01	11	10
00	0	2	3	1
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$Y = B_3B_2 + B_3B_1$

$\Rightarrow D_4 = B_3B_2 + B_3B_1$

⇒ kmap for D_3 $m(8, 9)$

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$Y = B_3\bar{B}_2\bar{B}_1$

$\Rightarrow D_3 = B_3\bar{B}_2\bar{B}_1$

⇒ kmap for D_2 $m(4, 5, 6, 7, 14, 15)$

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$Y = \bar{B}_3B_2 + B_2B_1$

$\Rightarrow D_2 = \bar{B}_3B_2 + B_2B_1$

Date: / / Page no:

⇒ kmap for D_1 $m(2, 3, 6, 7, 12, 13)$

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$Y = \bar{B}_3B_2 + B_3\bar{B}_2$

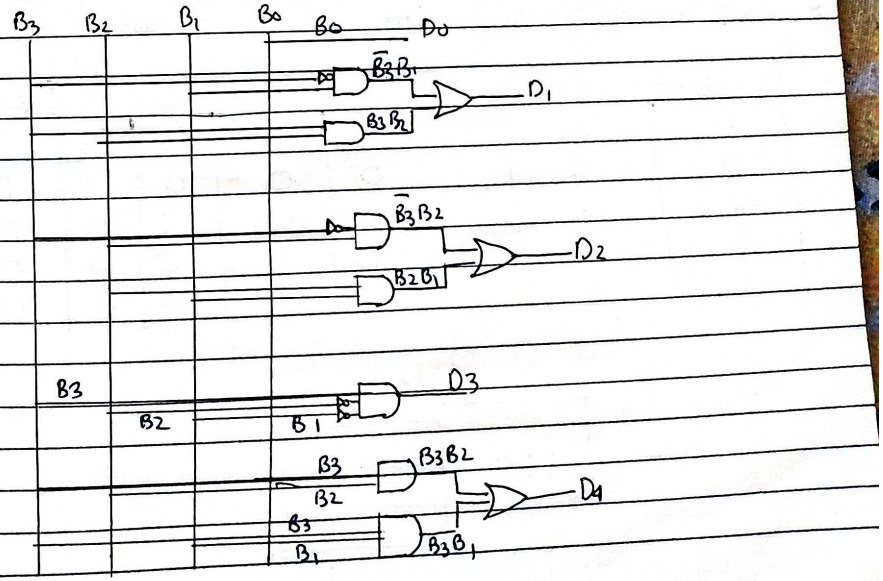
$\Rightarrow D_1 = \bar{B}_3B_1 + B_3\bar{B}_2$

⇒ kmap for D_0 $m(1, 3, 5, 7, 9, 11, 13, 15)$

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

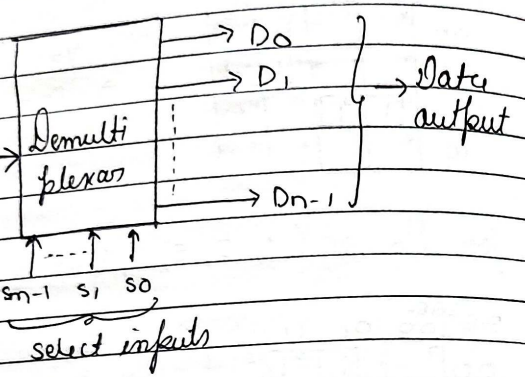
$\Rightarrow Y = B_0$

$\Rightarrow D_0 = B_0$



Demultiplexar :- These are opposite of multiplexar. It performs one to many operations

e) Block diagram



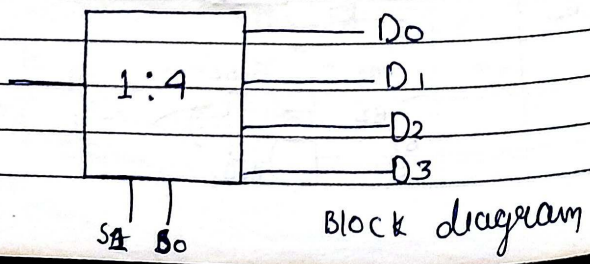
No. of output lines = n
Select lines = m
$\Rightarrow \boxed{n = 2^m}$

There are several types of Demultiplexar which are as follows:-
① 1:4 DemUX ② 1:8 Demux
③ 1:16 DemUX

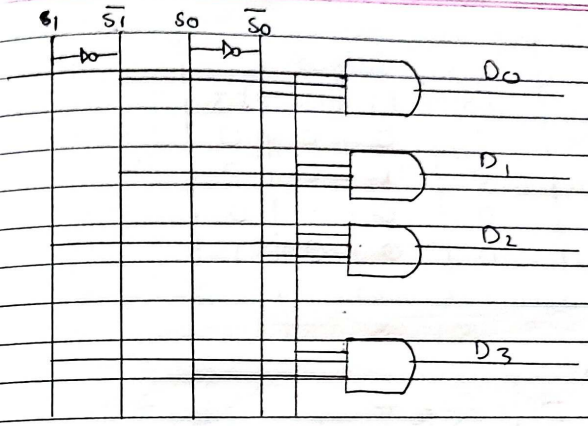
▶ 1:4 Demultiplexar (one line to 4 line)

Select Input		Outputs				Result
S_1	S_0	D_3	D_2	D_1	D_0	
0	0	0	0	0	1	D_0
0	1	0	0	1	0	D_1
1	0	0	1	0	0	D_2
1	1	1	0	0	0	D_3

Truth Table



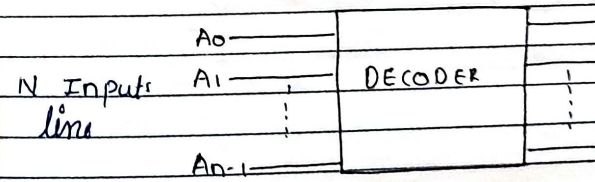
Block diagram



1:4 DEMULTIPLEXAR

Decoders :- It gets active through microprocessor and it is used to select multiple devices.

A decoder is a logic circuit that converts an N-bit binary input code into M output lines such that only one output line is activated for each one of the possible combination of inputs



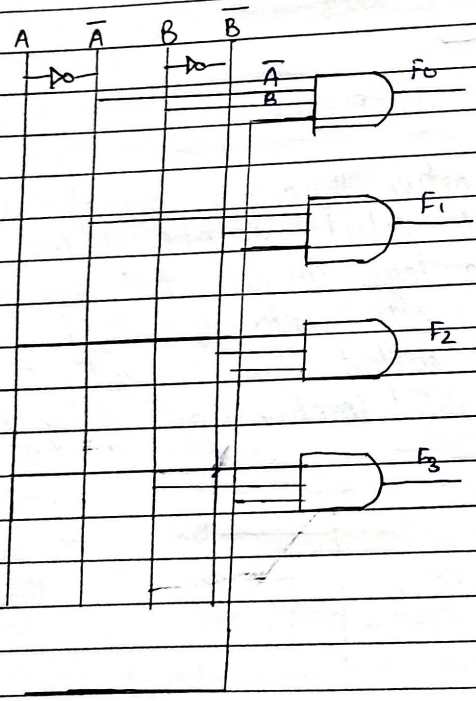
There are several types of decoder that are as follows

1) 2:4 decoder 2) 3:8 decoder
3) 4:16 decoder

Date: / / Page no:

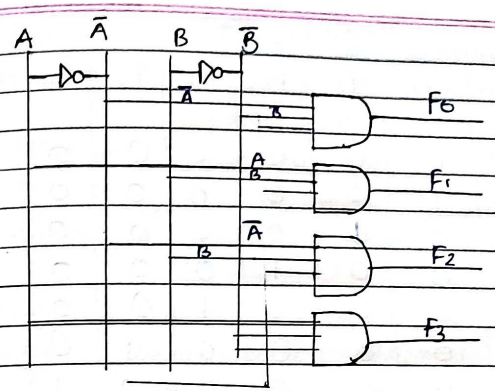
2:4 Decoder

select lines		Output			
A	B	F_0	F_1	F_2	F_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



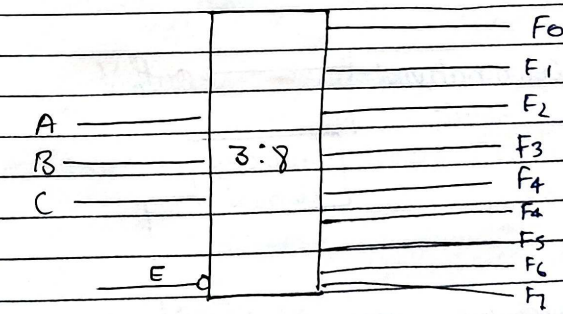
If they are selected through signal '0' then jaha jaha '1' hai vaha vaha 0 rakh denge or jaha 0 hai vaha 1.

Date: / / Page no:



A	B	F_0	F_1	F_2	F_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

3:8 Decoder



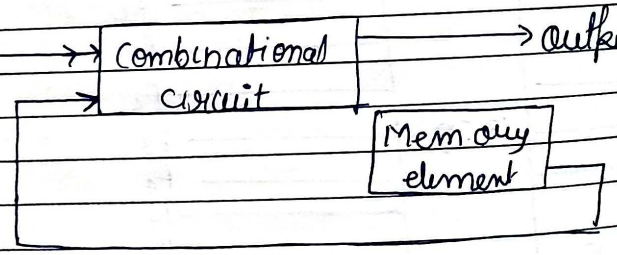
It is also known as binary to Octal decoder

Date: / / Page no:

	INPUT			OUTPUT							
E	A	B	C	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
1	x	x	x								
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	0	0	0	1

for making this enable at active low i.e '0' then put bubble in the block diagram and put '0' in place of 1 and also put 1 in place of 0

Introduction of Sequential Circuits



Block diagram of Sequential circuit

Date: / / Page no:

- State :- The element of memory element at any instant of time

⇒ Input + Present state = output

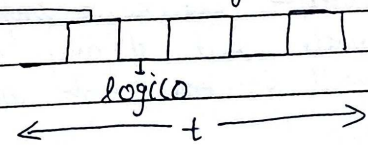
Sequential circuits are classified into two type

→ Asynchronous Sequential circuit = A Sequential circuit whose behaviour depends upon the sequence in which the input signals changes is referred to as 'asynchronous Sequential circuits'. The output will be effected whenever the input changes

→ Synchronous Sequential circuits = A Sequential circuit whose behaviour can be defined from the knowledge of its signal at discrete instants of time is refered to as 'Synchronous Sequential circuit'

⇒ Memory element in Asynchronous Sequential circuit is time delay devic also called gate type asynchronous system.
Asynchronous Sequential circuit is a combinational circuit with feedback

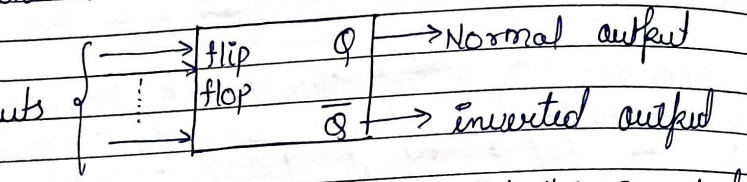
⇒ In Synchronous Sequential circuits we can make the use of pulse which define high as logic 1 and low as logic 0 as shown in figure



Date: / / Page no:

We can also use Master clock generator, it generates periodic train of clock pulse

Flip flop:- It is bistable multivibrator memory element which means it has two stable states



- When Q = 1, then we'll say flip flop is at high state or logic 1 or set
- When Q = 0, then we'll say flip flop is at low state or logic 0 or reset

⇒ Applications of flip flop
1) can be used in storage devices
2) can be used in counter, shift register

⇒ Latch = Non clocked flip flop are called latch
It lock any single binary bit into '1' or '0'

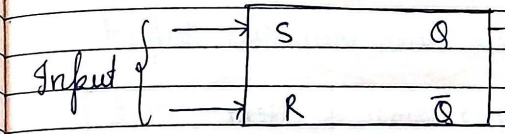
⇒ Gated latch = In the if enable signal is active/High then only the output can be store otherwise not. That is why enable is also called Gating signal

⇒ Active high input latch:- Set/Reset input will be only on low state but if any of them is changed then it will go on high state

Date: / / Page no:

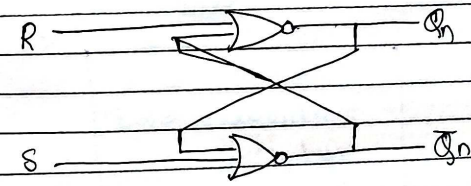
⇒ Active low Input Latch:- Generaly the set/Reset will remain on active high state and if any of set or Reset will be pulsed/changed then it'll goes on active low

SR Latch (set - Reset)



⇒ NOR SR Latch

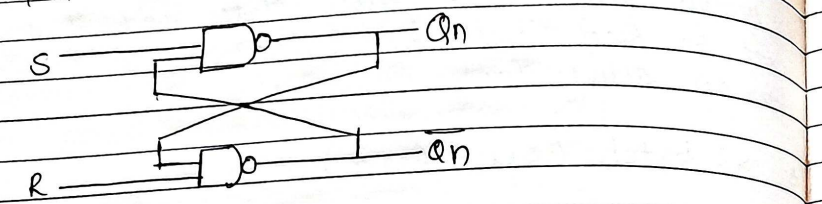
	S	R	Q_n	$\overline{Q_n}$	Q_{n+1}	$\overline{Q_{n+1}}$	
case1	0	0	0	1	0	1	No change
	0	0	1	0	1	0	
case2	0	1	0	1	0	1	Reset
	0	1	1	0	0	1	
case3	1	0	0	1	1	0	set
	1	0	1	0	1	0	
case4	1	1	0	1			forbidden state
	1	1	1	0			



depends on present input and previous output

Date: / / Page no:

⇒ NAND Gate SR LATCH



depends on present state and previous output

Present Input		Present state		Next state	
S	R	Qn	$\overline{Qn}$	Qn+1	$\overline{Qn+1}$
0	0	0	1	forbidden	
0	0	1	0	forbidden	
0	1	0	1	1	0 (Set)
0	1	1	0	1	0 (Set)
1	0	0	1	0	1 (Reset)
1	0	1	0	0	1 (Reset)
1	1	0	1	0	1 (No change)
1	1	1	0	1	0 (No change)

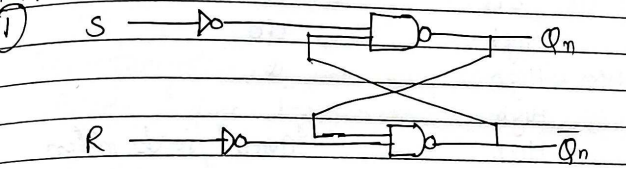
Short truth table

S	R	Comment
0	0	forbidden stat
0	1	Set
1	0	reset
1	1	No change

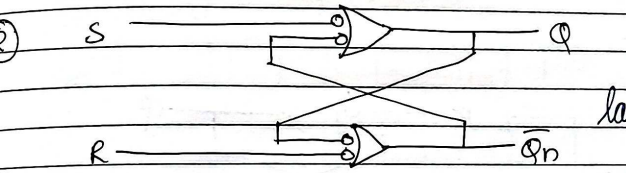
Date: / / Page no:

Converting this latch into OR latch ∴ High Nand Latch

Step:-

(1) 

This is active High Nand Latch

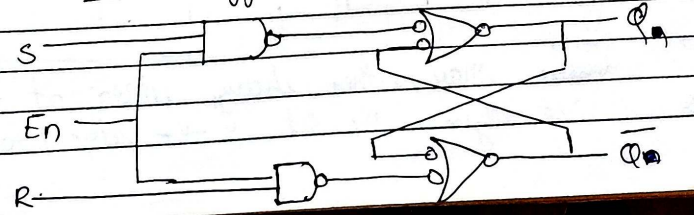
(2) 

This is active low OR Latch

Inputs		Outputs		Comments
S	R	Qn	$\overline{Qn}$	
0	0	Q0	$\overline{Q0}$	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	?	?	Forbidden

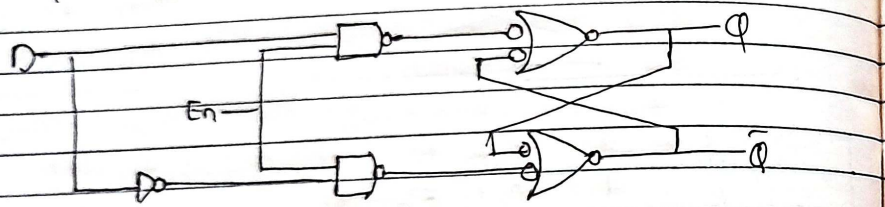
The Gated S-R Latch :- In this latch we required enable signals. Gated S-R Latch is also called clocked S-R Latch. This latch works only for active enable signal that is for logic 1 only.

When the level of clock signal is High then we call it level triggered flip flop



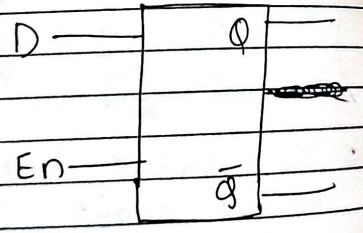
Inputs		Enable	Output
S	R	En	Q
0	0	High	Q_0
0	1	High	0
1	0	High	1
1	1	High	Invalid / forbidden

⇒ The Gated D- latch

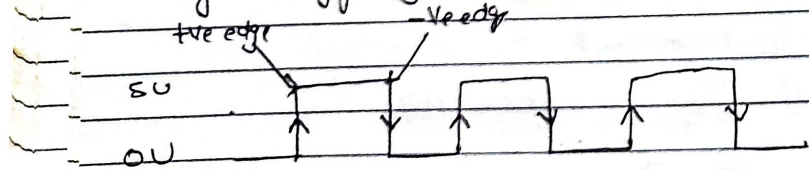


D represented data

Input	Enable	
D	En	Q
0	High	0
1	High	1



Edge triggering:-

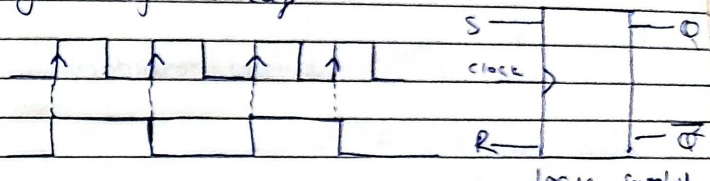


The State may either change either at +ve edge or rising edge or at -ve edge / falling edge

- positive transition of pulse is called +ve edge.
- negative transition of pulse is called -ve edge.

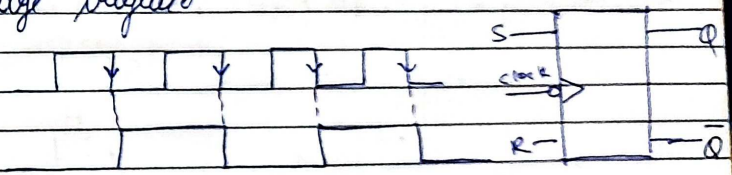
The transition by +ve edge or negative edge is called 'edge triggering'

⇒ clock cycle by +ve edge



Logic symbol

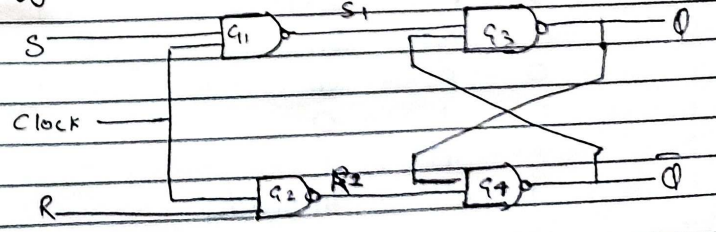
⇒ -ve edge triggered



There are three types of edge triggered flip flop

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

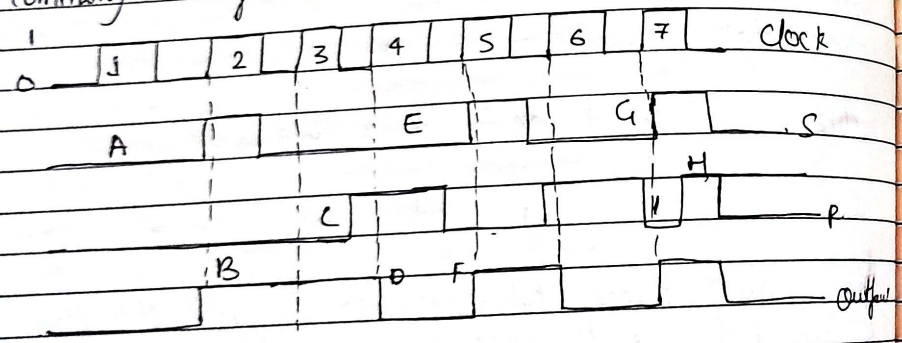
Triggered SR flip flop



- for Q_1, $S_1 = (\overline{S \cdot clock}) \Rightarrow S_1 = \overline{S} + \overline{clock}$
- for Q_2, $R_2 = (\overline{R \cdot clock}) \Rightarrow R_2 = \overline{R} + \overline{clock}$

clock	S	R	Q	$\bar{Q}$
0	X	X	No change	
1	0	0	No change	
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid / forbidden	

Timming diagram:-



⇒ Truth Table of SR flip flop

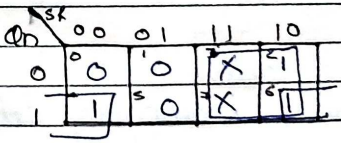
clock	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Invalid

⇒ characteristic table

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

⇒ Excitation table:-

Input		Output	
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0



⇒ $Q_{n+1} = Q_n\bar{R} + S$

Date: / / Page no:

Important Question

Ques1) Write a short note on weighted & Non weighted code

Sol: Weighted binary codes are those binary codes which obey the positional weight principal. Each position of the number represents a specific weights. Decimal system of the codes are used to express the decimal digit 0 through 9. In these codes each decimal digit is represented by a group of four bits. Example :- BCD codes, 8421 codes, 6421, 4221, 5211, 3321 etc

Non weighted code or un weighted codes are those codes in which the digit value does not depend upon their position i.e each digit position within the number is not assigned fixed value. Example :- Excess 3 code & gray code

Ques2) What are the uses of Excess 3 code?

Ans Excess -3 code is also known as Self complemantry because it can easily by complemented (9's compliment) to perform subtraction in the case of Subtraction

It is used to express code used to express decimal number. It is mainly used in arithmetic operation.

Date: / / Page no:

Ques3) What are the Uses of BCD code?

Ans
- It is useful for representation decimal number whose powers of two would give cumulative errors or for representing currency where 2 decimal places are only necessary.
- They are also used in alphanumeric displays where the data only need to be in the range 0 to 9, it is useful for storing two digit in one byte, when alpha ASCII characters are not required.
- It is a form of binary encoding where each digit in a decimal number is represented in the form of bits
- This encoding can be done in either 4 bit or 8 bit (usually 4 bit is prefered)
- It is fast and efficient System that converts the decimal number into binary number as compared to the existing binary System
- These are generally used in digit displays where is the manipulation of data is quite a task
- Thus BCD plays an important role here because the manipulation is done treating each digit as a separate single sub circuit

Ques 4) What are the uses of Gray code?

Ans
- The Gray code is used in the transmission of digital signal as it minimize the occurance of error
- The Gray code is prefered over the straight

binary code in angle measuring devices the of the Gray code almost eliminates the possibility of an angle misread, which is likely if the angle is represented in straight binary. The cyclic property of the gray code is a plus in this application

- The Gray code is used for labelling the axes of Karnaugh maps, a graphical technique used for minimization of Boolean expression.
- The use of gray code to address program memory in computer minimizes power consumption this is due to fewer address lines changing state with advances in the program counter
- Gray codes are also very useful in genetic algorithm since mutation in the code allow for mostly incremental changes. However occassionally a one bit change can result in a big leap, thus leading to a new properties.
- These codes are precisely used in electro optical switches and electrochemical signals
- These codes are also use in the conversion from analog to digital format.
- Use of this code is locating for rotational position of the shafts

Ques 5) What are the diff. b/w the sequential & combination circuit

Combinational Circuit

1) In this output depends only upon present input
2) Speed is fast
3) It is designed easy
4) There is no fedback b/w input and output
5) This is time independent
6) Elementary building blocks: Logic gates
7) Used for arithmetic as well as boolean operation
8) Combinational circuits don't have capability to store any state
9) As combinational circuits don't have clock, they don't require Triggering
10) These circuits do not have any memory element
11) It is easy to use and handle
12) Example - Encoder, Decoder, Multiplexer & Demultiplexer

Sequential Circuit

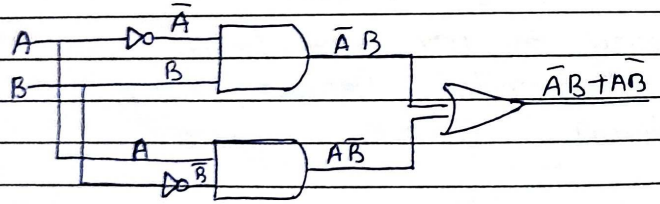
1) In this output depend upon present as well as past input
2) Speed is slow
3) It is designed tough as compared to combinational circuit
4) There exists a fedback path b/w input and output
5) This is time dependent

6) Elementary building block : flip-flops
7) Mainly used for storing data
8) Sequential circuit have capability to store any state or to retain earlier state
9) As sequential circuit are clock dependent they need triggering
10) These circuit have memory element
11) It is not easy to use and handle
12) Examples:- flip-flops, Counter

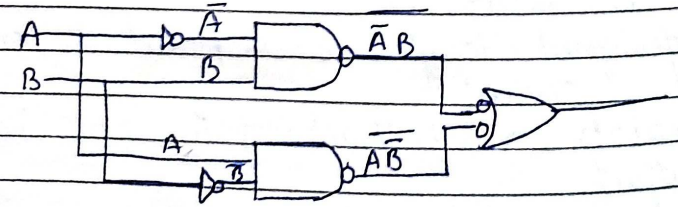
Ques6) Implement the following equation in NAND & NOR Gate $Y = \bar{A}B + A\bar{B}$

Sol: ① Implementing $Y = \bar{A}B + A\bar{B}$ in NAND Gate

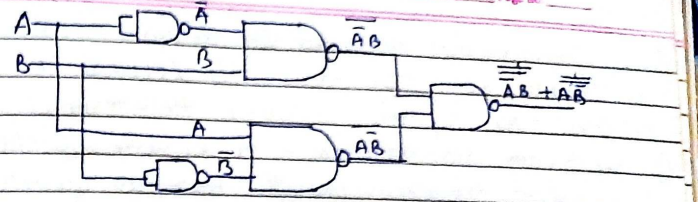
Step1:- Draw the basic logic diagram



Step 2:- Apply bubble at the output of And Gate and Input of OR Gate.



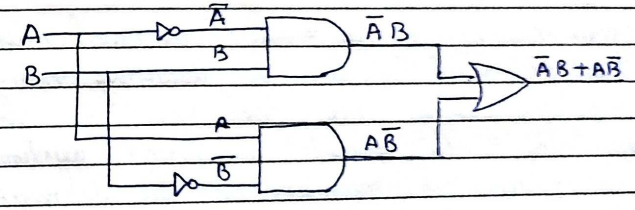
Step 3:-



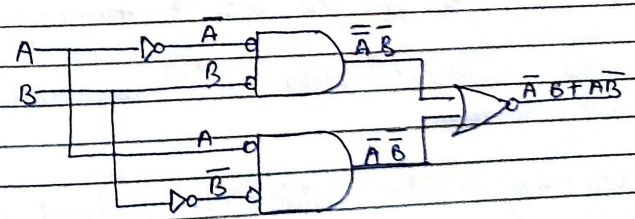
Hence this is the implementation of Boolean eq' in NAND Gate

② Implementing $Y = \bar{A}B + A\bar{B}$ in NOR Gate

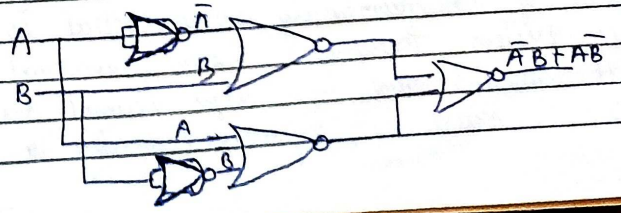
Step 1 :- Draw the Basic logic diagram



Step 2 :- Apply bubble at the output of NOR Gate and input of AND Gate



Step 3:-



Date / / Page no:

Quest) Explain the following in details
i) Binary No. System (ii) Decimal No. System
iii) Octal No. System (iv) Hexadecimal No. system

Ans i) Binary No. System = A binary number is a number expressed in the base 2 numeral system or binary numeral system, a method of mathematical expression which uses only two symbols typically "0" and "1". The base 2 numeral system is a positional notation with a radix of 2 each digit is referred to as a bit, or binary digit.

ii) Decimal number System = The decimal number system is the standard system for denoting integer and non-integer. It is the extension to non integer number of Hindu Arabic numeral system. The way of denoting number in the decimal system is often refered to as decimal notation. Base = 10

iii) Octal number System :- The octal numeral system or oct for short is the base 8 number system and uses the digit 0 to 7, that is to say 10 octal represent eight and 100 octal represent 60 four

iv) Hexadecimal number System = Hexadecimal is the name of the numbering system that is base 16. This system, therefore, has numeral from 0 to 15 that means that two digit decimal numbers 10-15 must be represented by a single numeral to exist in the numbering system.

Quest) Explain the terms 'SOP' & 'POS'

Ans SOP = SOP stands for Sum of products. The Sum of product expression comes from the fact that two or more product (AND) are summed (OR) together. That is the output from two or more AND gates are connected to the input of an OR gate so that they are effectively ORied together to create the final AND-OR logically output

POS = POS stands for Product of Sum. The Product of Sum expression comes from the fact that two or more sums (OR) are added (AND) together. That is the output from two or more OR gates are connected to the input of an AND gate so that they are effectively AND'ed together to create the final OR-AND output

Quest) Convert SR flip flop to D flip flop

Soln Truth Table for D flip flop

Step 1:-

D	Qn	Qn+1
0	X	0
1	X	1

Excitation table for SR flip flop

Date: / / Page no:

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

Truth Table of D flip flop → Excitation table of SR flip flop

	D	Q_n	Q_{n+1}	S	R
0 —	0	0	0	0	X
1 —	0	1	0	0	1
2 —	1	0	1	1	0
3 —	1	1	1	X	0

conversion table.

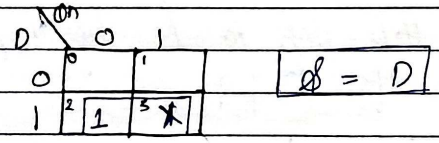
Now we'll mak K map :-

[NOTE:- Q_{n+1} is an output]

step 2:-

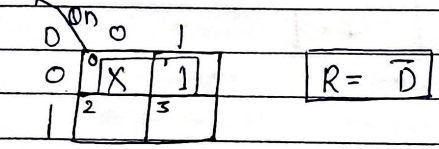
- K map for S

:



$S = D$

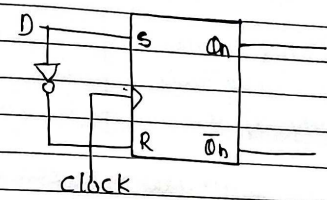
- Kmap for R



$R = \bar{D}$

Date: / / Page no:

step 3:- Now we are going to draw the circuits



circuit diagram

Ques 10) Write the diff. b/w Synchronous & Asynchronous logic circuit

Synchronous sequential circuit	Asynchronous sequential circuit
1) These circuit are easy to design	1) These circuits are difficult to design
2) A clocked flip flop acts as memory element	2) An unclocked flip flop or time delay element is used as memory element
3) They are slower	3) faster as clock is Not present
4) The states of memory element (i.e flip flop) is affected only at the active edge of clock, if input is changed	4) The states of memory element will change any time as soon as input is changed
5) flip flop are use in Synchronous sequential circuit	5) Latches are used in Asynchronous sequential circuit.

Date: / / Page no:

Ques 11) Diff b/w Analouge Signals and digital Signa

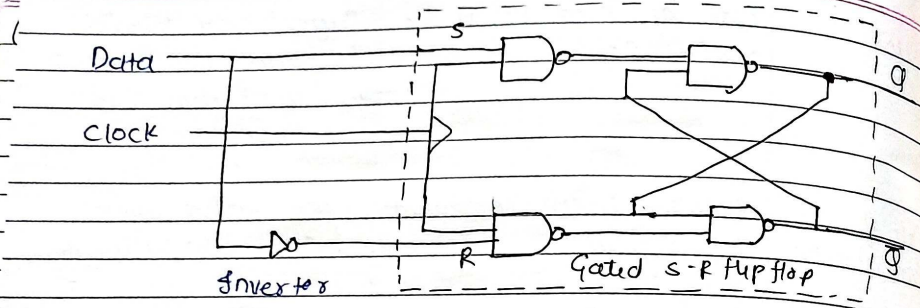
Characteristics	Analog signal	Digital signal
Adaptability	Are less adaptable to variety of use	Are more adaptable to variety of use
continuity	Takes on continuous range of amplitude wave	Takes on finite set of discrete values at uniform space point in the time
Type of data	continuous in nature	Discrete in nature
wave type	Sin waves	Square wave
Medium of Transmission	wire / wireless	wire
Type of values	positive / Negative	positive values only
Security	Non encryption	encryption
= Power consumption	Analog instrument consume large amount of power	Digital instrument consume very small amount of power
Recording data	Records sound waves as they are	converts into a binary waveform
Capacity / Bandwidth	Low	High
use	can be used in analog device solely	suitable for Digital electronic device such as computers and cell phone
Rate of transmission	slower when compared to digital	Much faster with better productivity
Applications	Thermometer	PDA, PCs
Examples	Human voice, Thermometer, Analog phones	computer, Digital phones, Digital pens

Ques 12) What is D flip flop?

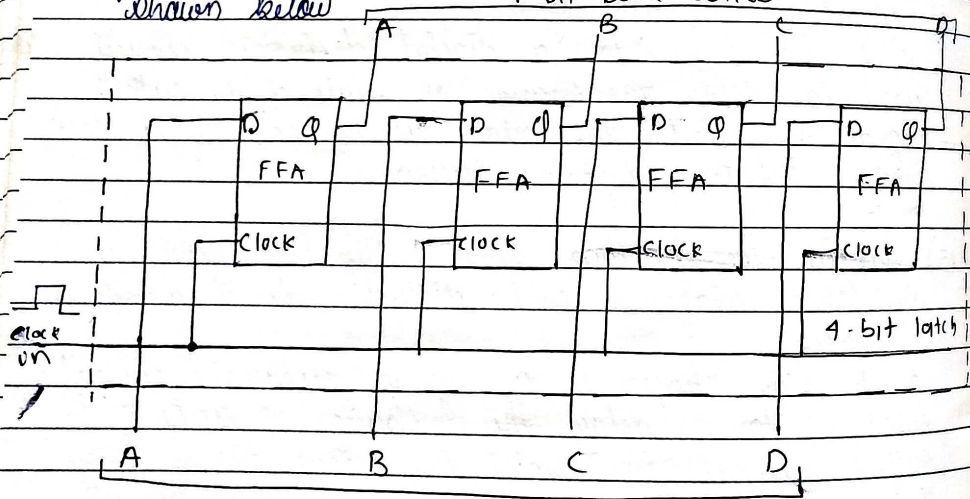
A D flip flop is a digital electronic circuit used to delay the change of state of its output signal (Q) until the next rising edge of a clock timing inputs signals occurs.

Ques 13) Explain the function of a D flip flop using a suitable diagram and discuss how it works as a latch.

A D flip flop is a digital electronic circuit used to delay the change of state of its output signal (Q) until the next rising edge of a clock timming input signal occurs



By connecting together four, 1-bit data latches so that all their clock inputs are connected together and are clocked at the same time, a simple "4 bit" Data Latch can be made as shown below



Ques 14) Convert J-K flip flop to D flip flop ?

Soln NOTE - for conversion of any flip flop you need truth table of destination flip flop and excitation table of source flip flop

Truth table of D flip flop

D	Q_n	Q_{n+1}
0	X	0
1	X	1

Excitation table of J K flip flop

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Truth table of D Flip Flop ∫ Excitation table of J K flip flop

D	Q_n	Q_{n+1}	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

conversion table

⇒ kmap for J :-

D \ Q_n	0	1
0	0	'X
1	2 1	3 X

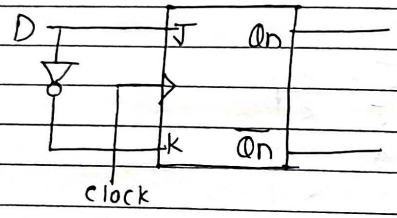
J = D

Date: / / Page no:

⇒ k map for D :-

D \ Qn	0	1
0	0 X	1 1
1	2 X	3

$K = \overline{D}$



Ques 15) Convert SR flip flop to JK flip flop

Truth table of JK flip flop

J	K	Qn	Qn+1
0	0	X	Qn
0	1	X	0
1	0	X	1
1	1	X	$\overline{Qn}$

Excitation table of SR flip flop

Qn	Qn+1	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Date: / / Page no:

Truth table of JK Flip Flop — excitation table of SR flip flop

	J	K	Qn	Qn+1	S	R
0	0	0	0	0	0	X
1	0	0	1	1	X	0
2	0	1	0	0	0	X
3	0	1	1	0	0	1
4	1	0	0	1	1	0
5	1	0	1	1	X	0
6	1	1	0	1	1	0
7	1	1	1	0	0	1

⇒ kmap for S :-

Qn \ JK	00	01	11	10
0	0	1 X	3	2
1	4 1	5 X	7 0	6 1

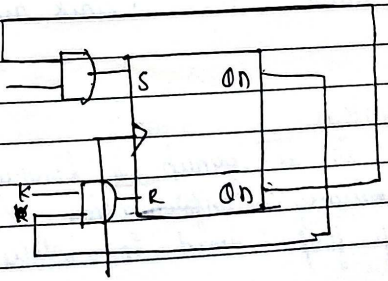
S = ~~Qn J + Qn K~~ ~~Qn~~

S = ~~Qn~~ $(\overline{J}+K)$ $\overline{Qn}J$

⇒ K map for R

Qn \ JK	00	01	11	10
0	0 X	1	3 1	2 X
1	4	5	7 1	6

R = Qn K



Date: / / Page no:

Ques16) What is J-K flip flop?

Ans The JK flip flop is basically a Gated SR flip flop with the addition of a clock input circuit that prevents the illegal or invalid output condition that can occur when both input S and R are equal to logic level "1".

A J-K flip flop is called an universal flip flop because it can be configured to work as an SR flip-flop, D-flip flop or T flip flop.

Ques17) which circuit is used in Traffic signal?

Ans Coming to the working principle of Traffic lights the main IC is 4017 counter IC which is used to glow the Red, Yellow and green LED respectively. 555 timer acts as a pulse generator providing an input to the 4017 counter IC

Ques18) Define Counter

Ans Counter is a device which stores (and sometime displays) the number of times a particular event or process has occured often in relationship to a clock. The most common type is a sequential digital logic circuit with an input line called the clock and multiple output line.

Ques19) Define Registers

Ans A register is a group of binary cells suitable for holding binary information. A group of cascaded flip-flops used to store related bits of information is known as registers. Registers are a type of a computer memory used to quickly accept, store and transfer data and instruction that are being used immediately by the CPU. The register used by the CPU are often termed as prodssor register

Ques20) Convert Excess 3 code as decimal

1) $(0110\ 0111\ 1000)_{Ex-3} = (?)_{10}$

Soln
0110 0111 1000
6 7 8
-3 -3 -3
3 4 5 $\Rightarrow (345)_{10}$

2) $(0110\ 1011\ 1100\ 0111)_{XS3} = (?)_{10}$

Soln
0110 1011 1100 0111
6 11 12 7
-3 -3 -3 -3
3 8 9 4 $\Rightarrow (3894)_{10}$

Ques21) Convert Decimal to BCD

1) $(345)_{10} = (?)_{BCD}$

Soln
3 4 5
↓ ↓ ↓
0011 0100 0101 $\Rightarrow (001101000101)_{BCD}$

2) $(26)_{10} = (?)_{BCD}$

Soln
2 6
↓ ↓
0010 0110 $\Rightarrow (00100110)_{BCD}$

3) $(2019)_{10} = (?)_{BCD}$

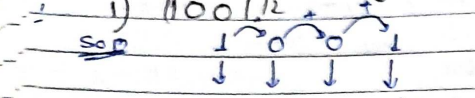
2 0 1 9
↓ ↓ ↓ ↓
0010 0000 0001 1001

$(0010\ 0000\ 0001\ 1001)_{BCD}$

Ques22) Convert Binary to gray

1) $(1001)_2$

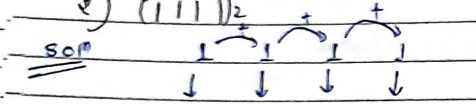
Sol:



(1 1 0 1) Gray

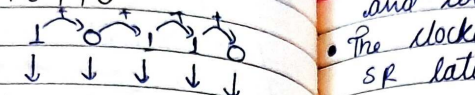
2) $(1111)_2$

Sol:



(1 0 0 0) Gray

3) 10110



(1 1 1 0 1) Gray

4) 1 1 1 0 1 1 1

1 0 0 1 1 0 0

Ques23) Master slave flip flop. Explain?

Sol: Master slave flip flop is a type of clocked flip flop consisting of master and slave element that are clocked on complementary transition of the clock signal

Firstly the master flip flop is positive level triggered flip flop and the slave flip flop is negative level triggered, so the master responds before the slave. If J=0 and K=1 the high Q' output of the master goes to the K input of the slave and the clock forces the slave to reset, thus the slave copies the master

Two similar flip flop in sequence. The first is called Master, the second is called the slave because he follows the action of the Master. And when the master is awake, the slave is sleeping and vice versa. If the master is triggered by +ve edge of the clock, the slave on the -ve edge of the same clock. So, they never meet and collide with each other

- The clocked JK latch acts as the master and the clocked SR latch acts as the slave

NOTE :- The JK flip flop name has been kept on the inventors name of the circuit known as Jack Kilby

⇒ T flip flop stands for 'toggle' flip flop.

Ques24) What is the uses of flip flop

Ans: Flip flops are used to design Registers

USE of SR FLIP FLOP

→ It is used to keep record of different values of variable state like intermediate, input or output

→ The SR flip-flop is very effective in removing the effects of switch bounce, which is the unwanted noise caused during the switching of electronics devices

USE of JK FLIP FLOP

→ It is widely used in shift registers, counters, PWM and computer applications

USE of T FLIP FLOP

→ It is used in counter design

→ These flip flop are used for constructing binary counters

→ They are used in frequency dividers

→ This types of sequential circuits is also present in binary addition devices

Synchronous data transfer	Asynchronous Data Transfer
1) Two unit shares a common clock	1) Two units are independent & each have its own clock
2) Data transfer b/w sender & receiver is synchronized with same clock pulse	2) Data trans. b/w sender & receiver is not synchronized with same clock pulse
3) Used b/w devices that matches in speed	3) Used b/w devices that do not matches in speed
4) Bits are transmitted continuously to keep the frequency synchronous in both units	4) Bits are sent only when it is available & line remains idle when there is no info. to be transmitted
5) Synchronous means 'at the same time' (due to common clock)	5) Assynchronous mean "regular interval" (due to no common clock)
6) fast	6) slow
7) costly	7) economical

⇒ Asynchronous Data transfer b/w two independent unit requires control signals to be transmitted b/w communicating units to indicate the line at which data is to be transmitted

There are two types of method used in Asynchronous

1) Strobe control 2) Handshaking method

1) Strobe control = In this method Sender sends the strobe pulse to the receiver which informs the receiver that the Sender is going to send the data transfer.

There is a drawback of this method i.e when Sender sends strobe pulse signal to receiver to inform that the data transfer is about to start but in return the receiver don't send any kind of signal.

2) Handshaking method = In this method the Sender tells the receiver by sending signal that the data transfer is about to start in return the receiver also send the signal by which the Sender understand that receiving device is ~~ready for taking data~~ received the data

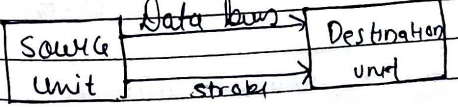
S ⟷ R

Strobe control =

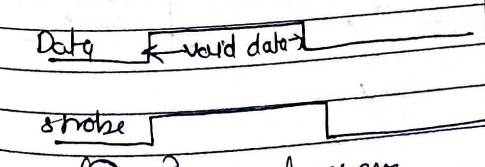
⇒ It employs a single control line to time each transfer

⇒ The strobe may be activated by either Source or Destination

Source initiate strobe for data transfer

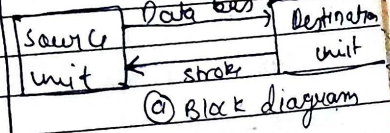


(a) Block diagram

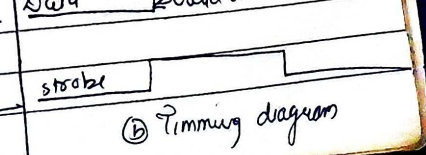


(b) Timming diagram

Destination - initiated strobe for data transfer



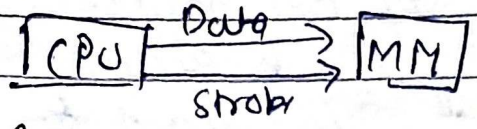
(a) Block diagram



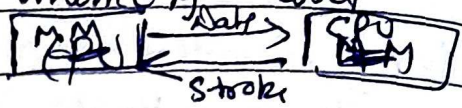
(b) Timming diagram

Date: / / Page no:

eg :- memory write control signal from CPU to memory unit



eg - memory read control signal from CPU to memory unit



⇒ Disadvantage of Strobe control

1) Source unit that initiates the transfer has no way of knowing whether the destination unit has actually received the data item that was placed in the bus.

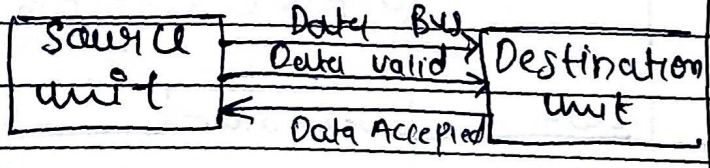
2) Similarly, Destination unit that initiates the transfer has no way of knowing whether the source unit has actually placed the data on the bus

Handshaking Method

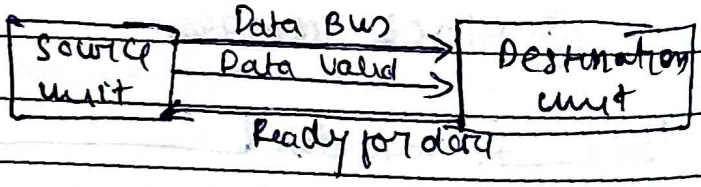
⇒ It solves the problem of strobe method by introducing a second control signal that provides a reply to the unit that initiates the transfer

⇒ Strobe control + Acknowledgement signal [Two wire control]

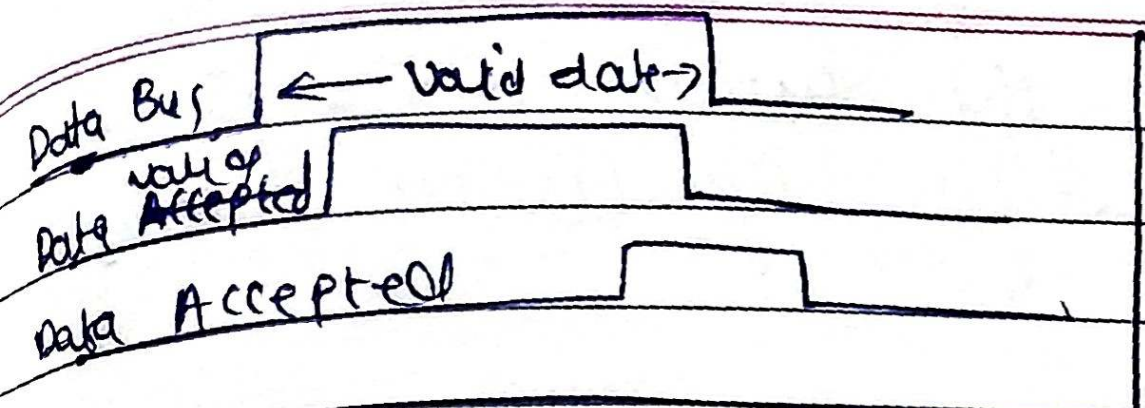
Source initiated

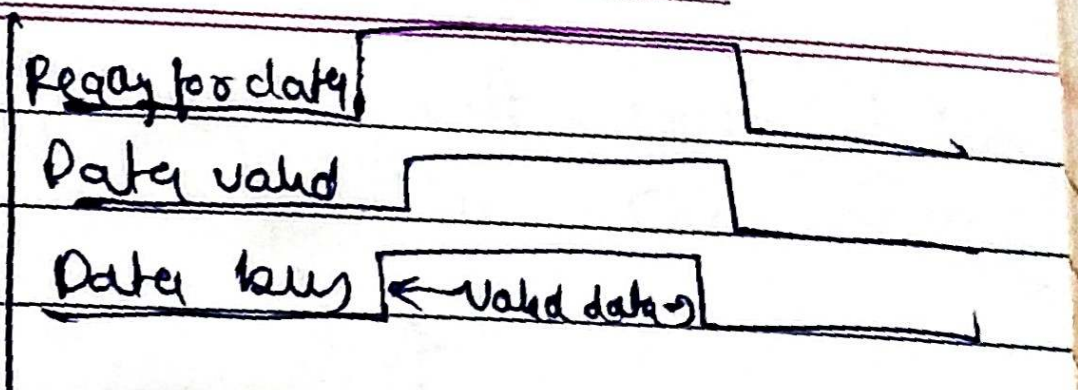


Destination initiated



Date: / / Page no:





Advantage

1) It provides a high degree of flexibility & reliability
2) If one unit is faulty, the data transfer is not completed such an error can be detected by time out mechanism which produces an alarm if the data transfer is completed within a predetermined time.