Chapter 2 Digital Design Fundamentals

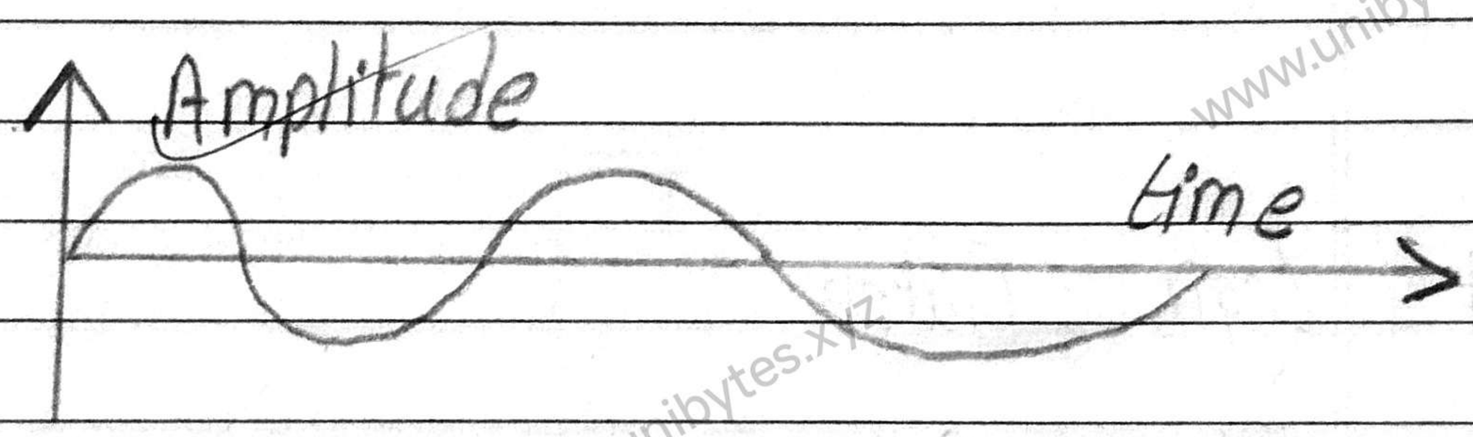
Along (Analog) System

Along (Analog) System is a continous signal which represent the physical quantities. It uses continous range of the values to represent information.

For examples:

i) The temperature of an environment,
ii) Voice in the air
iii) Bike speed

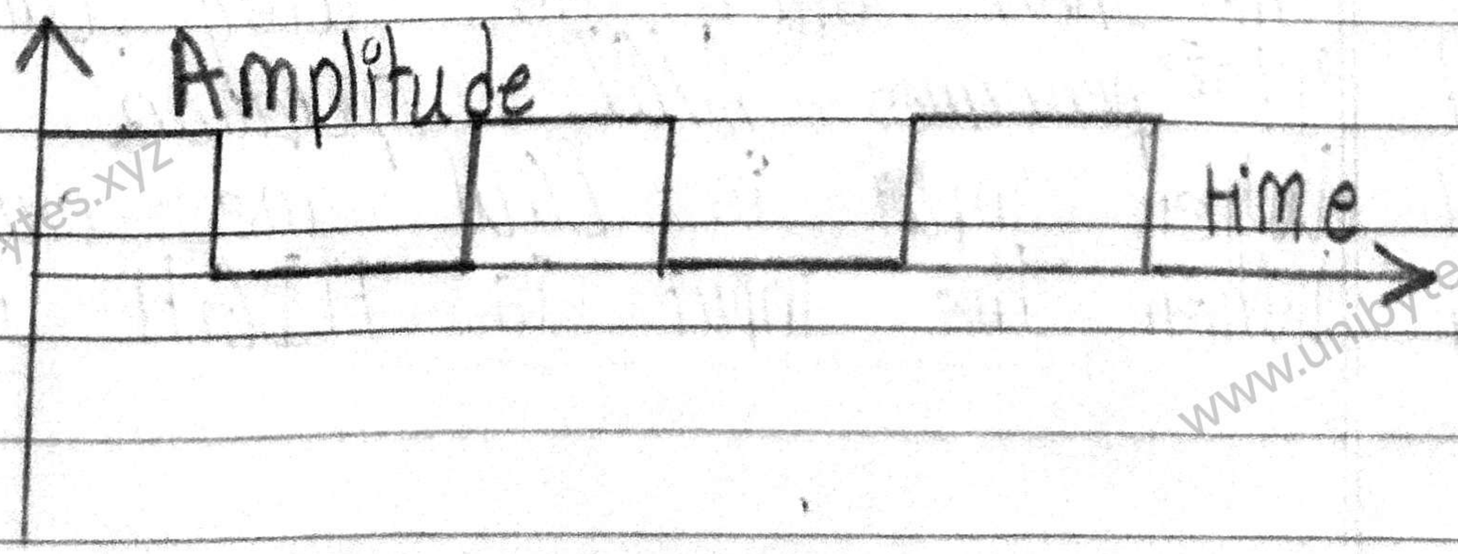
Wave form :



Digital system

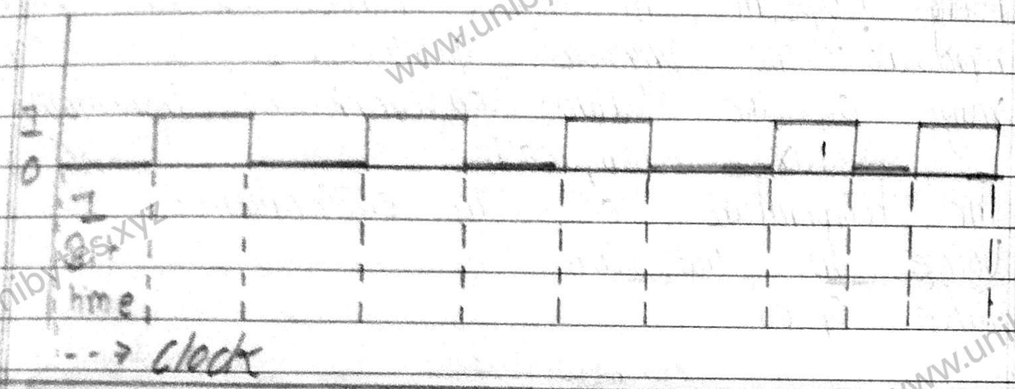
i) Digital signals are the ~~direct~~ discrete time signals that is generated modulation.
ii) This is denoted by square wave.
iii) This uses discret values to represent information.
iv) The examples of digital system are digital computer, electronic devices, digital devices, etc.

Wave form :



Clock

A clock is the periodic wave form in which each interval between the ~~pluses~~ pulses (the period) equals the time for one bit.



Logic Gates

* Basic Gates

A logic gate is a device that act as building block for digital circuits. They perform basic logical functions that are fundamental to digital circuits.

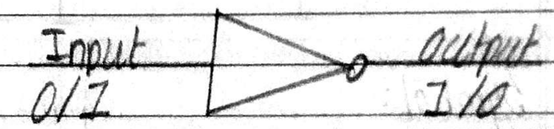
1) NOT Gate (Inverter):

i) The NOT Gate is also called inverter.
ii) It performes NOT operation.
iii) When input is LOW the output is HIGH
iv) When the input is HIGH the output is LOW

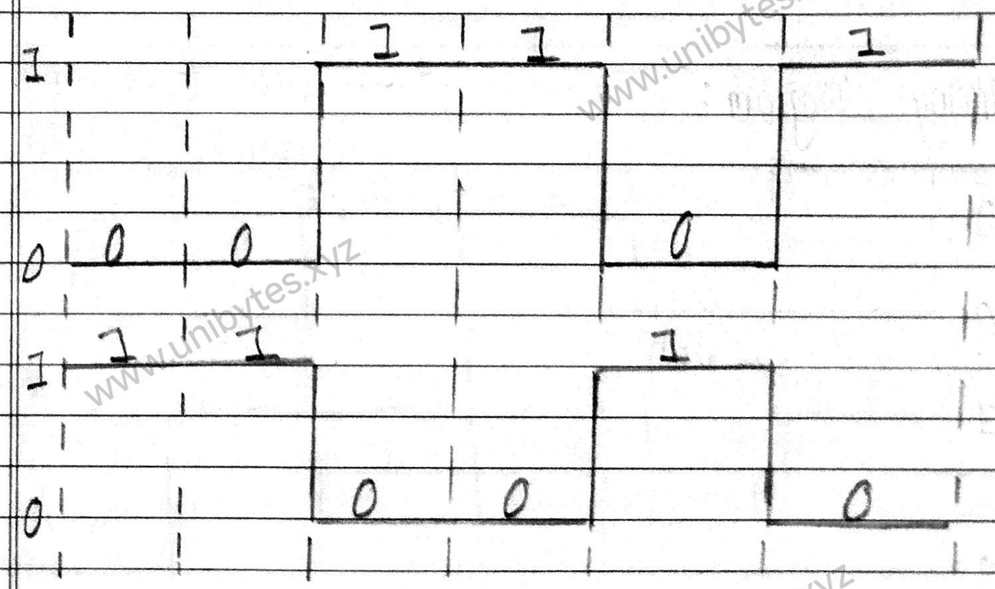
Truth table:

Input	Output
0	1
1	0

Symbolic Diagram:



Waveform of Inverter:



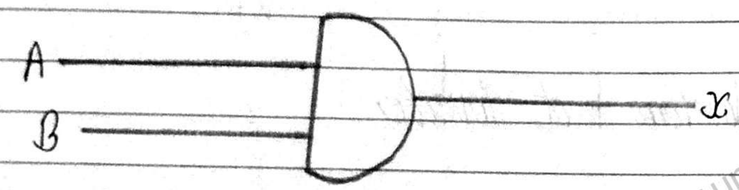
2) AND Gate

i) The AND Gate produces HIGH output when all the inputs are HIGH.
ii) Otherwise the output is LOW.

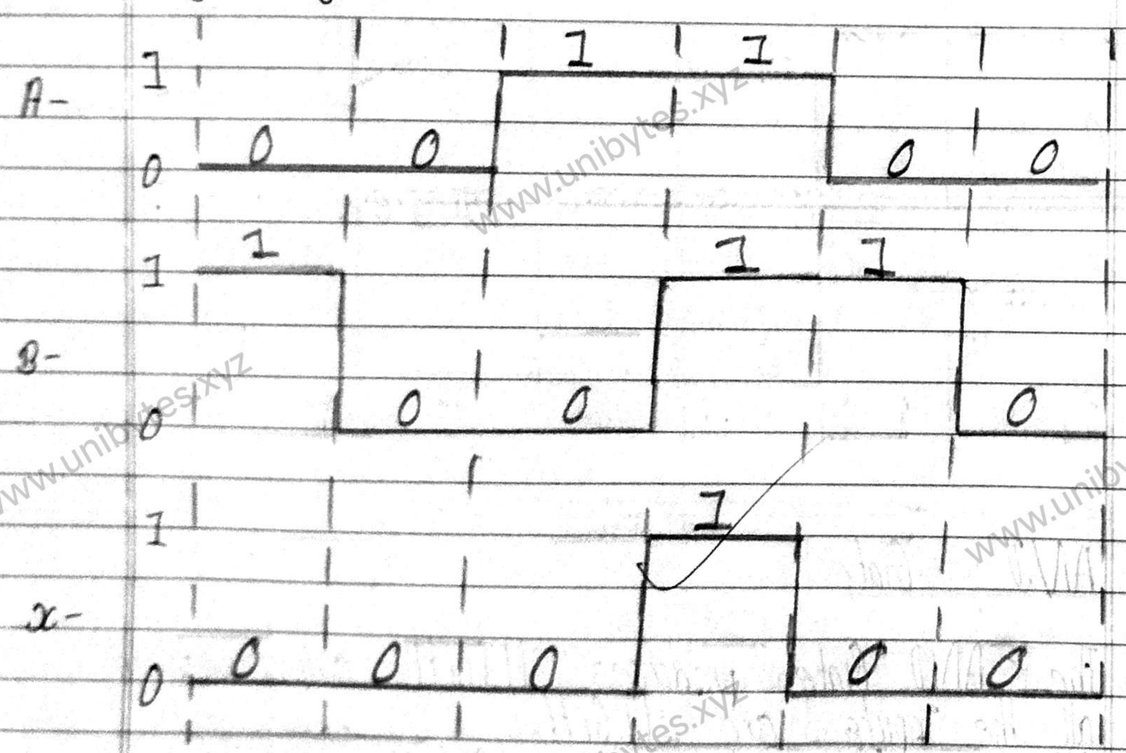
Truth table:

Input		Output
A	B	X
0	1	0
0	0	0
1	0	0
1	1	1

Symbol:



Timing Diagram:



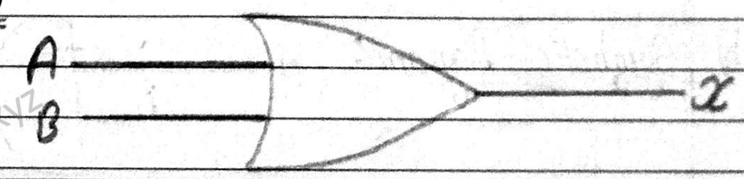
iii) The OR Gate

The OR gate produces a HIGH output if any input is HIGH; if all input are LOW, the output is LOW.

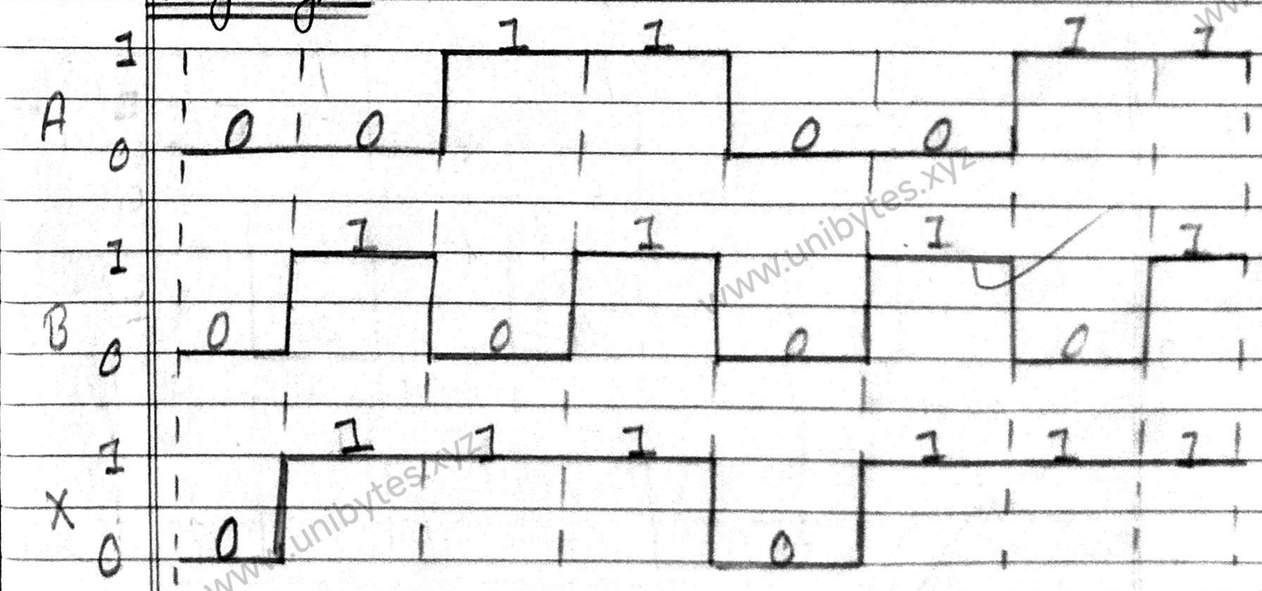
Truth table:

Inputs		Outputs
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Symbolic Diagram



Timing Diagram



* Derived Gates

The gates which are created by using basic gates are called derived gates.

i) NAND Gate

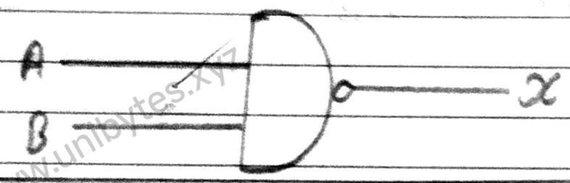
→ It produce low output when all inputs are HIGH otherwise:- It produce ~~low~~ high output.

a) Truth table :

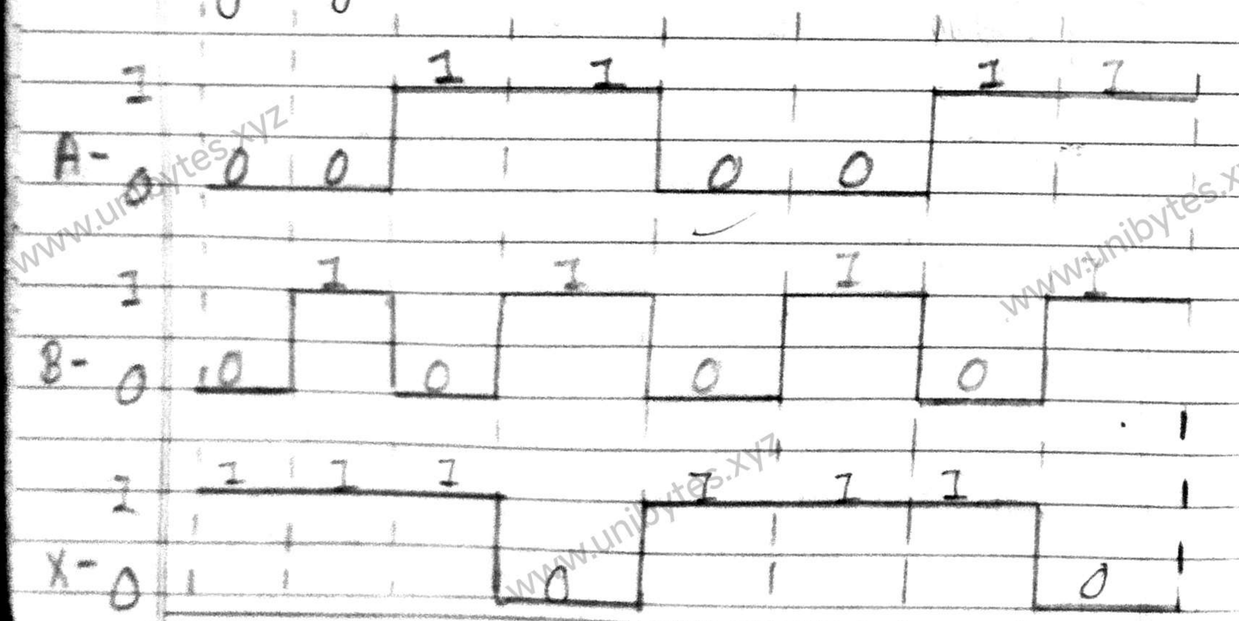
$X = (A \cdot B)'$

Inputs		Outputs
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

b) Symbolic Diagram :



c) Timing Diagram



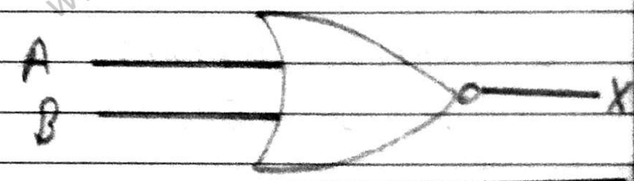
ii) NOR Gate

The NOR Gate produces a LOW output if any input is HIGH. If all input are HIGH the output is LOW. The operation of NOR gate is $X = (A+B)'$

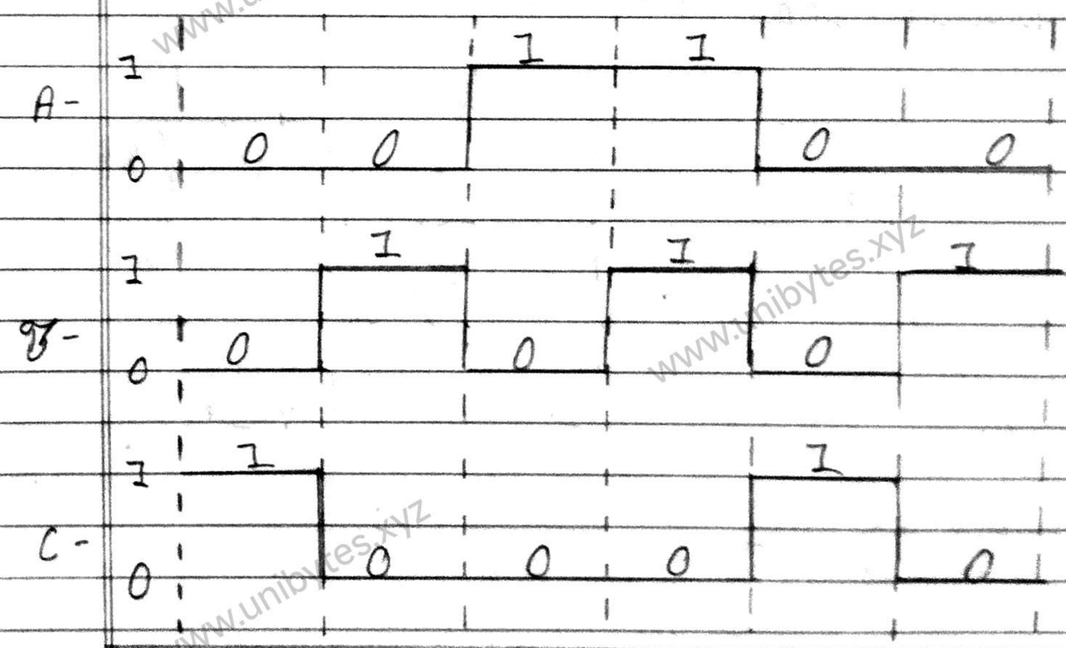
a) Truth table :

Inputs		Outputs
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

b) Symbolic Diagram :



c) Timing Diagram :



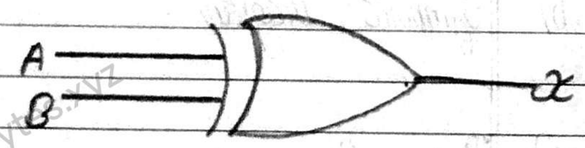
iii) The XOR Gate

The XOR gate produces a HIGH output only when both inputs are at opposite logic levels. The operation XOR written as $x = A'B + AB'$. Alternative it write as $A \oplus B$.

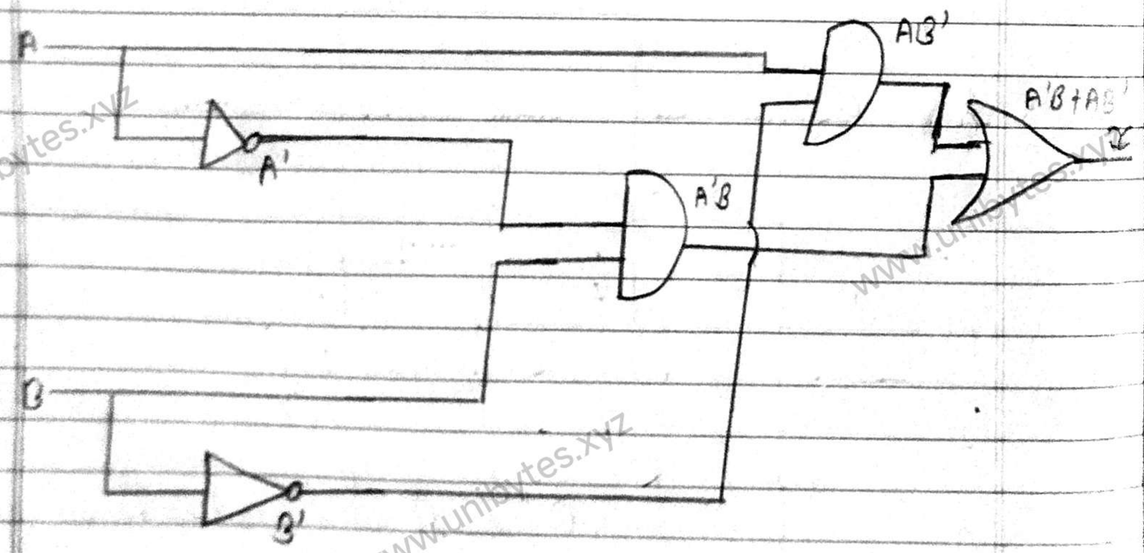
a) Truth table:

Input		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

b) Symbolic Diagram:



c) Equivalent Symbol $(A'B + AB')$



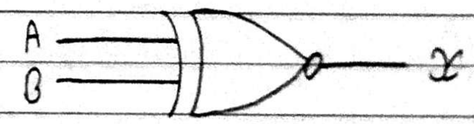
iv) The XNOR Gate:

The XNOR gate produces a HIGH output only when both inputs are at the same logic level. The XNOR operation shown as $X = A'B' + AB$.

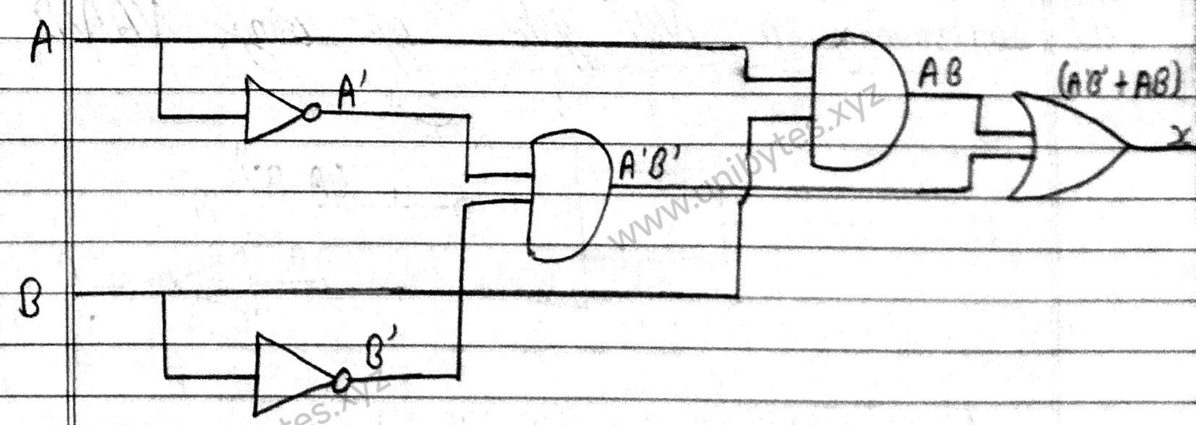
a) Truth Table:

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

b) Symbolic Diagram:



c) Equivalent Symbol $(A'B' + AB)$



www.unibytes.xyz

Date
Page

- Gate
  - Basic
    - NOT
    - AND
    - OR
  - Derived
    - NAND
    - NOR
    - XOR
    - XNOR
  - Universal
    - NAND
    - NOR

* Universal Gate :

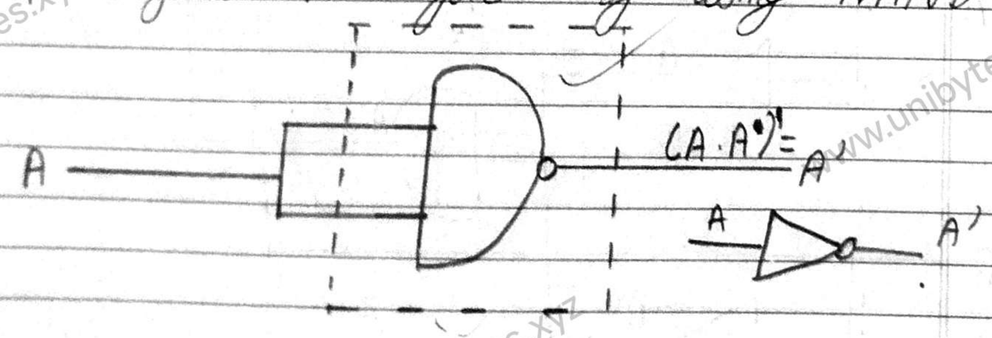
A Universal Gate is a gate which can ~~impor~~ implement any boolean function with out need of any other gate type.

There are two Universal Gates :

i) NAND
ii) NOR

* NAND Gate as a universal Gate.

a) Implementing a NOT gate by using NAND

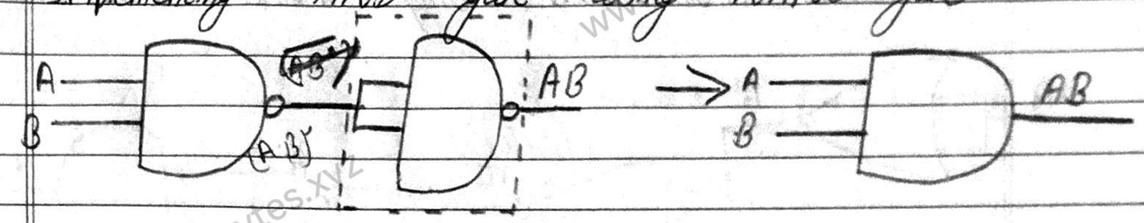


(Equivalent to NOT gate)

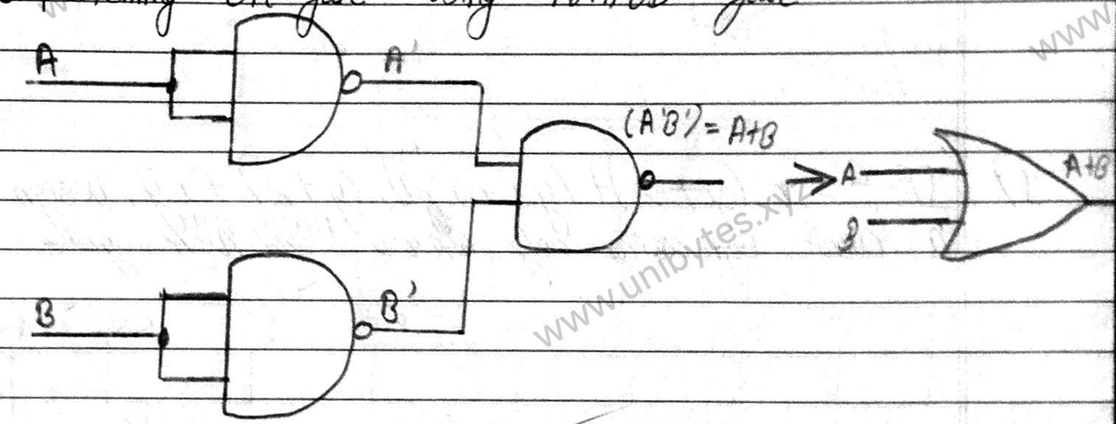
www.unibytes.xyz

Date
Page

ii) Implementing AND gate using NAND gate



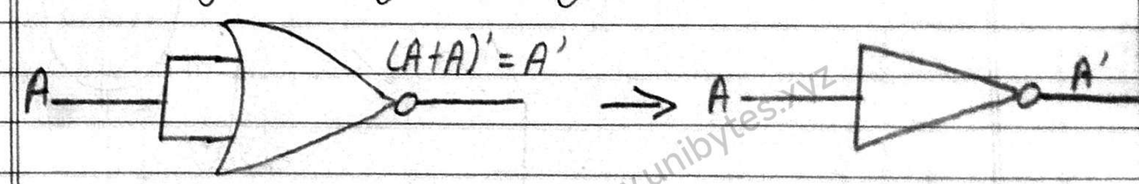
iii) Implementing OR gate using NAND gate



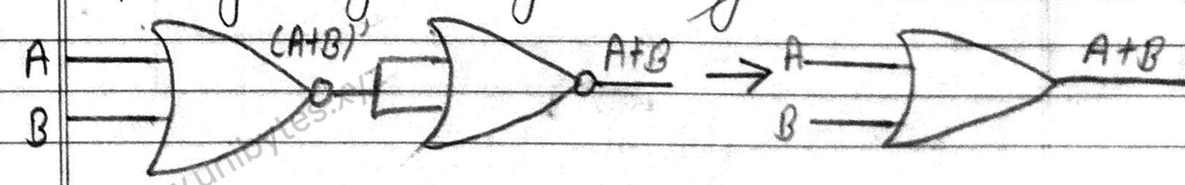
~~iv) Implementing~~

* NOR Gate is as Universal Gate

i) Implementing NOT gate using NOR Gate



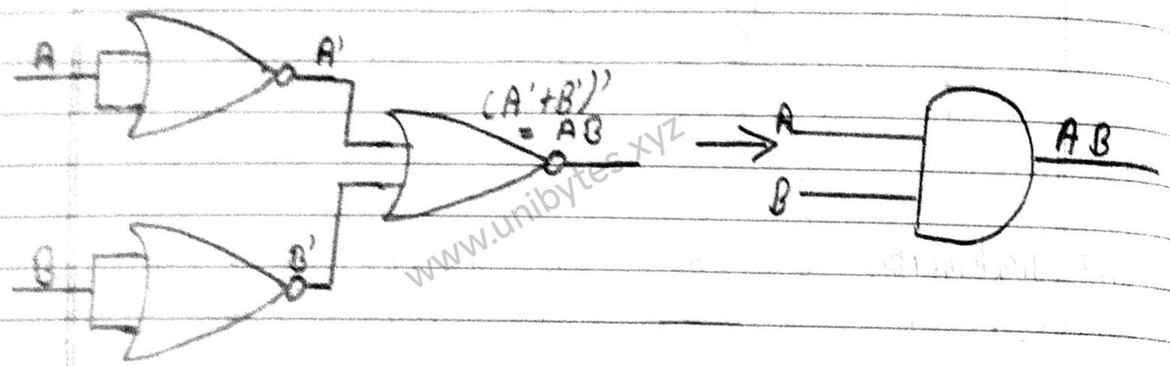
ii) Implementing OR gate using NOR gate



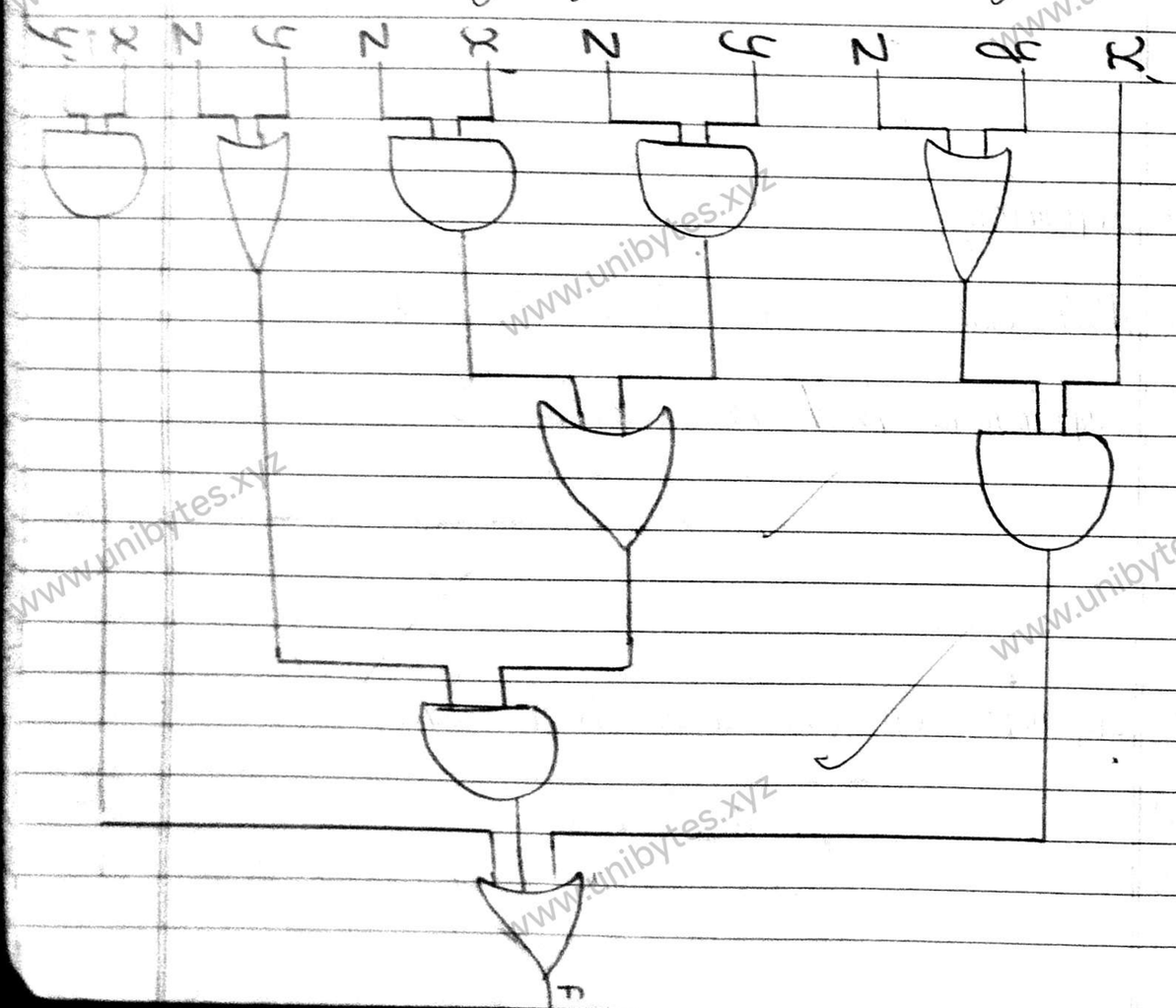
Mathworks
5th NOD

Date
Page

Implementing AND Using only NOR Gates

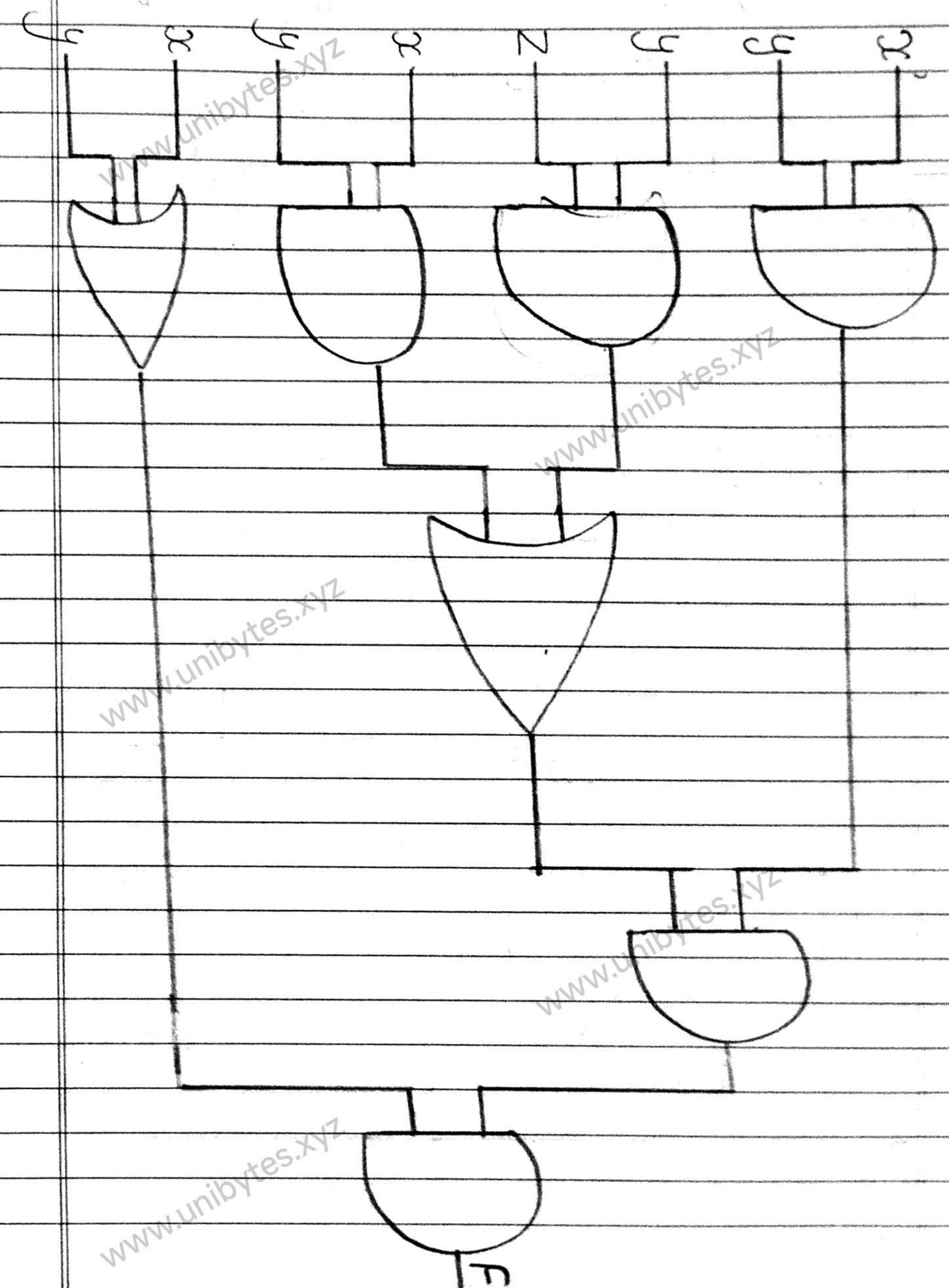


1) If $f = x'(y+z) + (yz + x'y).(y+z) + x\bar{y}$. Design a circuit ~~sign~~ using only ~~NAND and NOR~~ Basic gate



Date
Page

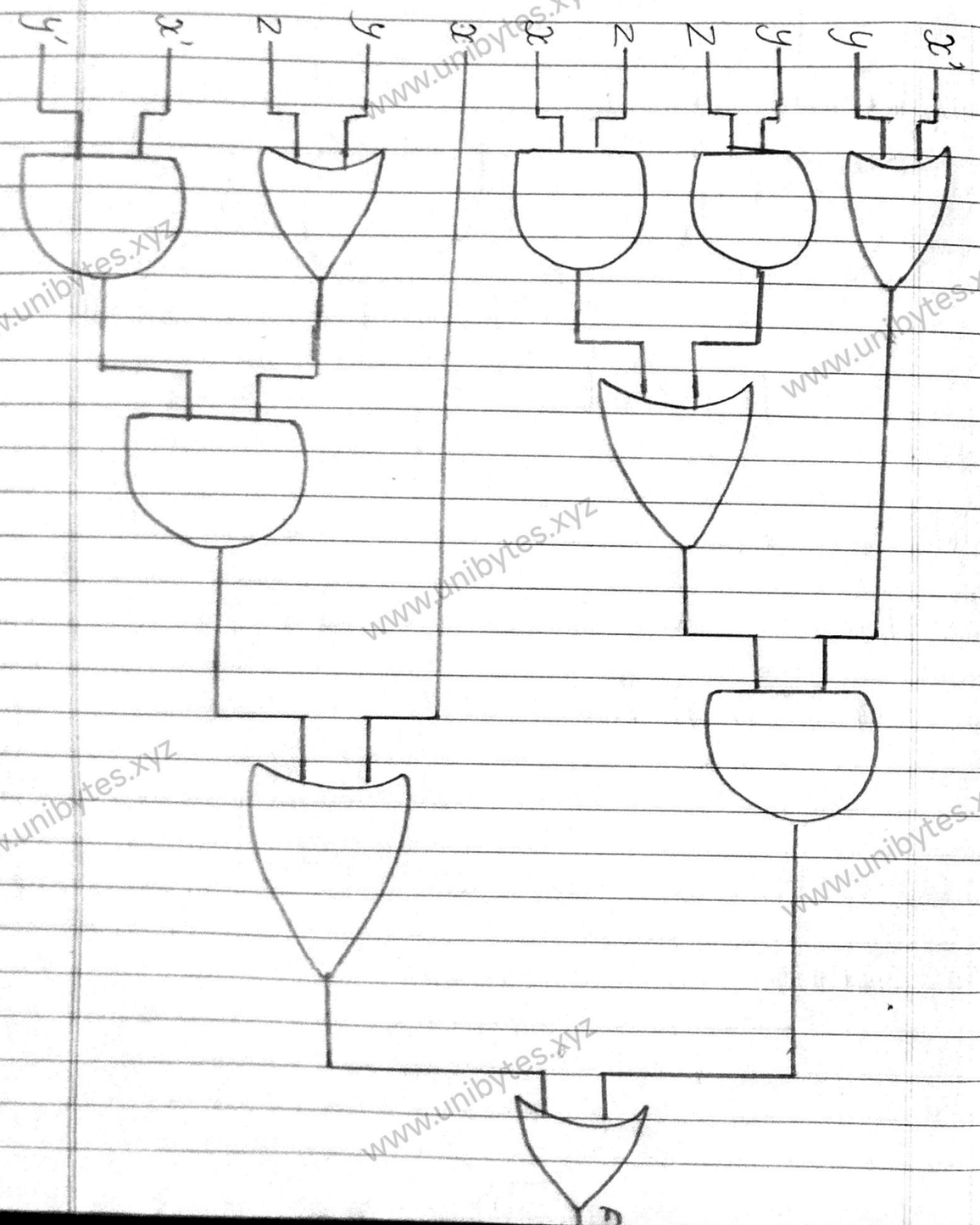
2) If $f = x'+(x'y(yz+xy).(x+y)$. Design a circuit using only ~~NAND and NOR~~ Basic gate.



www.unibytes.xyz

~~3) If $f = x(y+z) + (yz + x'y).(y+z) + x\bar{y}$~~

3) $f = [(x'+y).(yz+zx)] + [x + (y+z).(x'y')]$



Boolean Algebra and Logic Simplification

a) Law of Identify :- Any algebric expression is equal to itself.

For e.g

$X = X$

$0 = 0$

$1 = 1$

b) Communicative law : The order of application of two separate term is not important.

For e.g

i) $A.B = B.A$, ii) $A+B = B+A$

$0.1 = 1.0$

$0 = 0$

c) Double negative law : An input that is AND'ed or OR'ed with itself is equal to the input.

For e.g

$A.A = A$

$A+A = A$

If the value of A is 0 then $A.A = 0$

If the value of A is 1 then $A+A = 1$

d) Law of intersection

e.g $A.1 = A$

$A.0 = 0$

suppose,

If the value of A is 1 then, $1.1 = 1$

$1.0 = 0$

e) Law of Union

i) $A+1=1$

ii) $A+0=A$

i) If the value of A is 0 then $0+1=1$

ii) If the value of A is 1 then $1+0=1$

f) Law of Complementary

For e.g

$\bar{\bar{A}}=A$

$A+\bar{A}=1$

Here,

If $A=0$

$\bar{0}=1$

$\bar{1}=0$

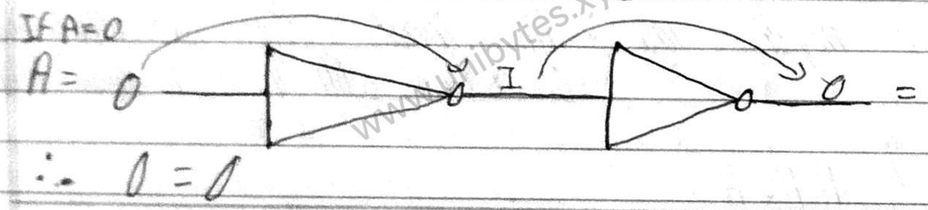
$\therefore 0=0$

If $A=0$

$0+\bar{0}=1$

$0+1=1$

$\therefore 1=1$



$\therefore 0=0$

Demorgan's Theorems

i) According to the first demorgan's theorem the complement result of the "AND" operation is equal to the 'OR' operation of the compliment of the variable".

i.e $\overline{AB}=\bar{A}+\bar{B}$

Truth table :

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

Demorgan's Second

According to the second theorem the compliment result of 'OR' operation is equal to 'AND' operation of the complement of the variable.

i.e $\overline{A+B}=\bar{A}\cdot\bar{B}$

Truth table:

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

Canonical Form

- The standard form of equation is called canonical form
- In boolean algebra the boolean function can be expressed as canonical form which is know as minterms.
- In minterm we look for the functions where the output is '1' while in maxterm we look for the function where the output result is '0'.

book page 85

- ~~Sente~~ Sum of minterm is also known as Sum of Product (SOP).
- Product of maxterm is also known as Product of Sum (POS).

* Standard Form

In standard form, boolean function will contain all the variables in either True form or Complimented form.

Note: i) To find the combination of given bit The formula is 2^n, where n is the number of bit

ii) To show the combination in table we use simple idea. like in 3 bit we have A, B, C and at the first point we put 0 for all of them the from last one 'C' we put 1 and 0 rationally and in 'B' put 0 for 2 time and 1 for 2 times and for C 4 for 0 and 4 times for 1 and after for more put 8 times for 0 8 time for 1 $8 \div 2 = 2$, and for A $= 2 \times 2 = 4$, and go on, lik, $2 \times 4 = 8$, $8 \times 2 = 16$....

2 Bit Combination

$= 2^n = 2^2 = 4$

A	B
0	0
0	1
1	0
1	1

3 Bit combination

A	B	C	$2^n = 2^3 = 8$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Minterm (SOP) = 1
Maxterm (POS) = 0

Minterm (m) and Maxter (M) for three variables

x	y	z		Min term (product term)		Maxter^n (Sum term)
0	0	0	m_0	$\bar{x}.\bar{y}.\bar{z}$	M_0	$x+y+z$
0	0	1	m_1	$\bar{x}.\bar{y}.z$	M_1	$x+y+\bar{z}$
0	1	0	m_2	$\bar{x}.y.\bar{z}$	M_2	$x+\bar{y}+z$
0	1	1	m_3	$\bar{x}.y.z$	M_3	$x+\bar{Y}+\bar{z}$
1	0	0	m_4	$x.\bar{y}.\bar{z}$	M_4	$\bar{x}+Y+z$
1	0	1	m_5	$x.y.z$	M_5	$\bar{x}+\bar{Y}+\bar{z}$
1	1	0	m_6	$x.y.\bar{z}$	M_6	$\bar{x}+\bar{Y}+z$
1	1	1	m_7	$x.y.z$	M_7	$\bar{x}+\bar{Y}+\bar{z}$

Note:
i) Small 'm' denote Minterm
ii) Capital 'M' denote Maxterm
iii) In minterm if the value of variable (x,y,z) are 0 then they will be complement ($\bar{x}\bar{y}z$) otherwise if the value of any one variable become one(1) then the variable is not become compliment but other remaing variable will be compliment (*).
iv) Just opposite of minterm is Max term (+).

Question
i) $M_7 = Max(x,y,z) = x' + y' + z'$
ii) $m_6 = Min(x,y,z) = x.y.\bar{z}$

The process of using standard equation instead of the truth table is called cannical form.

www.unibytes.xyz

* Karnaugh MAP (K-map)

- Karnaugh Map is also known as Veitch Diagram or Simply known K-map. It is the two dimensional form of truth table. The K-map will simplify the boolean expression which can be seen from the location of 1s in the map.

i) Two variable Boolean function can be represented as follow:

B \ A	0	1
0	$A'B'$	AB'
1	$A'B$	AB

ii) Three variable

C \ AB	00	01	11	10
0	$A'B'C'$	$A'BC'$	ABC'	$AB'C'$
1	$A'B'C$	$A'BC$	ABC	$AB'C$

Note:

At a time only one bit (number) can be change like in (01, 11). In first the last one is '1' and we can only change only one value so we put 1. Both the bit can't change at a time.

iii) Four variable

AB \ CD	00	01	11	10
00	$A'B'C'D'$	$A'B'C'D$	$ABC'D'$	$AB'C'D'$
01	$A'B'C'D$	$A'B'C'D$	$ABC'D$	$AB'C'D$
11	$A'B'CD$	$A'BCD$	$ABCD$	$AB'CD$
10	$A'B'CD'$	$A'BCD'$	$ABCD'$	$AB'CD'$

Note:

i) 0 mean Compliment ($\bar{A}$, A').

ii) 1 mean non-compliment (A).

1) Design K-Map of the following.

i) $A'BC'D + A'BCD + ABC'D' + ABC'D + ABCD + ABCD' + AB'CD + AB'CD'$

Sol:

AB \ CD	00	01	11	10
(00)			1	
(01)		1	1	
(11)		1	1	1
(10)			1	1

P means pair

P_1, P_2, P_3

$P_1 = A'BC'D + A'BCD + ABC'D + ABCD$
$= BD$

$P_2 = ABC'D' + ABC'D + ABCD + ABCD'$
$= AB$

$P_3 = ABCD + ABCD' + AB'CD + AB'CD'$
$= AC$

$F = P_1 + P_2 + P_3$
$= BD + AB + AC$

2) K-MAP with Circuit Diagram

i) $F = a'b + ab$

Sol:

B \ A	0	1
0		
1	1	1

P_1

$P_1 = a'b + ab$
$= b$

ii) $F = A'B'C' + A'B'C + A'BC' + ABC' + ABC$

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

P_2 P_1 P_3

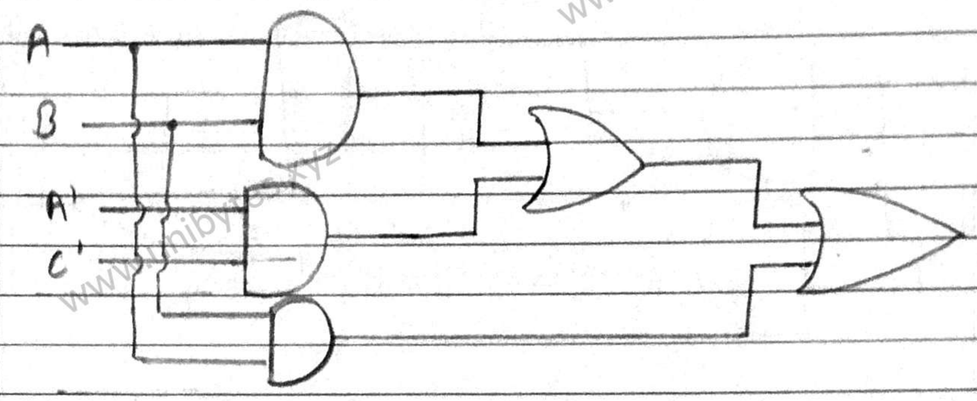
$P_1 = ABC' + ABC$
$= AB$

$P_2 = A'B'C' + A'BC'$
$= A'C'$

$P_3 = A'B'C' + A'B'C$
$= AB$

Note: yed Direct F= AB C+ BCA yesto deko xa vane AB mathi hunxa.
yedi $\Sigma(A,B,C,D) = [1,2,3,4]$ yesto deko xa vane AB ~~tala~~ hunxa ani CD mathi.

$F = AB + A'C' + AB$



iii) Note:-

i)

C \ AB	00	01	11	10
0	0	1	3	2
1	4	5	7	6

i) We have interchange the last two value lik 2 and 3 or 7 and 6

ii)

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

∴ We have to change the value ~~when~~ in the place "11 or 10" for both vertical and horizontal side.

iii) $F(A,B,C) = \Sigma m(0,1,5,6,7)$

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

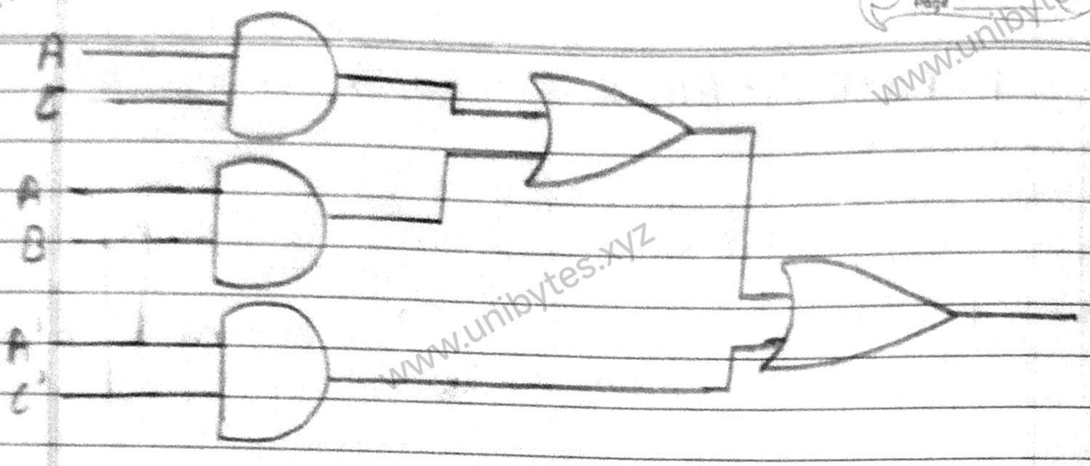
P_1 P_2 P_3

$P_1 = ABC + AB'C'$
$= AB$

$P_2 = A'BC' + A'BC$
$= AB$

$P_3 = A'B'C' + A'BC'$
$= A'C'$

$F = AC + AB + A'C'$



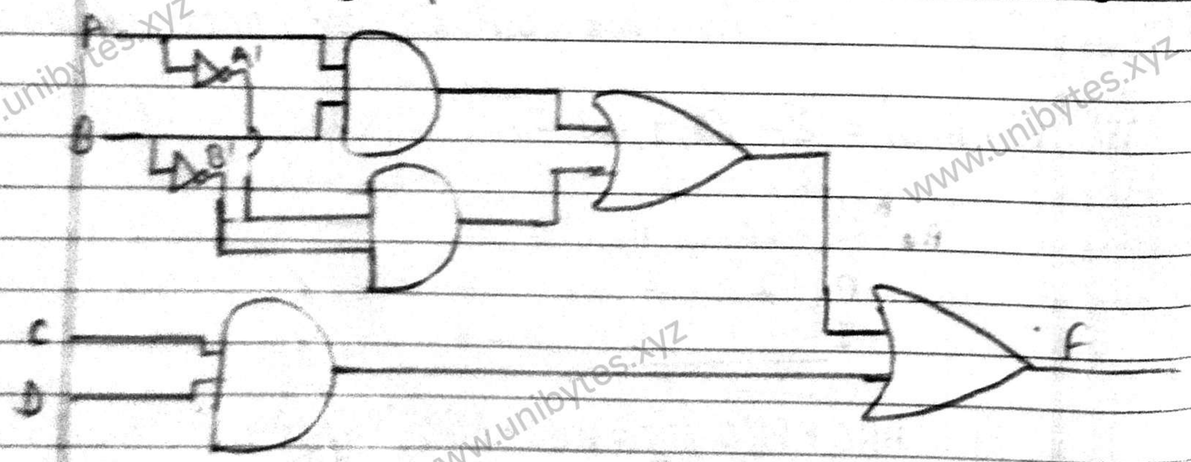
iv) $F(A,B,C,D) = \Sigma m(0,2,3,7,11,13,14,15)$

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

P_1 P_2 P_3 P_4

$P_1 =$ [redacted] , $P_2 = ABC$, $P_3 = ABD$, $P_4 = A'B'$

$F = P_1 + P_2 + P_3 + P_4 = CD + AB + AB + A'B' = CD + AB + A'B'$



* In the sum of products function $f(x,y,z) = \Sigma(0,3,4,5)$ ~~the~~ find the prime implicants.

x \ yz	00	01	11	10
0			1	1
1	1	1		

$\bar{x}y$

$x\bar{y}$

$\therefore$ The prime implicants are $\bar{x}y$, $x\bar{y}$

Don't Care Condition

- Some time we don't care where 1 or 0 occurs
- It may be that those inputs while occur so it make no difference what the output is.
- It is denoted by 'x'

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

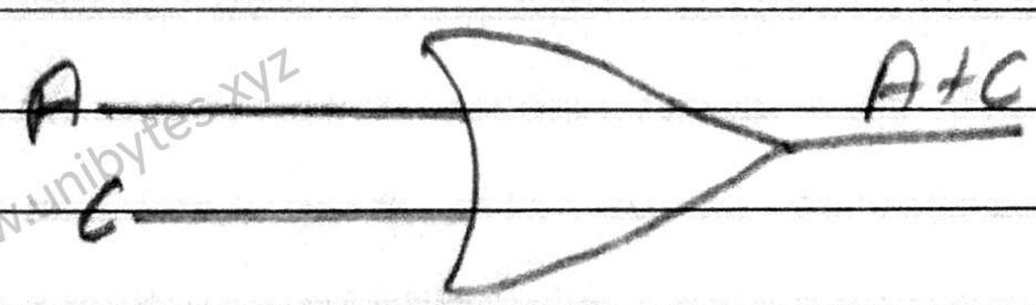
A \ BC	00	01	11	10
0			1	1
1	1	1	x	x

P_1 P_2

$P_1 = ABC' + AB'C' + ABC + AB'C$
$= A$

$P_1 = A'B'C + A'BC + ABC + AB'C$
$= C$

$F = P_1 + P_2$
$= A + C$



Chapter 3 Combinational Circuit (another page)

Adder (fix question)

There are two types of Adder

i) Half Adder

- A half adder is a logical circuit that performs an addition operation on two binary digits
- To ~~mak~~ make half adder circuit first of all we need 2 bits, (A and B)
- The inputs are named as A and B and the outputs are named as Sum (S) and carry (C).
- The truth table of half adder circuit

Input		Output	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

1+1 = 10
so sum = 0
carry = 1

- Boolean function of sum and carry (K-MAP)

K-MAP of Sum

A \ B	0	1
0	0	1
1	1	0

$AB' + A'B = A \oplus B$

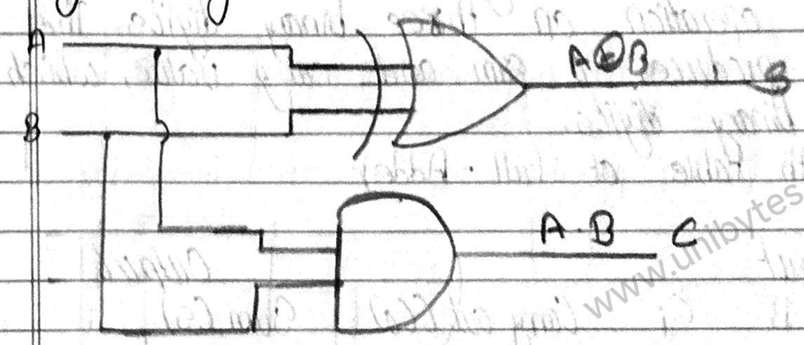
Note applyin NOR equation $AB' + A'B = A \oplus B$

K-MAP of Carry

A \ B	0	1
0		
1		1

AB

Logic diagram of half-adder



Combination Circuit

- Does't use any memory.
- Combination circuit consist of logic gate whose output at any time is determine directly from the present combination of input.
- In general combination circuit has 'M' input 'N' output.
- Output is dependent upon combination of input variables

Designing of combination circuit

Steps for designing combinational circuit

- The problem is stated
- The number of input variable or output variable are determine

- The truth table is created
- The simplified boolean function is obtained
- The logic diagram is drawn.

* Full Adder

A full adder is a logical circuit that perform an addition operation on three binary digits. The full adder produces a sum and carry value, which are both binary digits.

Truth Table of Full-Adder

Input			Output	
A	B	C_i	Carry out (C_o)	Sum (S)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$C_o = A'BC_i + ABC_i' + ABC_i + ABC_i$, $S = A'B'C_i + A'BC_i' + AB'C_i' + ABC_i$

$C_o = A'BC_i + ABC_i' + ABC_i + AB'C_i$ by manipulating C_o, we can see that $C_o = C_i(A \oplus B) + AB$

$S = A'B'C_i + A'BC_i' + ABC_i' + ABC_i$ by manipulating S, we can see that $S = C_i \oplus (A \oplus B)$.

Note: Carry maa 1 vako ma ~~ra~~ ball value ~~her~~ rakhi like 4th place maa 1 xa so $A'BC_i$, same for sum.

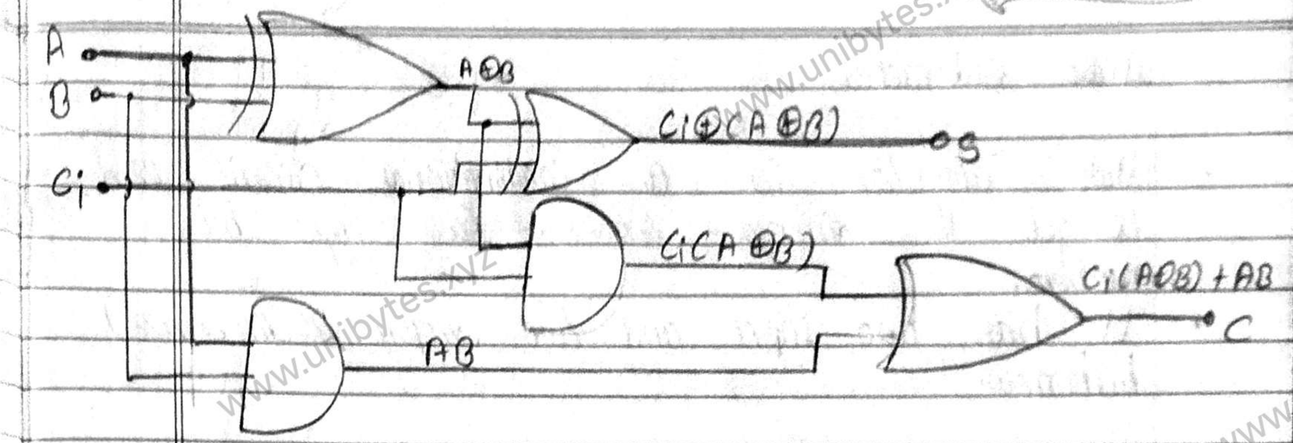
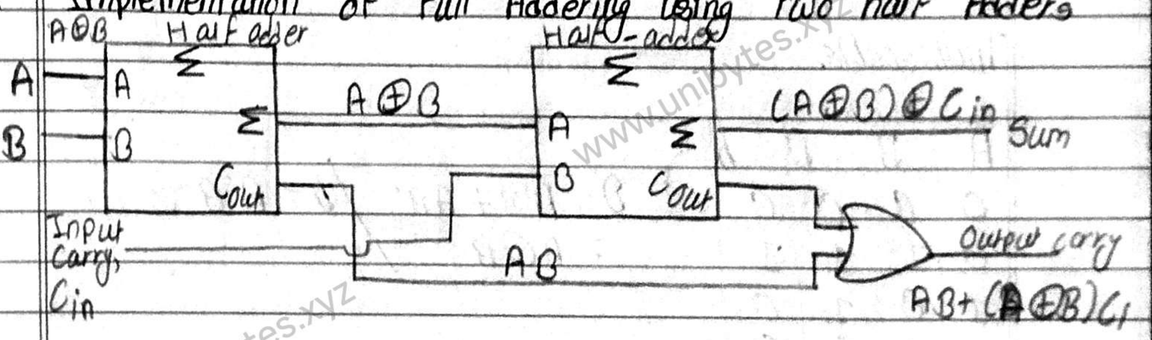


Figure Logic diagram, block diagram of full adder.

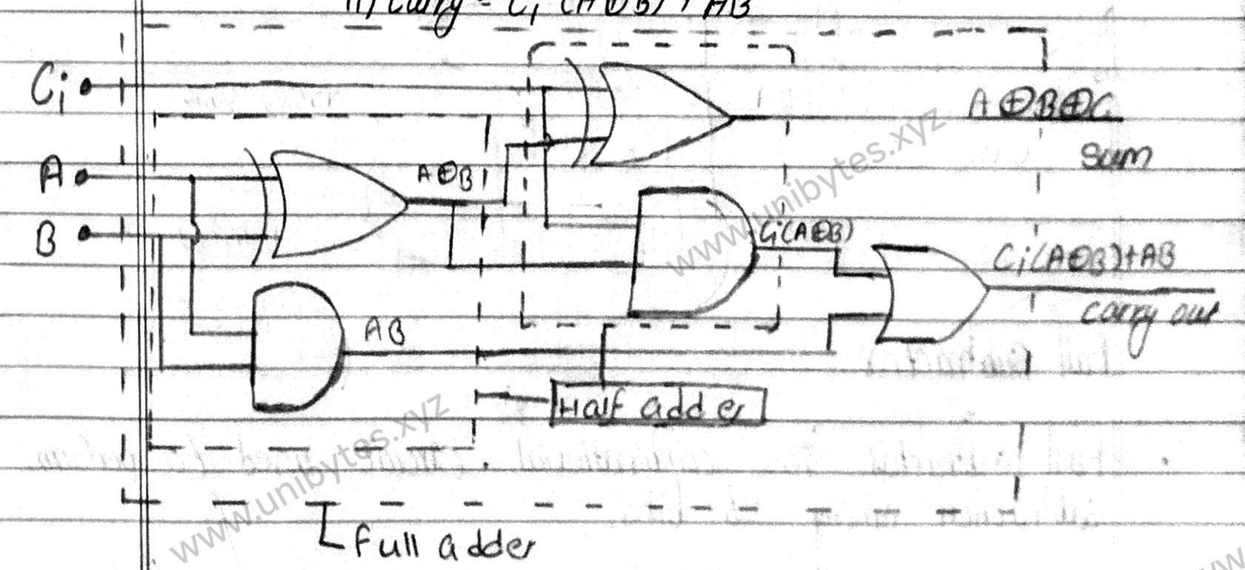
* Implementation of Full Adding using two half Adders



* Arrangement of two-half-adders to form a full-adder.

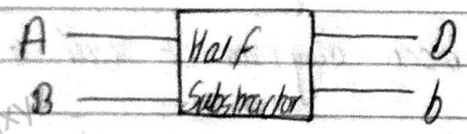
are Conditions: i) Sum = $A \oplus B \oplus C$

ii) Carry = $C_i(A \oplus B) + AB$



Half Subtractor

- Half Subtractor is a combinational circuit used to get the difference between two single bit numbers.
- It has two input and two output D (difference) b(borrow.

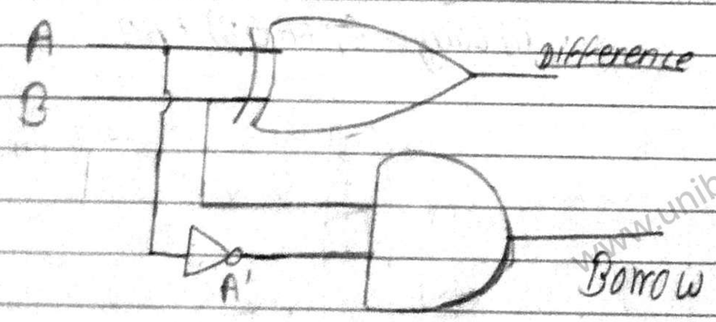


Truth table.

A	B	D	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = A'B + AB'$ / $b = A'B$
$= A \oplus B$

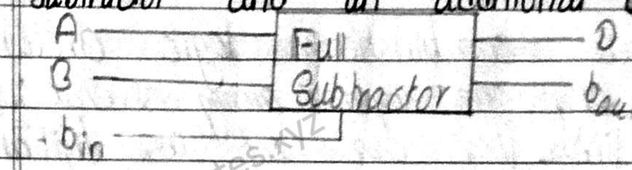
Logic Diagram:



Full Subtractor

- Full subtractor is combinational circuit used to perform subtraction among 3 bits.

- A full subtractor is made by combining two Half subtractor and an additional OR-gate.



Truth table

A	B	b_{in}	D	b_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-map for difference

A \ $B b_{in}$	00	01	11	10
0		1		1
1	1		1	1

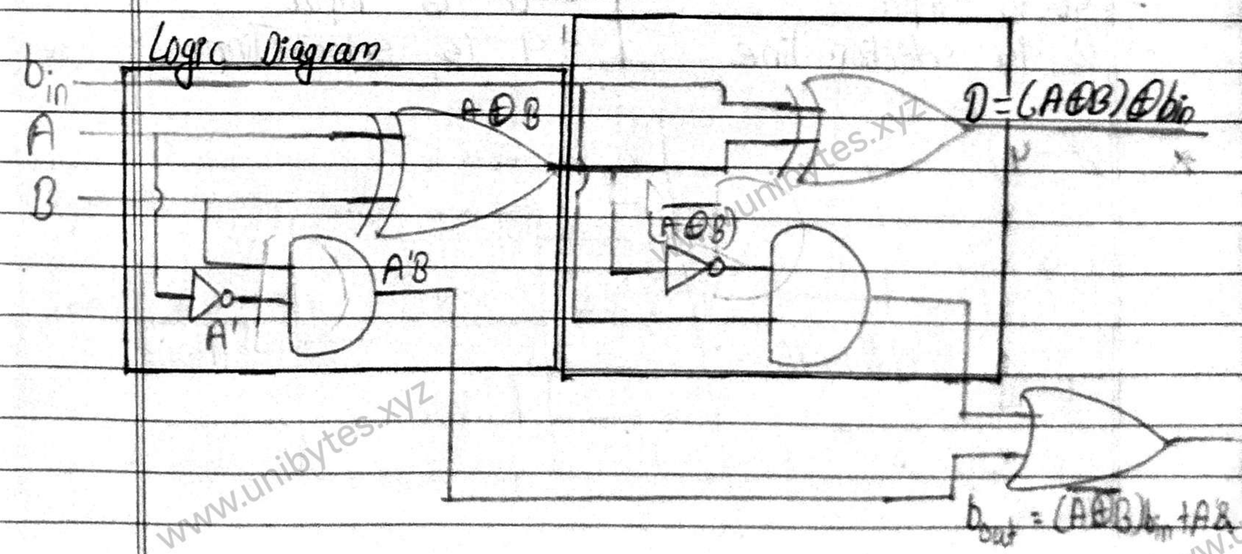
$D = A'B'b_{in} + AB'b_{in}' + A'Bb_{in}' + ABb_{in}$
$= (A \oplus B) \oplus b_{in}$

K-map for Borrow (b_{out})

A \ $B b_{in}$	00	01	11	10
0		1	1	1
1			1	

$\bar{A}B + Bb_{in} + \bar{A}b_{in}$
$= b_{in}(\overline{A \oplus B}) + A'B$

Logic Diagram



Multiplexer

- Multiplexer is a combinational logic circuit used to select only one input among several inputs based on selection lines. This can act as digital switch. This is also called as data selector. For a MUX there can be 2^n inputs, n selection lines & only one output.
- In multiplex the output is always 1 that is 1 input is selected at a time and the selected input is transmitted as a output.
- It is a combinational circuit that is given a certain number of input, power 2.

For example.

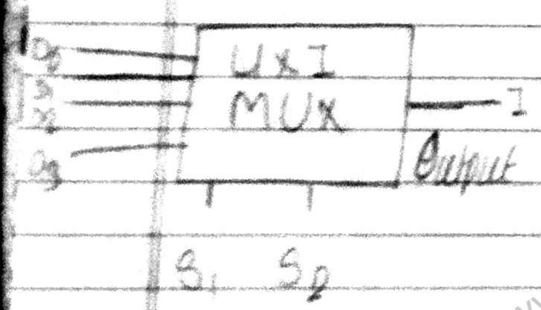
i) $2^1 = 1$
- 2 is input
- 1 is selection line

ii) $2^2 = 4$
- 4 is input
- 2 is selection line

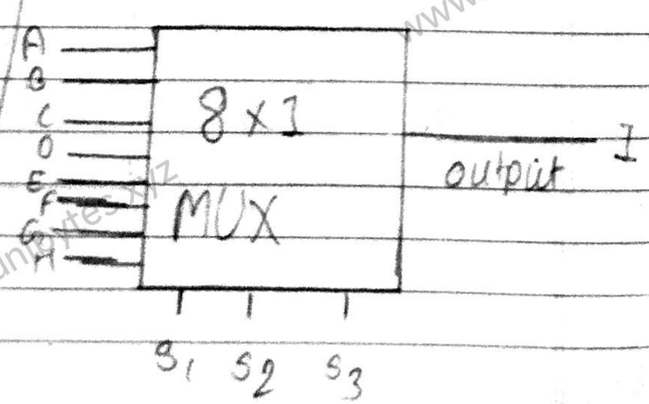
iii) $2^3 = 8$
- 8 is input
- 3 is selection line

iv) $2^4 = 16$
- 16 is input
- 4 is select line

* $2^2 = 4$

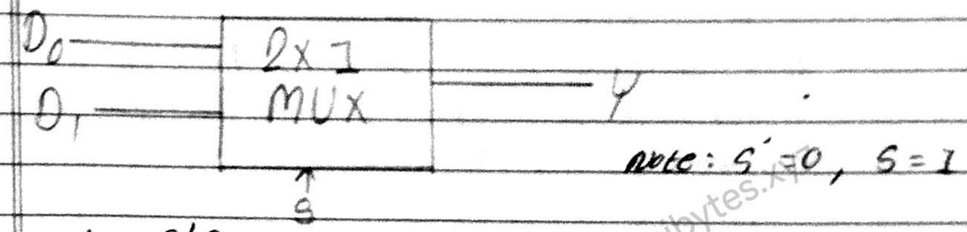


* $2^3 = 8$



2 X 1 Multiplexor

- A 2x1 multiplexor has two inputs 'D_0' and 'D_1' and one select line 'S' and one output 'Y'.
- If the select line is low (0) then the output will switched to D_0 where as if the select line is high (1) the the output will switched to D_1.



note: $S' = 0$, $S = 1$

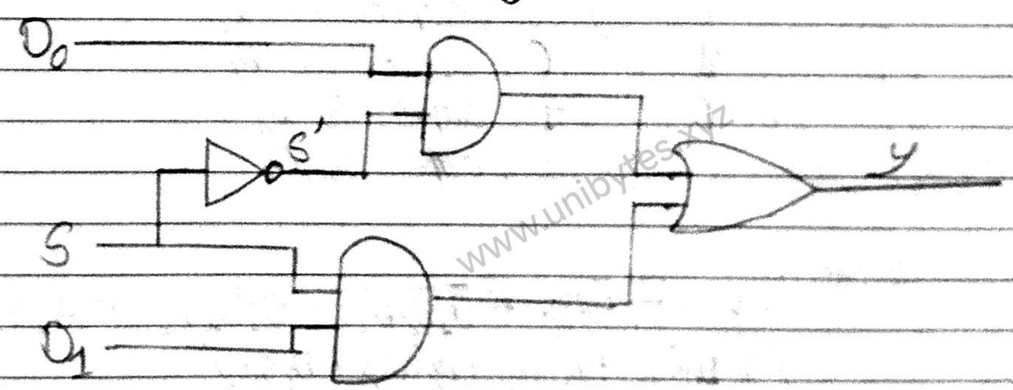
$y = S'D_0$, $y = SD_1$

The expression is: $y = S'D_0 + SD_1$

Truth table

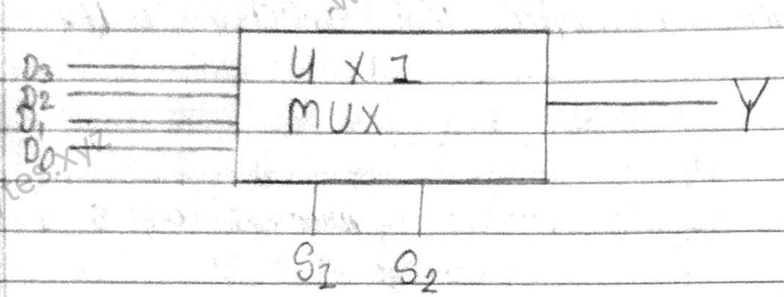
S	Y
0	D_0
1	D_1

logic Diagram



ii) 4-to-1 Multiplexer

4 to 1 Multiplexer consits 4 data Input line as D_0, D_1, D_2, D_3 two Select line as S_1 and S_2 and one single output line Y. The block Diagram of 4x1 MUX



Select		Data Input	Output
S_2	S_1		Y
0	0		D_0
0	1		D_1
1	0		D_2
1	1		D_3

If $S_1 = 0$, $S_2 = 0$, output is D_0
If $S_1 = 0$, $S_2 = 0$, output is D_1
If $S_1 = 0$, $S_2 = 1$, output is D_2
If $S_1 = 1$, $S_2 = 1$, output is D_3

Equation

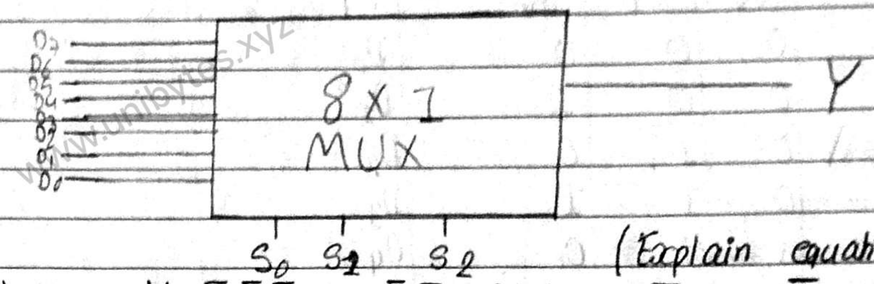
Equation is compulsory

$$\cancel{Y = \bar{S}_1\bar{S}_2D_0 + \bar{S}_1S_2D_1 + S_1\bar{S}_2D_2 + S_1S_2D_3}$$

$$Y = D_0\bar{S}_1\bar{S}_2 + D_1S_1\bar{S}_2 + D_2\bar{S}_1S_2 + D_3S_1S_2$$

* 8-to-1 Multiplexer

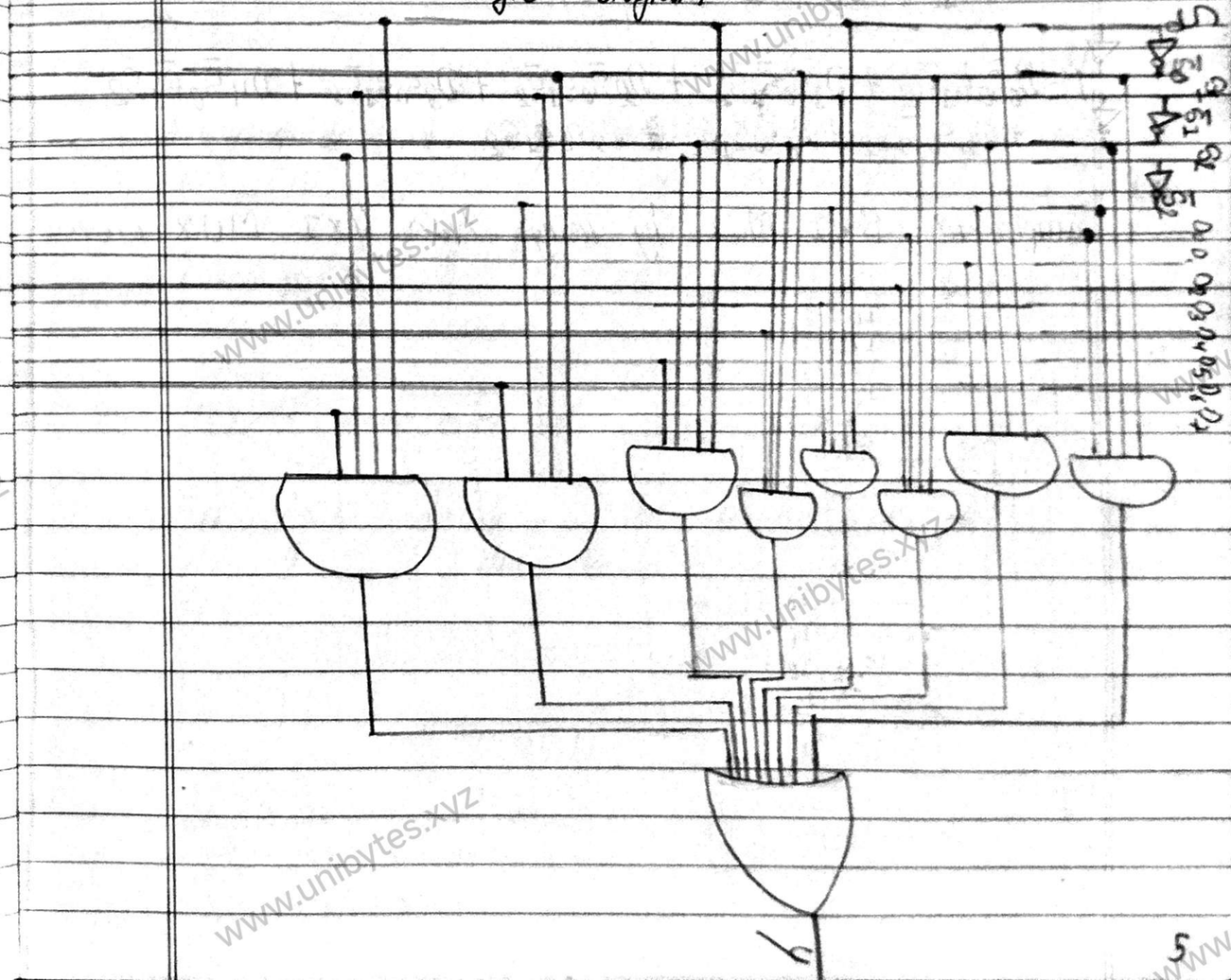
~~Bock~~ Block diagram of 8x1 MUX



(Explain equation yourself)

Equation: $\cancel{Y = \bar{S}_0\bar{S}_1\bar{S}_2D_0 + \bar{S}_0\bar{S}_1S_2D_1 + \bar{S}_0S_1\bar{S}_2D_2 + \bar{S}_0S_1S_2D_3 + S_0\bar{S}_1\bar{S}_2D_4 + S_0\bar{S}_1S_2D_5 + S_0S_1\bar{S}_2D_6 + S_0S_1S_2D_7}$

Logic Diagram



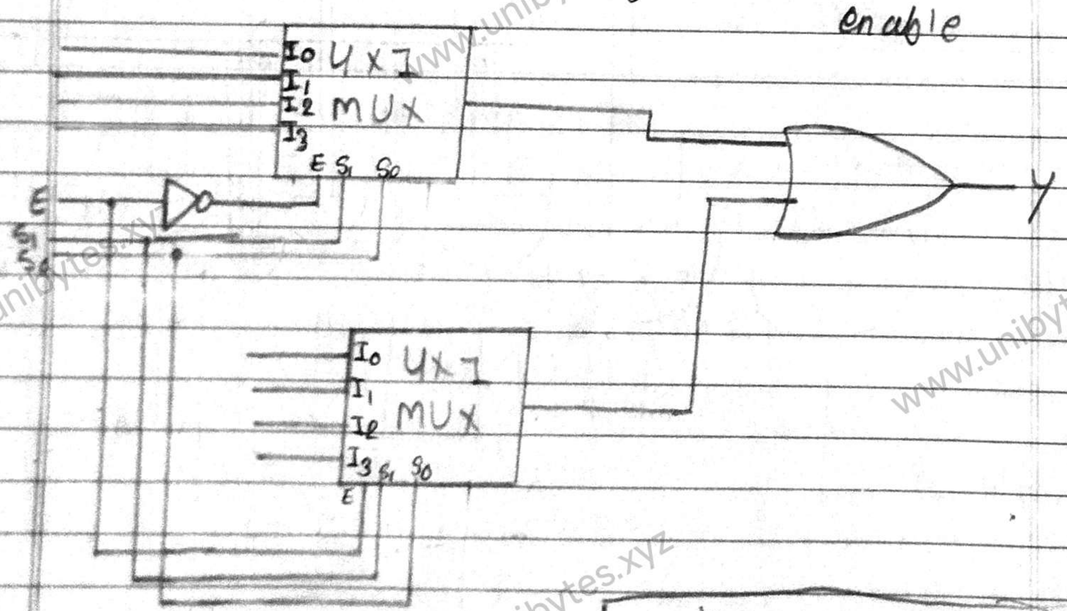
Truth table of 8x1 MUX

Select		Data Input	Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

$$Y = D_0\bar{S}_0\bar{S}_1\bar{S}_2 + D_1S_0\bar{S}_1\bar{S}_2 + D_2\bar{S}_0S_1\bar{S}_2 + D_3S_0S_1\bar{S}_2 + D_4\bar{S}_0\bar{S}_1S_2 + D_5S_0\bar{S}_1S_2 + D_6\bar{S}_0S_1S_2 + D_7S_0S_1S_2$$

I.m.p

Implement 8x1 MUX by using two 4x1 MUX with enable



without enable also importance - Book

$n = 2$

$0 - 2^n$ $\quad 2^2 = 4$

Select	Data Input		Output
E	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

$n = 3$

$2^3 = 8$

$n =$

$n =$

E ∝

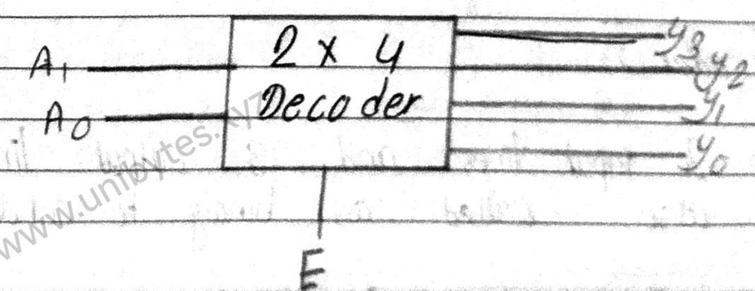
* Decoder

A decoder is a combinational circuit that has 'N' input lines and maximum of 2^n output line. One of these output will be active based on the combination of input present, when the decoder is enables.

i) 2 x 4 decoder

In 2x4 decoder the number of input lines are 2 and the number of output lines are 4.

The Block Representation



where E - Enable → E=0 Decoder is disable
→ E=1 Decoder is able

Truth table :

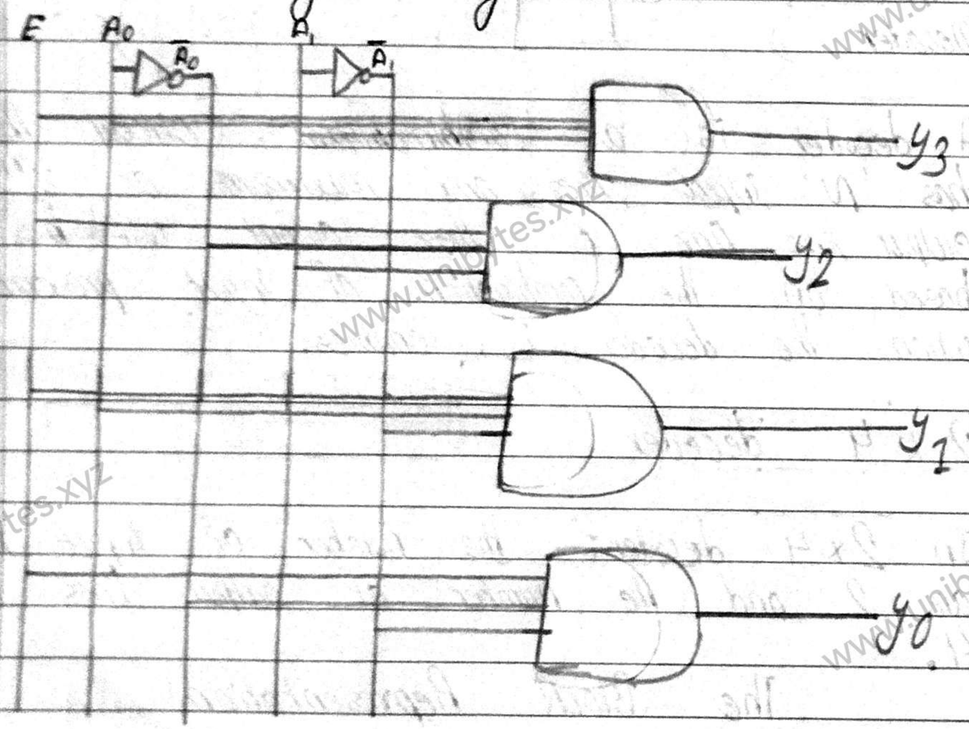
E	A_1	A_0	y_3	y_2	y_1	y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Equation :

$y_3 = EA_1A_0$ $\quad$ $y_1 = E\bar{A}_1A_0$

$y_2 = EA_1\bar{A}_0$ $\quad$ $y_0 = E\bar{A}_1\bar{A}_0$

Logic Diagram



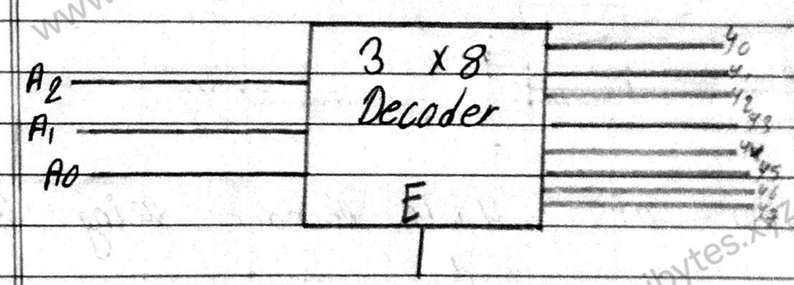
ii) 3:8 decoder

It has 3 input lines and 8 output lines.
It is also called as binary to octal decoder. It takes a 3-bit binary input code and activates one of the 8 (octal) outputs corresponding to that code.

Truth table

E	A_2	A_1	A_0	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Block Representation



Equation :

$y_0 = E\bar{A}_0\bar{A}_1\bar{A}_2$ $\quad$ $y_4 = E\bar{A}_0\bar{A}_1A_2$

$y_1 = EA_0\bar{A}_1\bar{A}_2$ $\quad$ $y_5 = EA_0\bar{A}_1A_2$

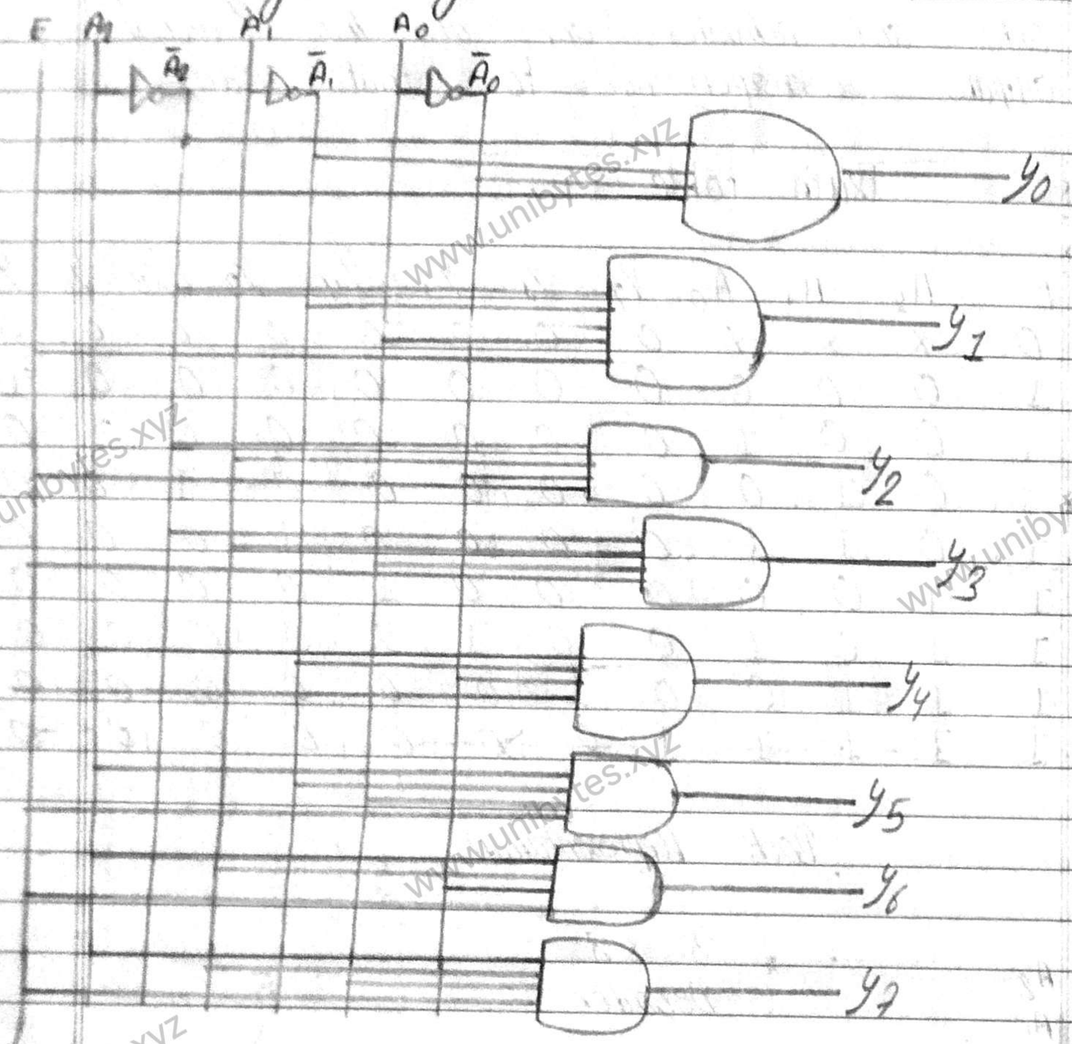
$y_2 = E\bar{A}_0A_1\bar{A}_2$ $\quad$ $y_6 = E\bar{A}_0A_1A_2$

$y_3 = EA_0A_1\bar{A}_2$ $\quad$ $y_7 = EA_0A_1A_2$

www.unibytes.xyz

Date
Page

Logic Diagram



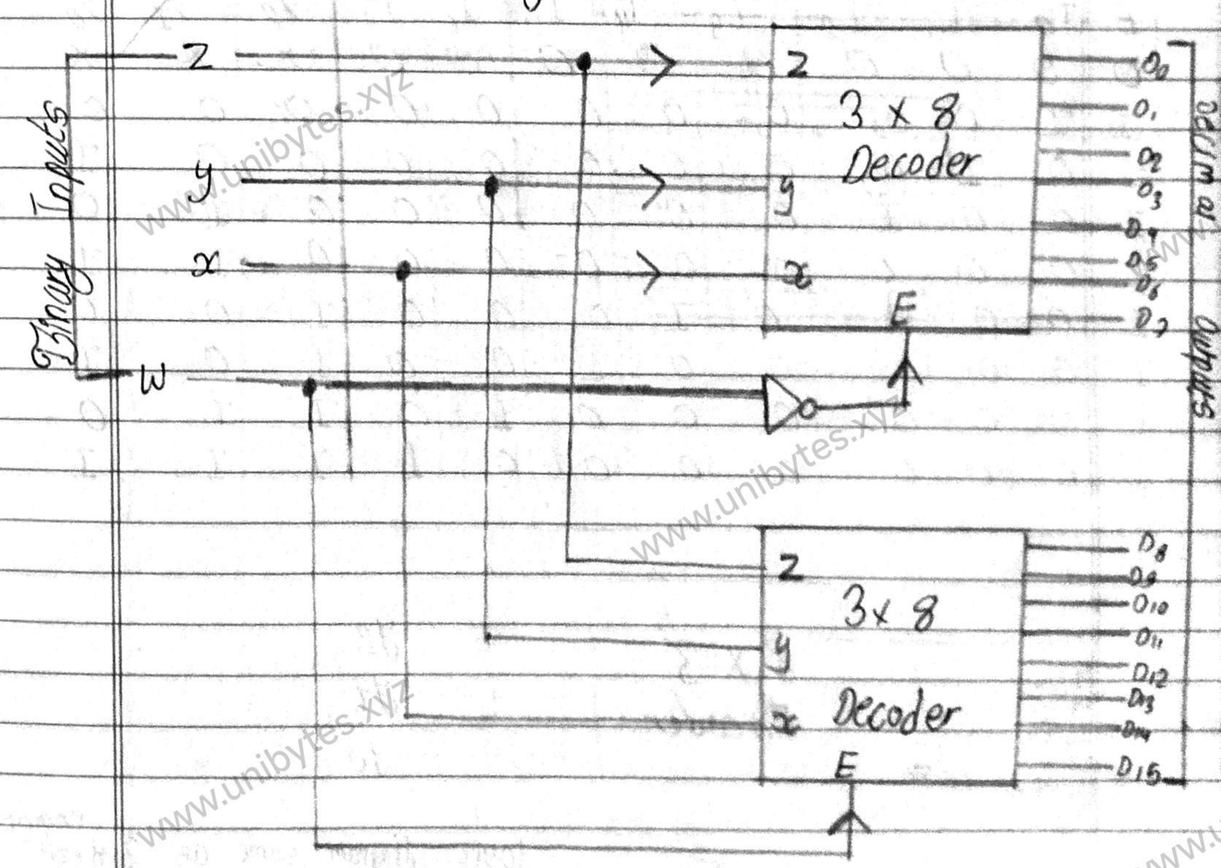
* Implementation of 4x16 Decoder using 3x8 Decorder.

Truth table / Function table

	Binary Input				Decimal Output
	W	X	Y	Z	
Upper Decoder	0	0	0	0	D_0
	0	0	0	1	D_1
	0	0	1	0	D_2
	0	0	1	1	D_3
	0	1	0	0	D_4
	0	1	0	1	D_5
	0	1	1	0	D_6
	0	1	1	1	D_7
Lower Decoder	1	0	0	0	D_8
	1	0	0	1	D_9
	1	0	1	0	D_{10}
	1	0	1	1	D_{11}
	1	1	0	0	D_{12}
	1	1	0	1	D_{13}
	1	1	1	0	D_{14}
	1	1	1	1	D_{15}

Date
Page

Logic Diagram



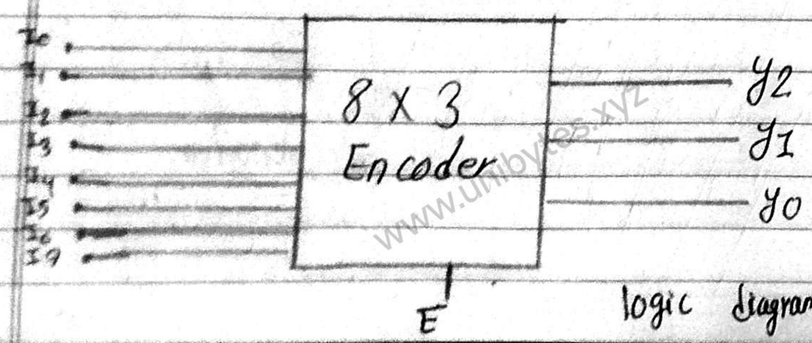
2) Encoder

Encoder is a logic circuit which performs the opposite action of the Decoder. An encoder has a number of input lines, only one of which is activated at a time. At the output it displays a value corresponding to the active ~~output~~ input.

I) Octal to Binary Encoder

Octal-to-Binary take 8 inputs and provides 3 outputs, thus doing the opposite of what the 3-to-8 decoder does.

E	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	X	X	X
1	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	1	1
1	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1	1



logic diagram same as 3x8 ~~decoder~~ encoder.

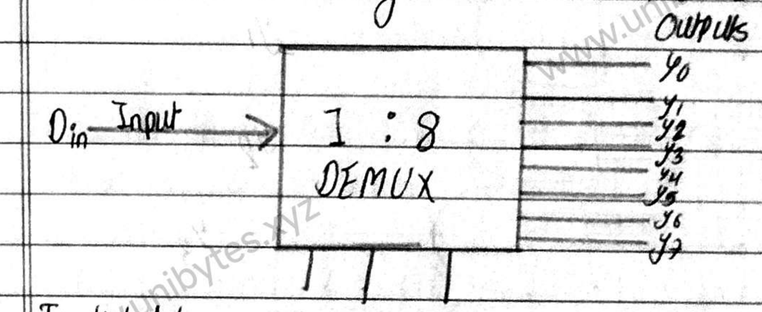
2) Demultiplexer

The demultiplex is the inverse of the multiplexer, in that it takes a single data input and n address inputs.

i) 1 to 8 Demultiplex

A 1 to 8 demultiplexer consits of one input line 8 output lines and 3 select lines.

Block diagram of 1 X 8 DMUX



Truth table:

Data Input	Select inputs			Outputs							
D	S_2	S_1	S_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	D	0
D	0	1	0	0	0	0	0	0	D	0	0
D	0	1	1	0	0	0	0	D	0	0	0
D	1	0	0	0	0	0	D	0	0	0	0
D	1	0	1	0	0	D	0	0	0	0	0
D	1	1	0	0	D	0	0	0	0	0	0
D	1	1	1	D	0	0	0	0	0	0	0

www.unibytes.xyz

Equation:

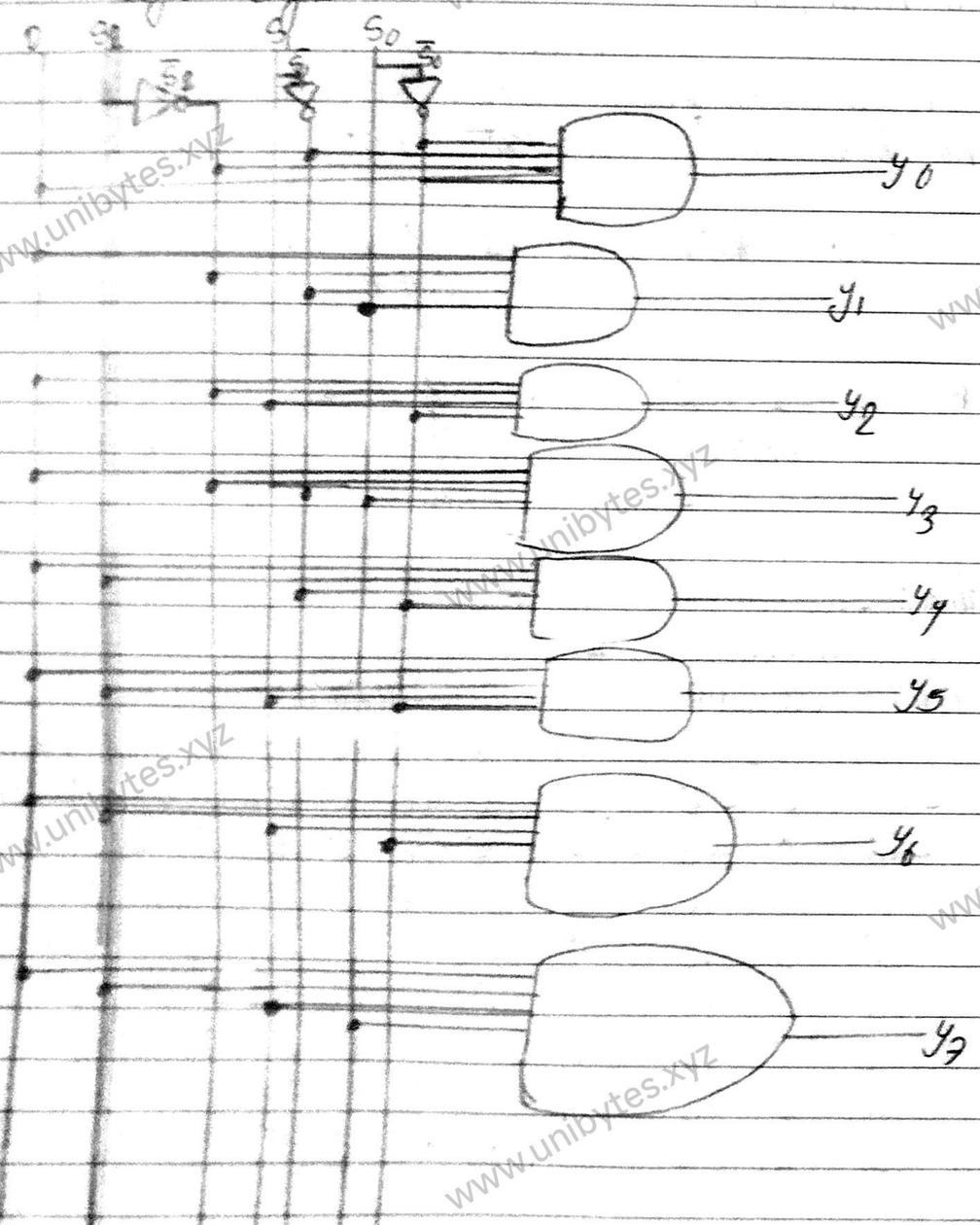
$y_0 = D\bar{S}_0\bar{S}_1\bar{S}_2$ $\quad$ $y_4 = D\bar{S}_0\bar{S}_1S_2$

$y_1 = D\bar{S}_0\bar{S}_1\bar{S}_2$ $\quad$ $y_5 = D\bar{S}_0\bar{S}_1S_2$

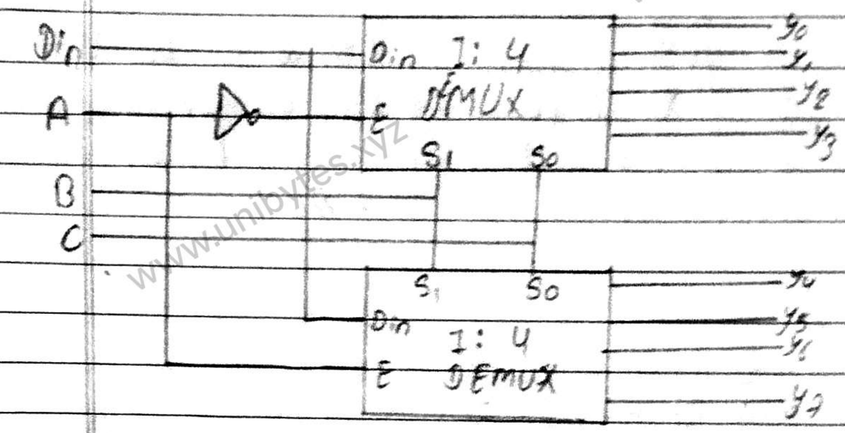
$y_2 = D\bar{S}_0S_1\bar{S}_2$ $\quad$ $y_6 = D\bar{S}_0S_1S_2$

$y_3 = DS_0S_1\bar{S}_2$ $\quad$ $y_7 = DS_0S_1S_2$

Logic diagram

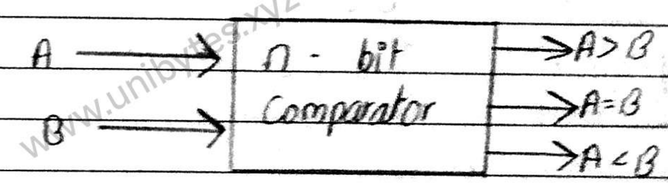


* 1 to 8 DMUX using two 1 to 4 Demux



* Magnitude Comparator

It is a combinational circuit that compare two numbers A, B and determines their relative magnitudes. There are three outcome $A > B$, $A = B$ or $A < B$.



1 Bit Comparator (Truth table)

A	B	A = B	A > B	A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

www.unibytes.xyz

Date
Page

K-Map of $A=B$

A \ B	0	1
0	1	0
1	0	1

$= A\bar{B} + AB$

K-Map of $A>B$

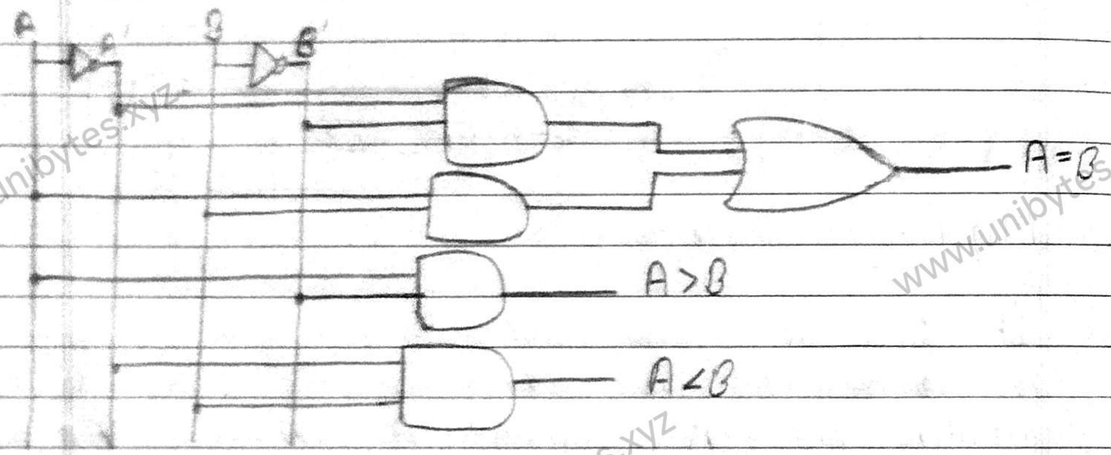
A \ B	0	1
0	0	0
1	1	0

$= A\bar{B}$

K-MAP of $A<B$

A \ B	0	1
0	0	1
1	0	0

$= \bar{A}B$



* 2 Bit Comparator

00 = 0
01 = 1
10 = 2
11 = 3

A		B		A>B	A=B	A<B
A_2	A_1	B_2	B_1			
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Date
Page

A_2	A_1	B_2	B_1	$A>B$	$A=B$	$A<B$
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

K- Map for $A>B$

B_2B_1 \ A_2A_1	00	01	11	10
00		1	1	1
01			1	1
11			1	
10				

$= \bar{B}_2 A_2 + \bar{B}_1 A_2 A_1 + \bar{B}_2 \bar{B}_1 A_1$

K- MAp for $A=B$

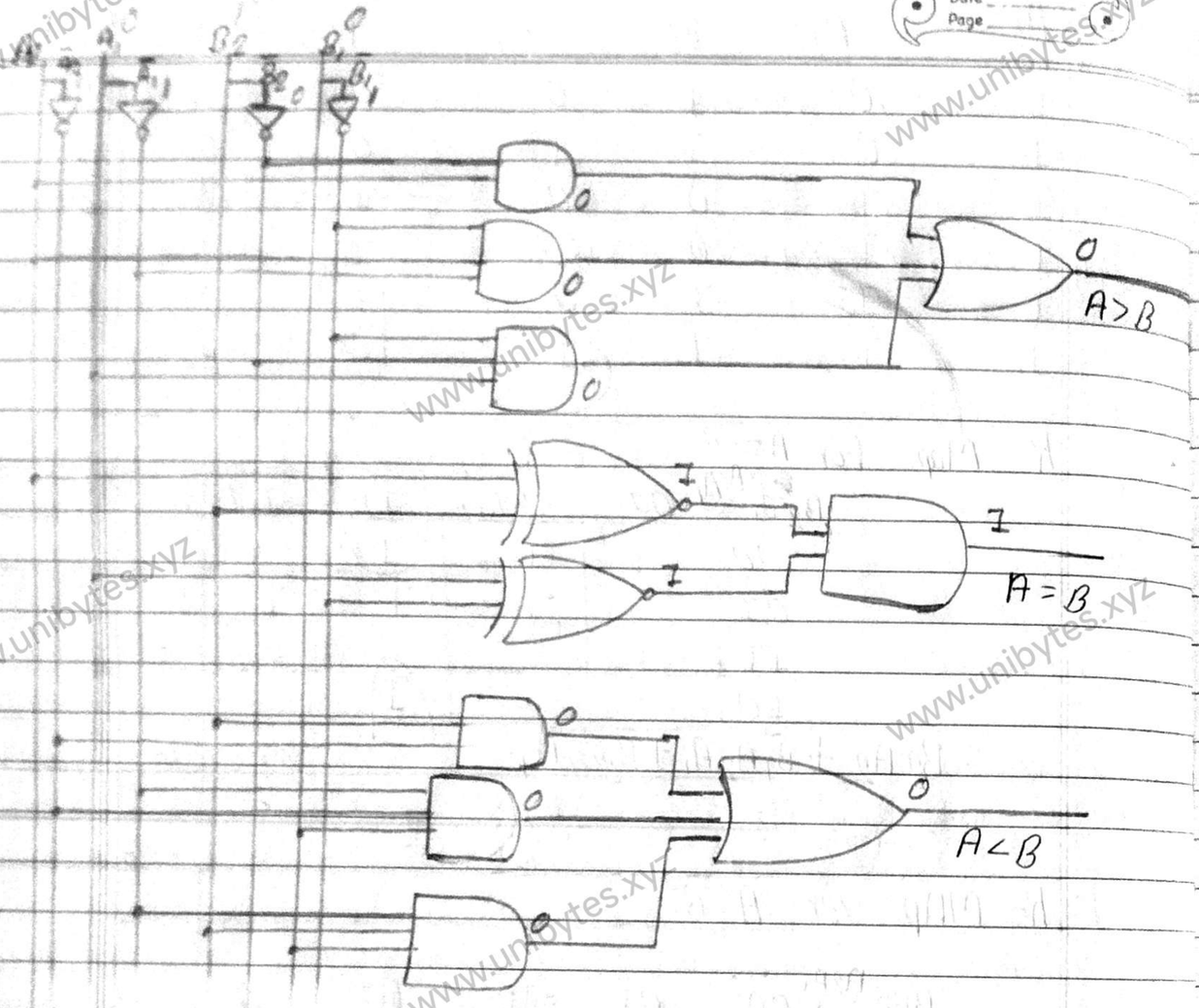
B_2B_1 \ A_2A_1	00	01	11	10
00	1			
01		1		
11			1	
10				1

$= (A_2 \odot B_2) \times (A_1 \odot B_1)$

K -Map for $A<B$

B_2B_1 \ A_2A_1	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

$= B_2\bar{A}_2 + B_1\bar{A}_2\bar{A}_1 + B_2B_1\bar{A}_1$



* Error Detection Code → U.U. Imp

The binary information is transferred from one location to another location through some communication medium. The external noise can change bits from 1 to 0 or 0 to 1. This changes in values changes the meaning of actual message and is called error. For efficient data transfer, there should be an ~~error~~ error detection and correction codes. An error detection code is binary code that detects digital error during transmission.

Note: i) In odd parity we use '0' for odd and '1' for even.
ii) In even parity we use '0' for even ~~and '1' for~~ odd.

i) Odd Parity

A Parity bit is an extra bit included in binary message to make a total number of 1's either odd or even.

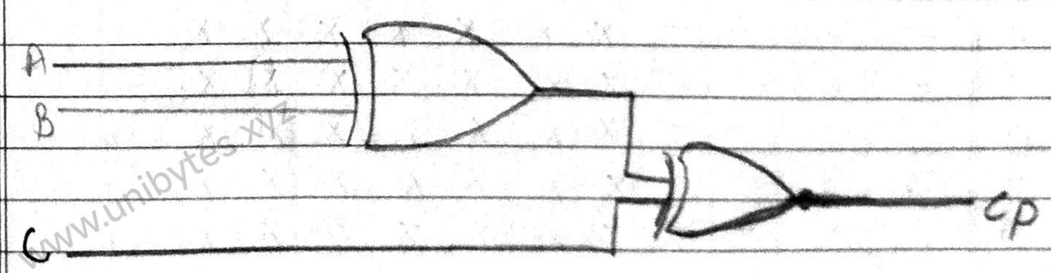
Truth table:

3-bit message			Odd parity bit generator
A	B	C	
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

K-Map:

A \ BC	00	01	11	10
0	1		1	
1		1		1

$= A'B'C' + A'BC + ABC' + ABC$
$= A'(B'C' + BC) + A(B'C + BC')$
$= A'(\overline{B \oplus C}) + A(B \oplus C)$
$= A \oplus B \oplus C$



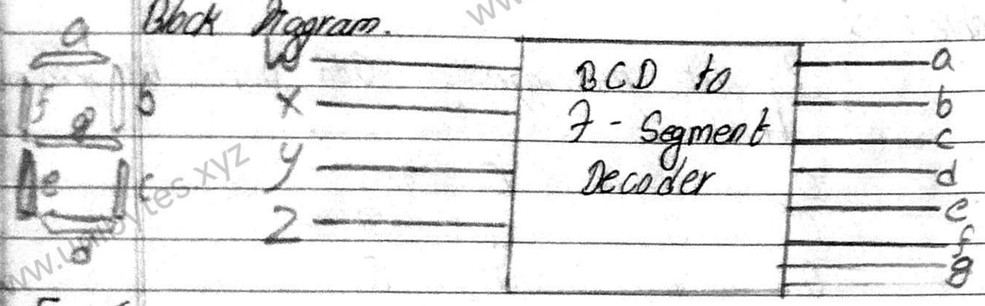
→ mcq maa sodna

Seven Segment Decoder

→ (Light Emitting Diode) vane paxi decoder

7 segment LED Displays numbers from 0 to 9. It is represented by a, b, c, d, e, f and g.

Block Diagram.



6 = 6

7 = 2

Truth table

number	w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10	–	–	–	–	x	x	x	x	x	x	x
11	–	–	–	–	x	x	x	x	x	x	x
12	–	–	–	–	x	x	x	x	x	x	x
13	–	–	–	–	x	x	x	x	x	x	x
14	–	–	–	–	x	x	x	x	x	x	x
15	–	–	–	–	x	x	x	x	x	x	x

i) K-MAP for a

wx \ yz	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

P_1 P_2 P_3

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

$A = z \odot x$

ii) K-Map for b

wx \ yz	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	x	x	x	x
10	1	1	x	x

IV I III II

$B = w + yz + \bar{x} + \bar{y}\bar{z}$

iii) C

wx \ yz	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	x	x	x	x
10	1	1	x	x

II I III IV

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

d)

wx \ yz	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	x	x	x	x
10	1	1	x	x

V I III IV II

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

e)

wx \ yz	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	x	x	x	x
10	1	0	x	x

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

f)

wx \ yz	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	x	x	x	x
10	1	1	x	x

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

g)

wx \ yz	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	x	x	x	x
10	1	1	x	x

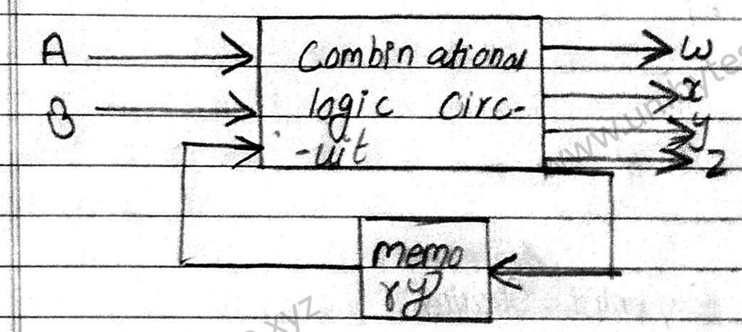
$g = w + y\bar{z} + x\bar{y} + \bar{x}y$

U. Imp

Chapter 4 Sequential Logic Circuits

The circuit whose output at any instant depends not only on the present input but also depends on ~~the~~ the past output is known as sequntial circuit. It has a memory unit to present the future output.

The examples of sequncial circuit are register, counter and flip-flop



~~Types of Sequential Circuit~~

i) ~~Asynchronous Sequential Circuit~~

Sequential circuit is a combinational circuit with memory. The output of sequential circuit depends upon present inputs and present state (past output). The information stored in sequential circuit represents present state. The present state and present inputs will define output and next state.

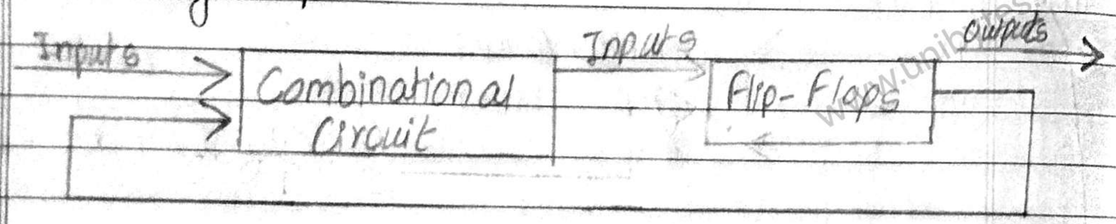
→ output can be changed at any instant of time by changing the input. Example asynchronous counter.

Types of Sequential Circuit

i) Asynchronous Sequential Circuits

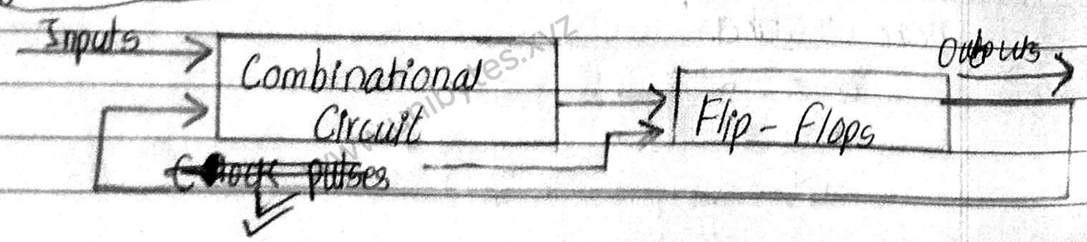
Asynchronous sequential circuits do not use clock signals but uses pulses instead. This circuit are faster than synchronous circuit. Therefore asynchronous circuit are used when speed of the operation is important. Output changes at discrete interval of time. Example: Flip-flop, asynchronous counter

Block diagram:)



ii) Synchronous Sequential Circuits

- Synchronous sequential circuits uses clock instead of pulse. It use logic gate and flip-flop storage devices.
- Synchronization is achieved by a timing device called clock pulse generator
- Since they wait for the next clock pulse to perform the next operation, these circuit are a bit slower compare to asynchronous.



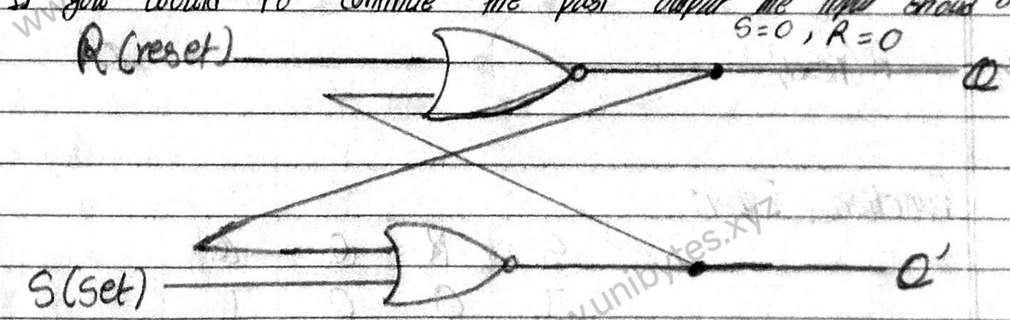
* Flip Flops

A flip flop is an electronic circuit with two stable state that can be used to store binary data. Flip flop are the application of logic gate. It is a circuit that maintains a state until directed by input to change the state. A basic flip flop can be constructed by using 4 'NAND' gate or 4 'NOR' gates.

i) S-R Flip-Flop using NOR gate

The design of the S-R flip-flop using NOR gate includes 2 inputs which is called set (S) and Reset (R) and there are 2 outputs Q and Q'. The logic diagram of S-R flip flop using NOR gate is below.

Conclusion: If you would to continue the past output the input should be S=0, R=0



Function Table:

S	R	Q	Q'	
1	0	1	0	(after S=1, R=0)
0	0	1	0	
0	1	0	1	
0	0	0	1	(after S=0, R=1)
1	1	0	0	

Q and Q' different huna parxa nama teo ~~hudina~~ ɣun hudina.

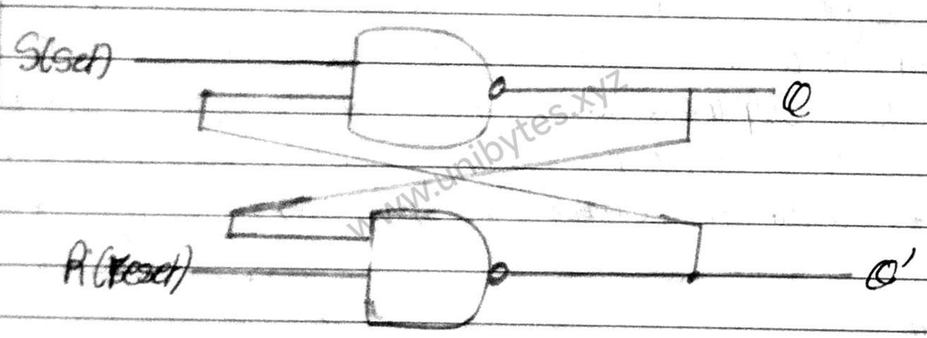
Case I : When S = 1 and R = 0 the output become Q = 1 and Q = 0. This is called set state.

Case II : When S = 0 and R = 1 the output become Q = 0 and Q' = 1. This called Reset state

Case III : When S = 0 and R = 0 the output is previous state.

Case IV : When S = 1 and R = 1 the output is Q = 0 and Q = 0 this state is called invalid state.

* S-R Flip using NAND gate



Function Tabel:

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	→ (after S=1, R=0)
0	1	1	0	
1	1	1	0	(after S=0, R=1)
0	0	1	1	

Note. Clocked S-R

S	R	
0	1	→ Q' = 1
1	0	→ Q = 1

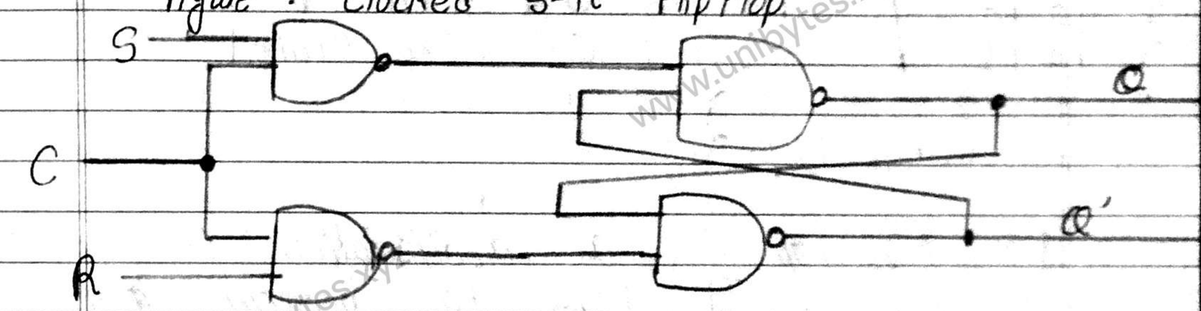
Book 780

1) S=1, R=0, Q=0 - - -

* Clocked S-R Flip Flop (OR Latch)

- It is also called Gated S-R flip flop
- The problems with S-R flip flop using NOR and NAND gate is the invalid stated when Q = 0, Q' = 0 and Q = 1, Q' = 1. To correct the circuit extra two NAND gates are added with C Clocked pulse

Figure : Clocked S-R Flip Flop



Another way of function table

Function Table:

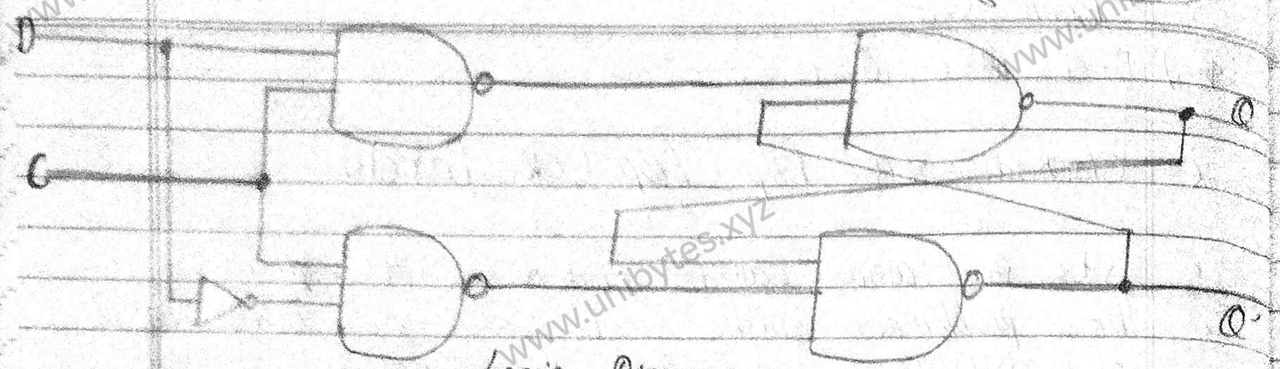
C	S	R	Q_{n+1}	
0	X	X	Q_n	
1	0	0	Hold	
1	0	1	0 → (Reset)	
1	1	0	1 → (set)	
1	1	1	Indeterminate	

C	S	R	Q_{n+1}	Q'	
0	X	X	Q_n	Q'	→ (No change)
1	0	1	0	1	
1	0	0	0	0	→ (No change)
1	1	1	1	1	→ (Indeterminate)
1	1	0	1	0	

→ Yo aph janxa

* D Flipflop

D Flipflop is the slight modification of clocked SR flipflop. If D = 1 then output goes to 1 and if D = 0 the output goes to 0.



Logic Diagram

ii) Function Table

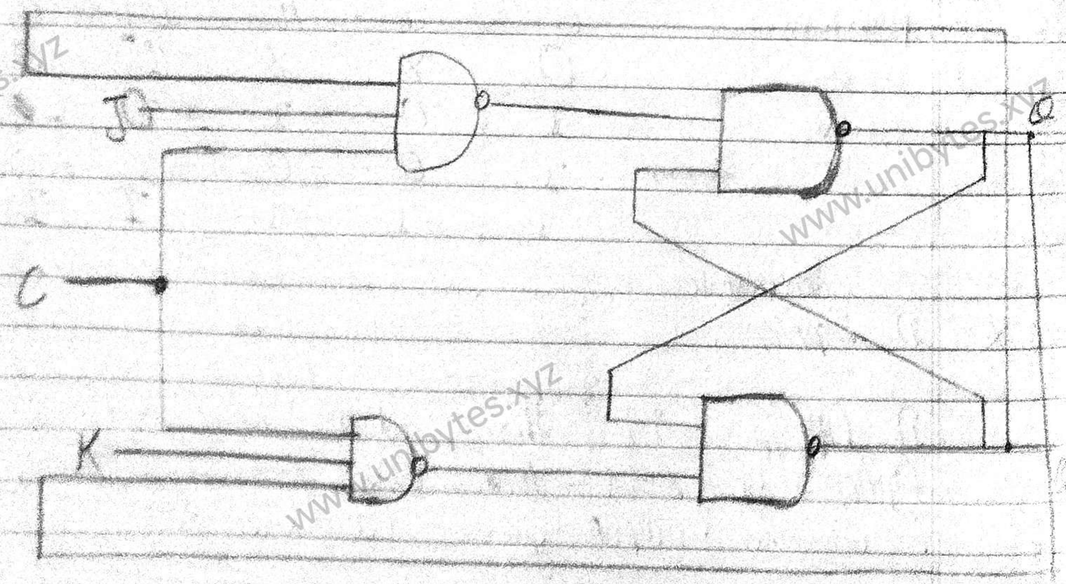
Excitation Table

Qn	D	Qn+1
0	0	0
0	1	1
1	0	0
1	1	1

C	D	Next Stage of Q
0	x	No change
1	0	Q=0 : Reset State
1	1	Q=1 : Set State.

4) J-K Flip Flop

J-K flip flop is a modification of the S-R flip flop. It helps to pervent from the Indeterminante condition which is occur in S-R flipflop.



Characteristics table

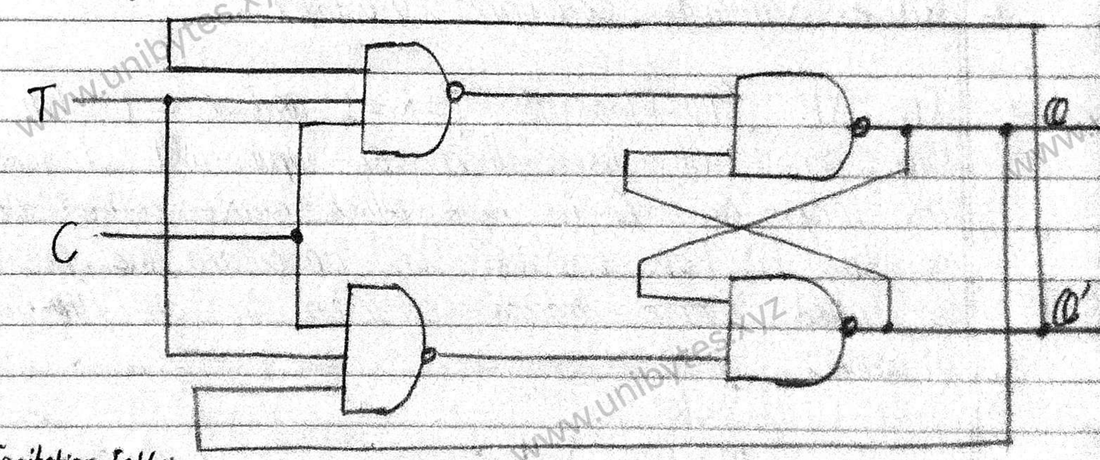
C	J	K	~~Qn~~	Qn+1
0	x	x		Memory
1	0	0		~~1 (Set)~~ memory
1	0	1		0
1	1	0		~~memory~~ 1 (Set)
1	1	1		Toogle

(same as SR But in 1, 1 it is giving toogle)

↳ 1,1 buyek are sabu SR ko output hun x a.

* T-Flip Flop

- T-Flip flop is simple version of JK flipflop
- In T-Flip flop both J and K are connected together and the clock pluse is seperated.
- When the clock pluse is high and T=1 (Input) then toggle is start



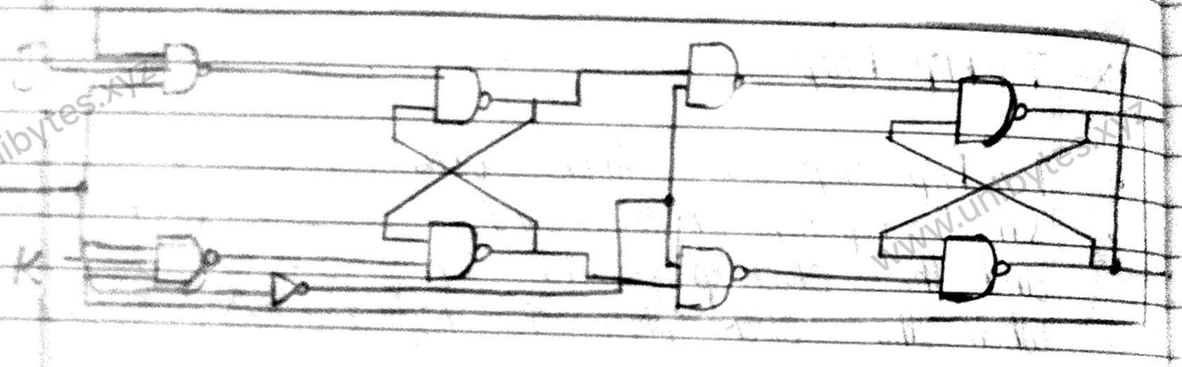
Excitation Table:

Qn	T	Qn+1
0	0	0
0	1	1
1	0	1
1	1	0

(Like as XOR)

* Master - Slave Flip Flop (V.V. imp)

- It has two separate flipflop, one acts as the master and another as slave.
- The output of master J-K Flip-Flop is used as the input to the slave Flip-Flop and vice-versa.
- The clock-pulse is given to the master J-K flipflop and send as to solve by using NOT gate.



* Race around condition - (V.V. imp)

- In JK Flip-Flop if $j = k = 1$ and $C = 1$ for a long period of time then Q output will toggle as long as C is high, which makes the output of the flip-flop unstable or uncertain. The problem is called race around condition in J-K flip flop.

Diagram.

i) J-K

ii) Master J-K

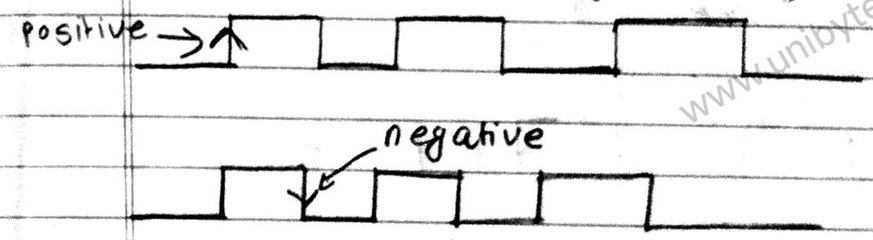
Note:

1) Toggle condition vane T flip flop ko gurni.
2) Race vane paile JK andi race of define and master.

* Edge trigger

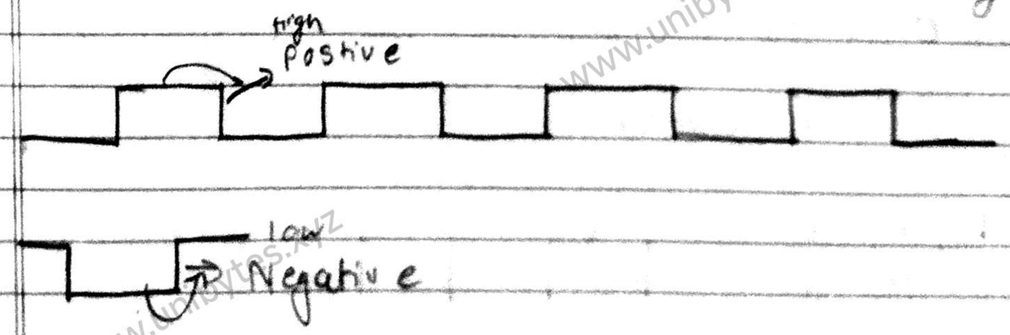
A triggering mechanism that is based on recongnisation of change in the signal level is called edge trigger. It is employed when ever it is important to identify the change in the input signals condition from low to high or high to low.

Edge triggering is frequently employed in synchronous circuit such as <u>counters</u>. It has positive and negative edge.



* level trigger

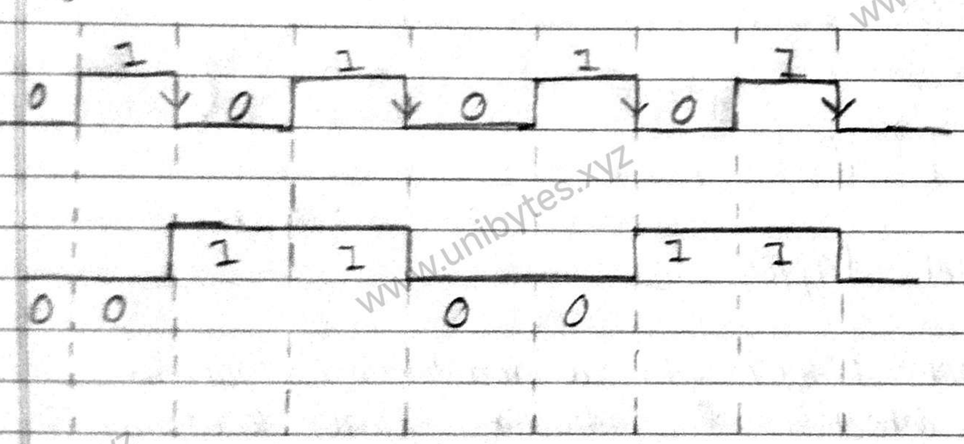
level trigger is a machanism that is based on detection of particular signal level. It is applied when detecting the input signal level at a specific moment rather than a change in state is required. It is used to control system



Chapter 5 Counter

* What is Counter → V.V.Imp

- A counter is a device which stores the numbers of times of particular event or process has occured often relationship to a clock signal.
- Counters are used for counting purpose. There are up counter and down counter.
- Up counter increases the count for every rising edge of the clock.



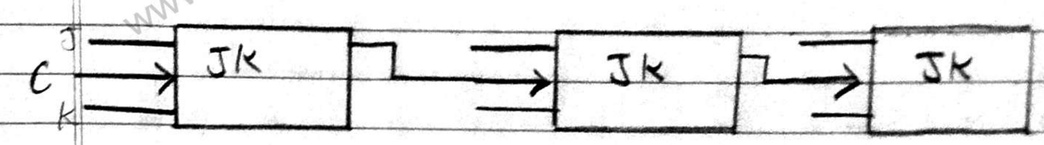
0 0 = 0	1 1 = 3
0 1 = 1	0 0 = 0
1 0 = 2	0 1 = 1

Types of Counter

1) Asychronous
2) Synchronous

1) Asynchronous Counter

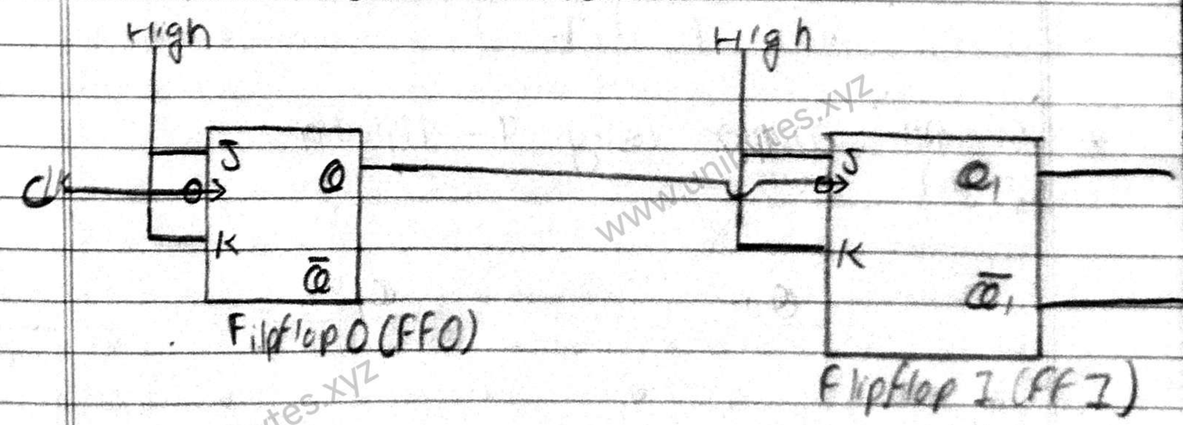
- In asynchronous counter universal clock is not used.
- The first flipflop is driven main clock and the clock input of remaining flipflop is driven by output of previous flip-flop.

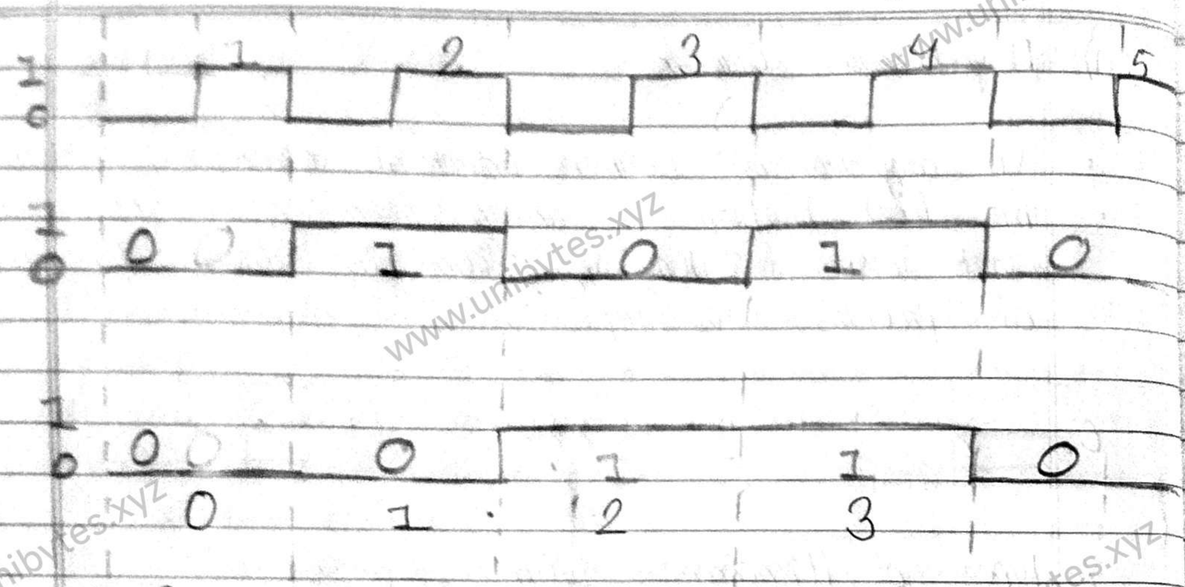


* Types of Asynchronous / Ripple Counter

i) 2-Bit Counter

- In a ripple counter output of first flipflop is provided as the clock to the second flipflop and it goes simultaneously.
- This circuit is simpler but the speed is slow.
- When the negative clock pulse is applied and input is given to FF0 (logic one) the output state of FF0 will toggle for every falling edge. It is also called mod 2 counter.

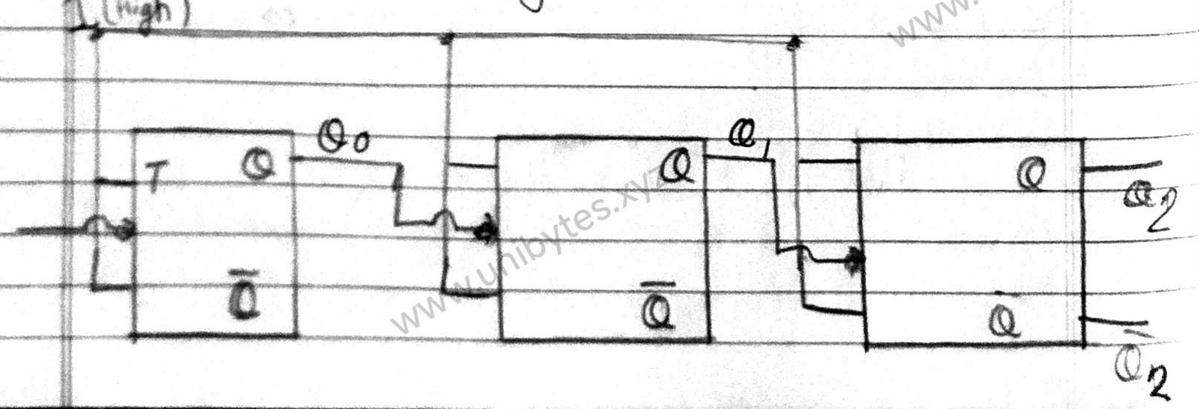




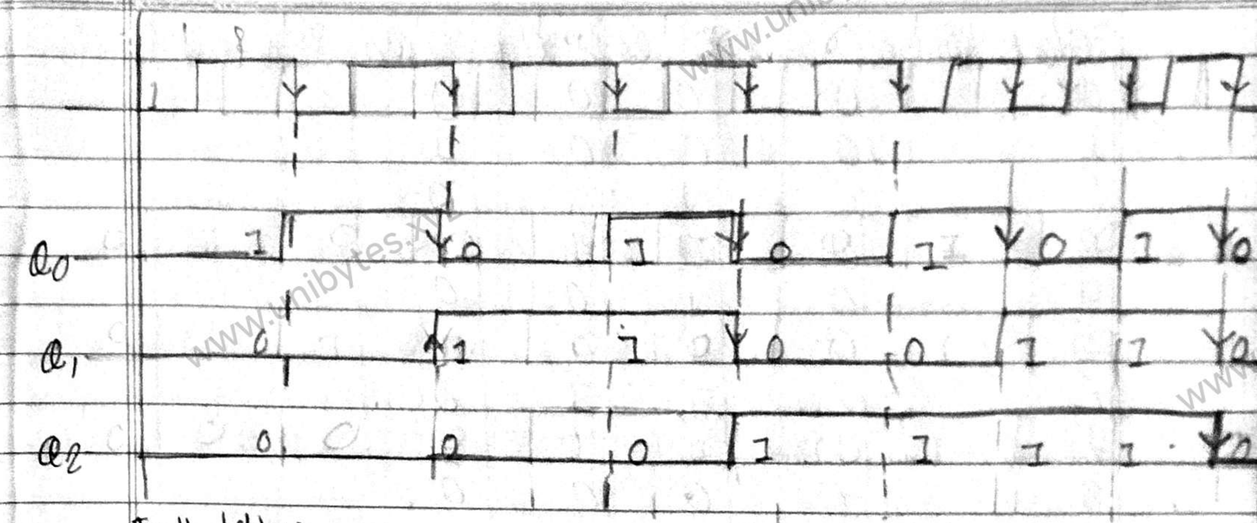
2-Bit up - counter Table

Clock Pluse	Q_1	Q_0
Initial	0	0
1	0	1
2	1	0
3	1	1
start again	0	0
	0	1
	1	0
	1	1

* 3 Bit counter using T- Flip flop



* UP gate 1
* Down gate 0



Truth table:

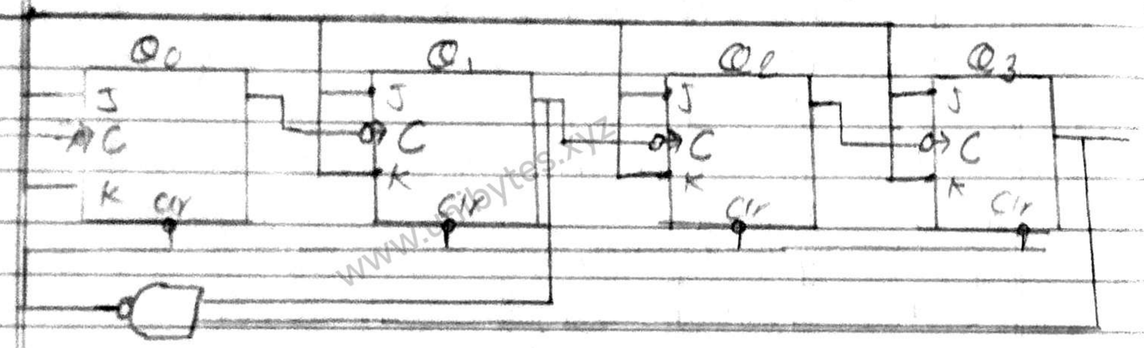
Clock Pluse	Q_2	Q_1	Q_0
Initial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (Reset)	0	0	0

* Asynchronous Decade Counters (Mod 10)

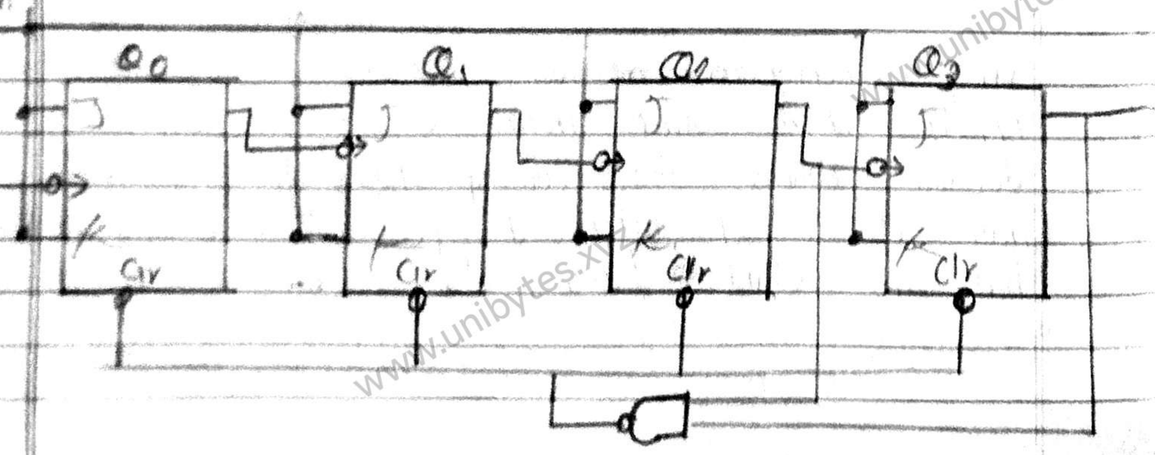
In asynchronous decade counters there are four flipflop which start from 0 to 15. But in decade counter the counting has to be done from 0 to 9 after 9 position the counter has to start from 0.

Clock pluse	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

High



* Modulus Twelve Asychronous Counter

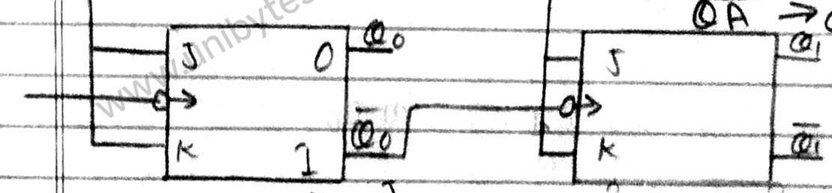


Clock Pluse	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0

* Design two bit down Counter

High

Note: $Q_A \rightarrow$ Clocked $\rightarrow$ -ve edge
$\overline{Q_A} \rightarrow$ Clock $\rightarrow$ +ve edge



Truth table: clock digram next page

~~Clock~~ State ~~Pluse~~	$\overline{Q_1}$	$\overline{Q_0}$
0	1	1
1	1	0
2	0	1
3	0	0
↳ (recycle)	1	1

Note:
2-Bit = 2 flip flop
3-Bit = 3 flip flop

www.unibytes.xyz

* UP/Down Counter of Asynchronous

M	Q_0	$\bar{Q}_0$	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

→ Control input say M
- M = 0 → upcounting → Q connected to the clock
- M = 1 → down counting → $\bar{Q}$ connected to the clock

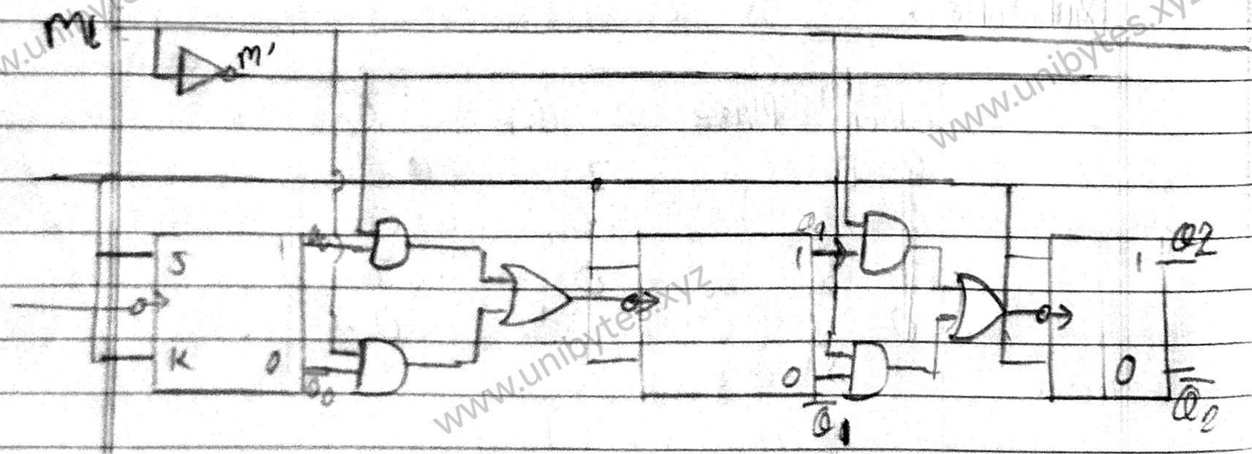
K-MAP of output y.

M \ Q_0Q_8	00	01	11	10
0	0	0	1	1
1	0	1	1	0

→ $\bar{M}Q_0$

→ $M\bar{Q}_0$

$\therefore y = \bar{M}Q_0 + M\bar{Q}_0$



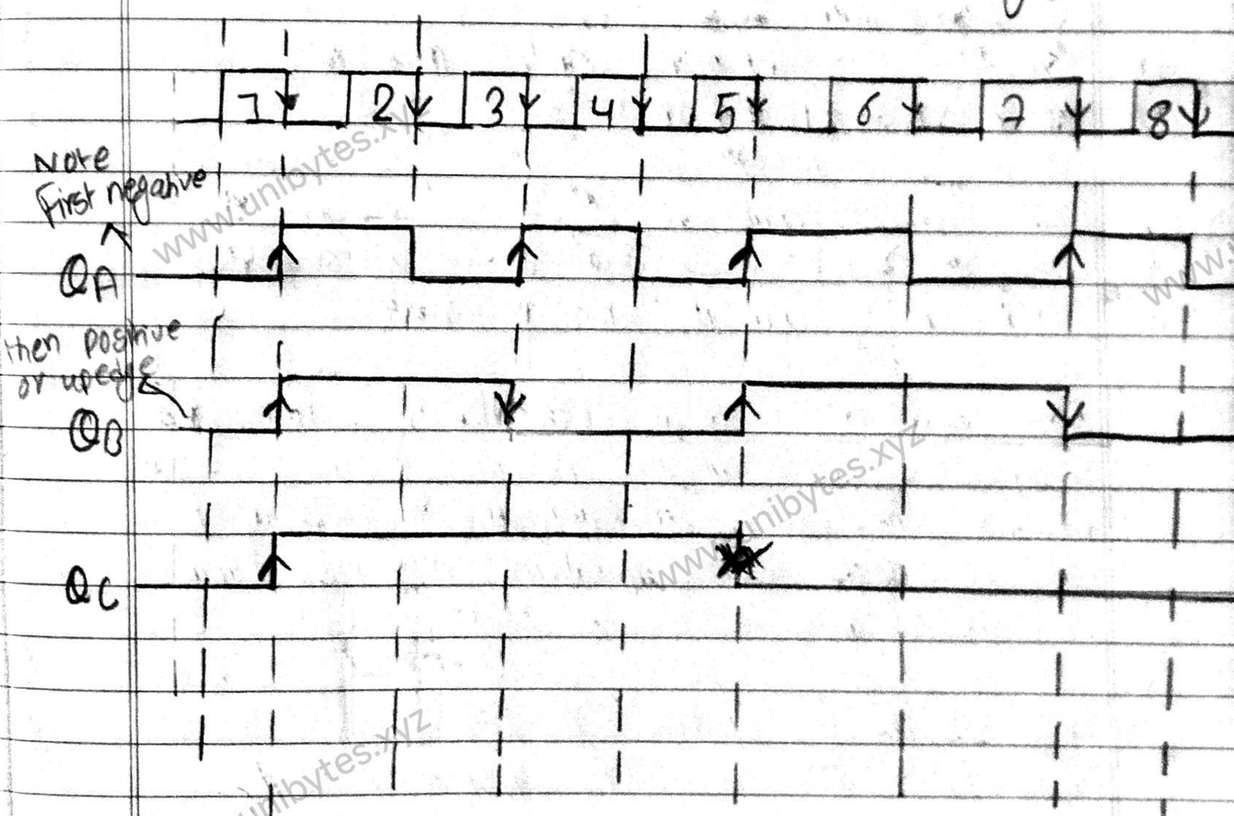
* Down gate 0
* UP gate 1
~~* Constant~~

The circuit above show three bit up-down counter if the value of mode (M) equals to '0'. The circuit acts as a up counter and if M equals to one(1). It act as a down counter.

In the truth table the output of y is the output of Q_0 when m=0 and the output of y is the output of $\bar{Q}_0$ when m=1. After receiving the truth table the K-Map of output is drawn. The equation of K-Map is $y = \bar{M}Q_0 + M\bar{Q}_0$.

Finally the circuit diagram of $y = \bar{M}Q_0 + M\bar{Q}_0$ is connect as the clock input of next flipflop.

Down Counter For 3 Bit, 2 Bit only (QA, QB)



Arko copy Herni

Chapter 6 Shift Registers

→ [V.vvvImp]

* What is register

ans A register is a device which stores the bit (1, 0). There are group of flip flops connected in a chain to form a register. It is also called set of N flip flops. It is mainly use for storing data. Each flip flop stores one bit. There are two basic function of register.

a) Data storage
b) Data movement

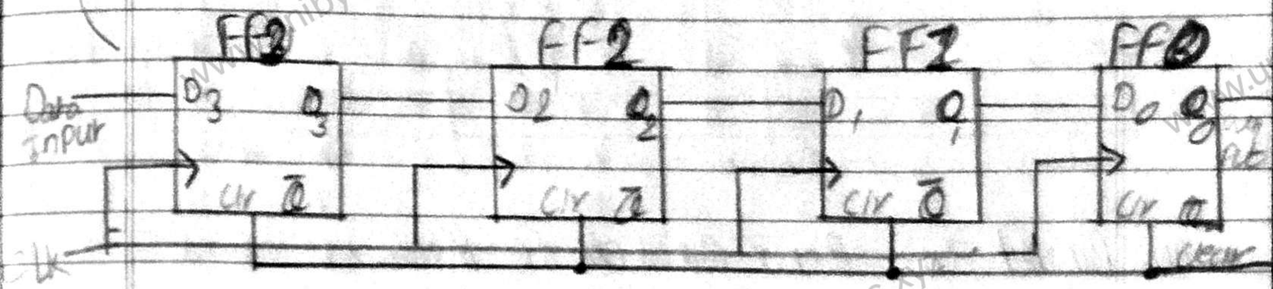
Types of Register

i) Serial In - Serial out shift Registers
ii) Parallel In - Parallel out " "
iii) Serial In - Parallel out " "
iv) Parallel In - Serial Out " "

i) Serial In - Serial out shift Registers

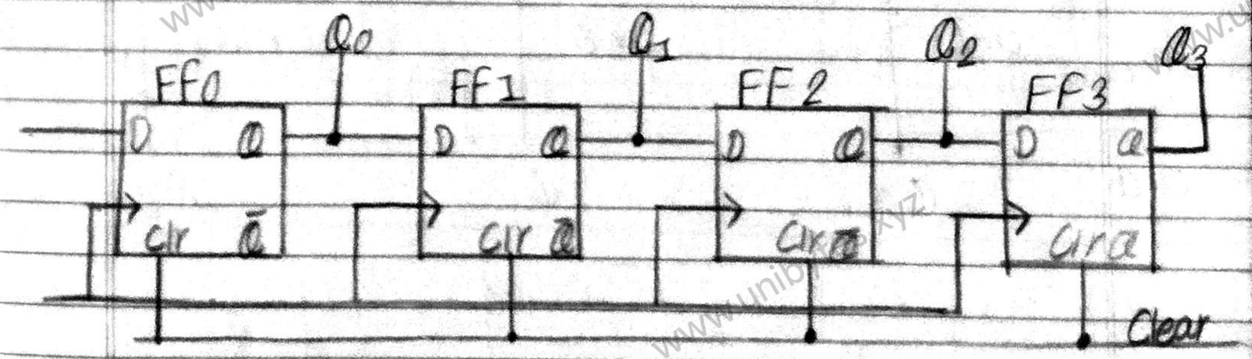
In serial In - serial out shift Register data are accepted serially that is one bit at a time. It produces the stored information in the serial form as a output. These operations of the circuit is as follow:

i) The register first cleared to zero
ii) The input data is then apply sequencially to the D flipflop as a input.
iii) During each clock pluse one bit transmitted from left to right.
iv) The ~~left~~ least significant bit of data has to be shifted through the register ~~has to be shifted~~ from FF3 to FF0.



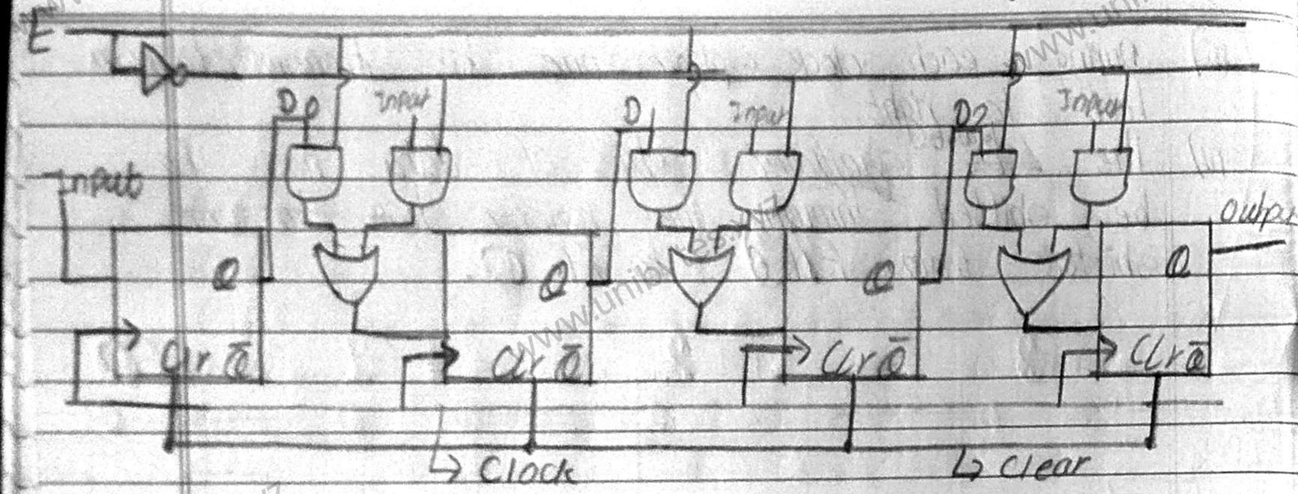
ii) Serial In - Parallel Out shift Register

In serial in - parallel out shift register data are input serially mostly from left to right flipflop and the data output is done simultaneously though all the flip-flop in a single clock pluse.



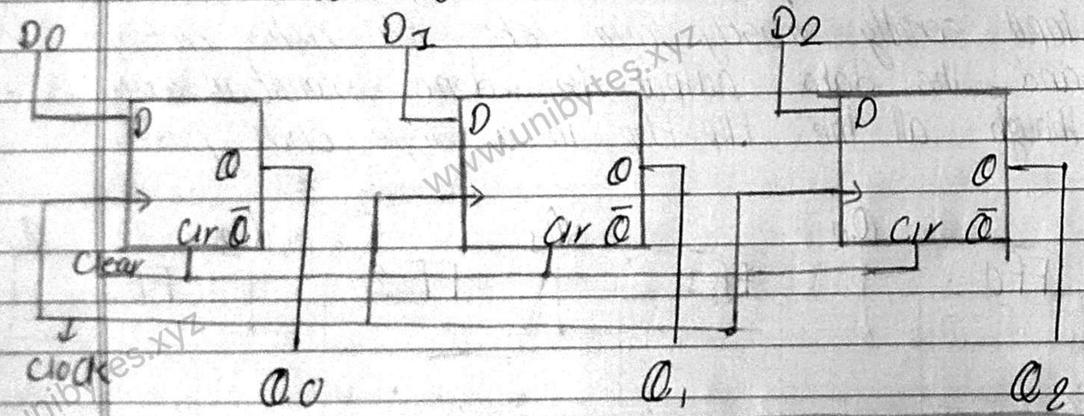
iii) Parallel In - Serial out shift Registers

In this data are insert in one go or in a single clock but output will occur serially.



iv) Parallel In - Parallel Out shift Register

In this all data bit appear on the parallel output immediately following the simultaneous entry of the data bits.



Clk	Q_3	Q_2	Q_1	Q_0

Chapter 9 Sequential Circuit Design

* Design of 3 - bit synchronous counter (using JK flipflop) → Imp

Steps :

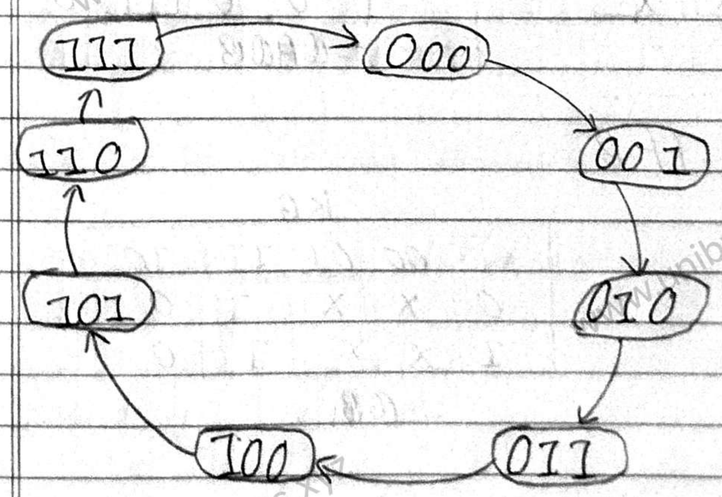
i) Number of flipflops

→ For 3 - bit there will be 3 flipflops

ii) Excitation Table of JK

Present state Q_n	Next state Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

iii) State Diagram



iv) Truth Table

Present state	Next state	FFC		FFB		FFA	
$Q_C Q_B Q_A$	$Q_C Q_B Q_A$	J_C	K_C	J_B	K_B	J_A	K_A
000	001	0	X	0	X	1	X
001	010	0	X	1	X	X	X
010	011	0	X	X	0	1	X
011	100	1	X	X	1	X	1
100	101	X	0	0	X	1	X
101	110	X	0	1	X	X	1
110	111	X	0	X	0	1	X
111	000	X	1	X	1	X	1

v) K-MAP

a) K-map for FF C

J_C

Q_C \ $Q_B Q_A$	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$= Q_A Q_B$

K_C

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	X	X	X
1	0	0	1	0

$= Q_A Q_B$

b) K-map for FF B

J_B

Q_C \ $Q_B Q_A$	00	01	11	10
0	0	1	X	X
1	0	1	X	X

$= Q_A$

K_B

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	X	1	0
1	X	X	1	0

$= Q_A$

Note: Whole K-Map vayo vane 1 hunxa equation.

c) K-Map for FF A

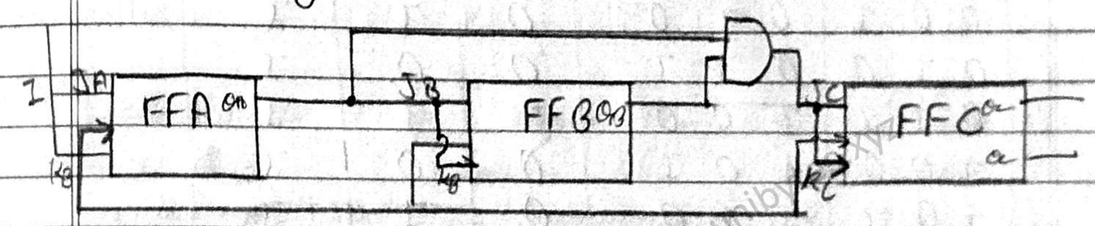
Q_C \ $Q_B Q_A$	00	01	11	10
0	1	X	X	1
1	1	X	X	1

$= 1$

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	1	1	X
1	X	1	1	X

$= 1$

vi) Logic Diagram



* Design of 3-bit synchronous counter (using T flip flop).

Steps:

i) Number of flip flop

→ There are 3 flipflop

ii) Excitation of T flipflop

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

iii) State Diagram
(same as J-K flipflop)

iv) Truth Table

Present state			Next state			Flip-Flop Input		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

iv) K-MAP

a) T_2

Q_2 \ Q_1Q_0	00	01	11	10
0	0	0	1	0
1	0	0	1	0

$= Q_1Q_0$

b) T_1

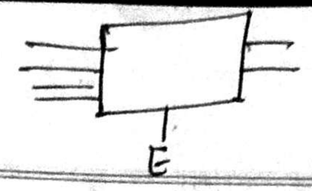
Q_2 \ Q_1Q_0	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$= Q_0$

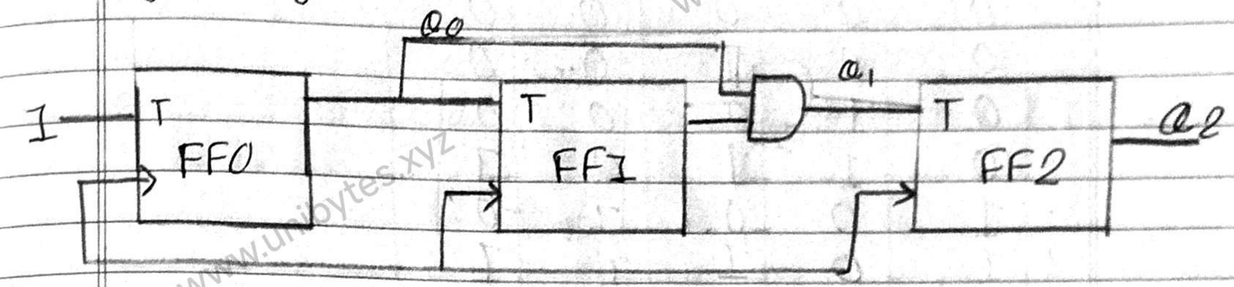
c) T_0

Q_2 \ Q_1Q_0	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$= 1$



vi) Logic Diagram



* What is PLA

Programmable Logic Array (PLA) is a fixed architecture logic device with programmable 'AND' gates followed by programmable 'OR' gates.
It is basically type of programmable logic device used to build a reconfigurable digital circuit.

Steps for PLA :

i) Design K-MAP of output y_1 and y_2 seperately
ii) Form the equation obtain from K-map design 'AND' gates.
iii) Obtain the programmable OR gate.

* Draw the PLA of the given.

A	B	C	Y_1	Y_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

i) Designing K-MAP

For Y_1

A \ BC	00	01	11	10
0	0	0	0	0
1	1	1	1	0

P_2 P_1

$P_1 = AC$, $P_2 = AB'$

$= P_1 + P_2$

ii) $= AC + AB'$

For Y_2

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	0

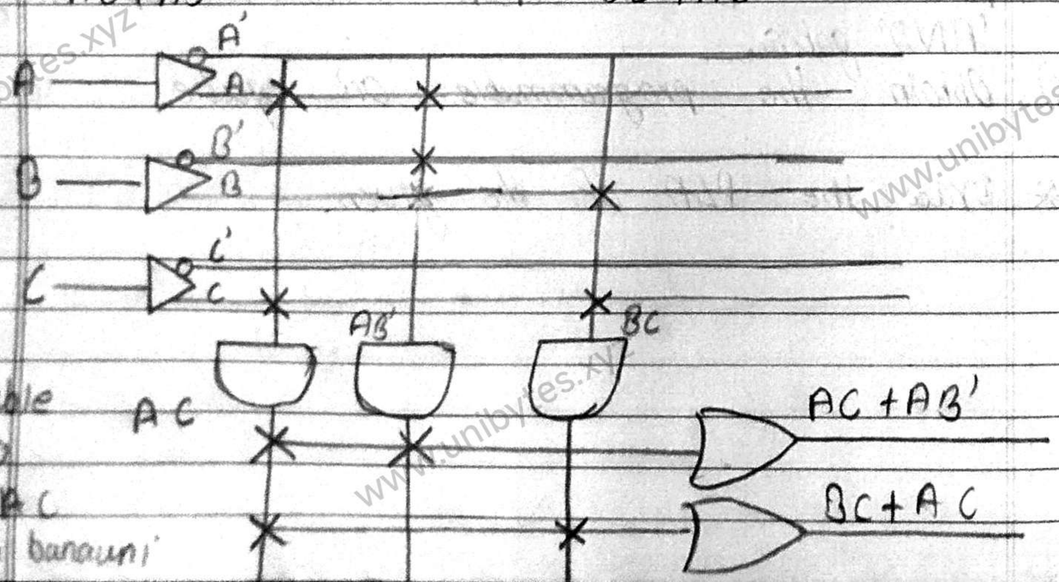
P_2 P_1

$P_1 = BC$, $P_2 = AC$

$= P_1 + P_2$

$= BC + AC$

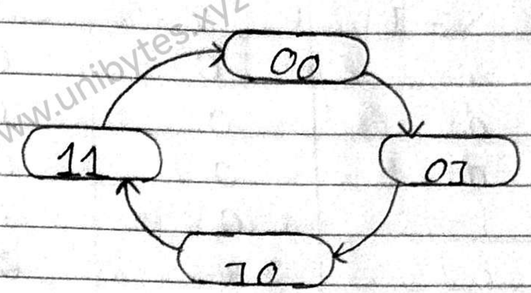
iii)



Note:
AC double xa so euta AC matra banauni

* What is state diagram?

ans State diagram is a pictorial representation of relationship between the present state, the input state, the next state and the output state of sequencial circuit.

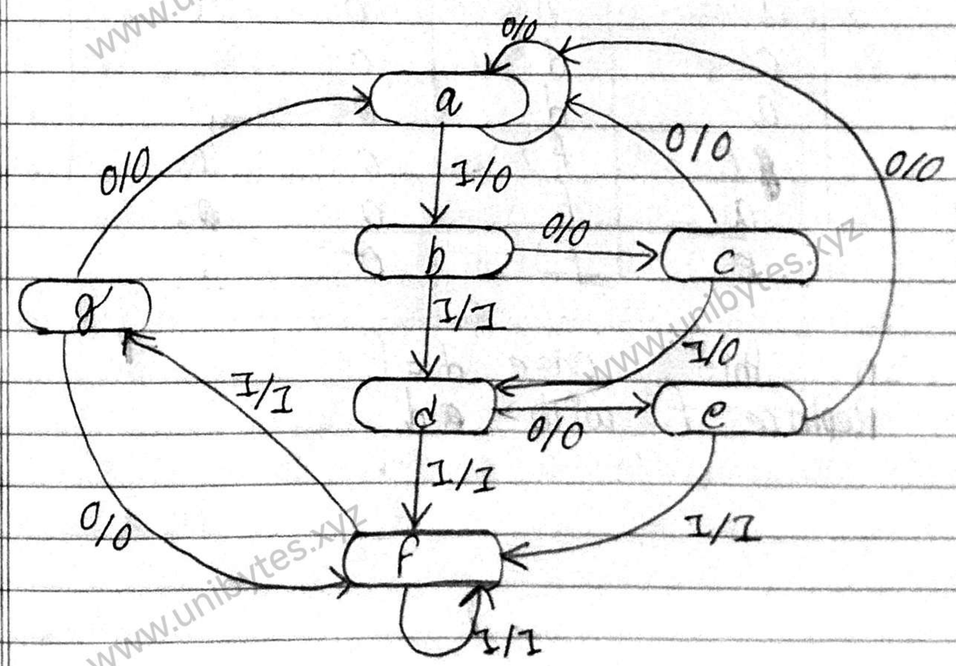


* State Reduction

ans It is the proceduer for reducing the number of states while keeping input output on change. It is helpful because it results in the reduction of number of flipflops to make circuit.

Quiz

* Reduce the state of the followin



Sol:

According to the diagram above,

Present State	Next State x=0	Next State x=1	Output x=0	Output x=1
a	a	b	0	0
b	c	d	0	1
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Step 2 : In this case e = g.
Replace g with e.

iii)

Present state	Next state x=0	Next state x=1	output x=0	output x=1
a	a	b	0	0
b	c	d	0	1
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

In this case d = f
Replace f with d

Present State	Next state x=0	Next state x=1	Output x=0	Output x=1
a	a	b	0	0
b	c	d	0	1
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

- In this case c = e
- Replacing e by c.

stop here

Present State	Next state x=0	Next state x=1	Output x=0	Output x=1
a	a	b	0	0
b	c	d	0	1
c	a	d	0	0
d	c	d	0	1

- In this case b and d are equal
- Replacing b by d.

Present state	Next state x=0	Next state x=1	Output x=0	Output x=1
a	a	b	0	0
b	c	b	0	1
c	a	b	0	0

- In this case a and c are equal
- Replacing c by a

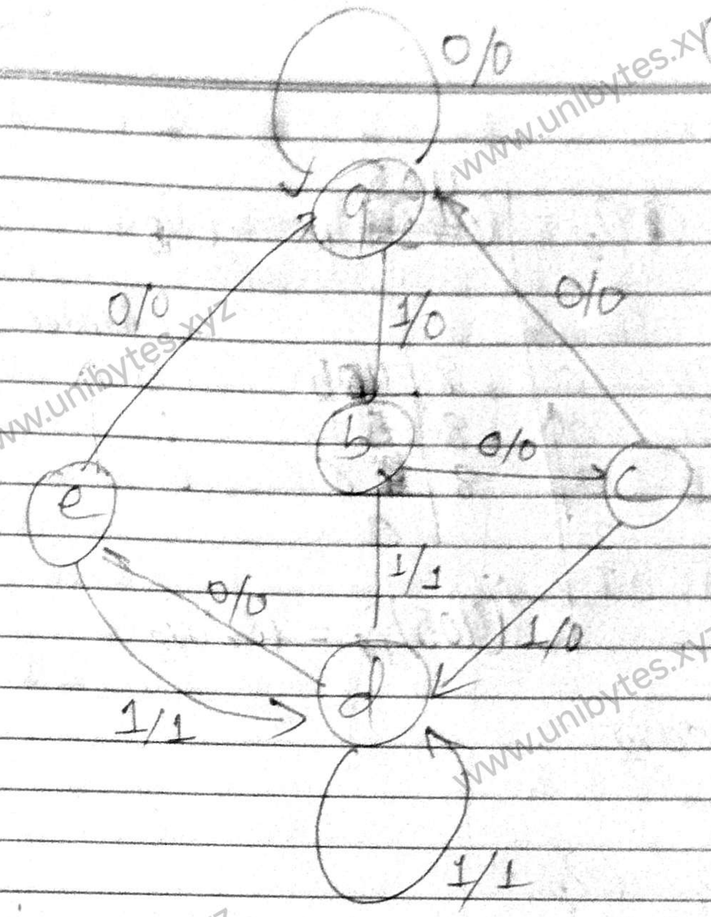
The Internship - movie name

Present state	Next state $x=0$	Next state $x=1$	Output $x=0$	Output $x=1$
a	a	b	0	0
b	a	b	0	1

- In this case $a = b$
- Replacing b with a

Present State	Next state $x=0$	Next state $x=1$	Output $x=0$	Output $x=1$
a	a	a		

Equation
Compliment
Sum up



Solved

Number System

* The base of radix of a number is defined as the number of digits used to represent the number system.

Types of number system

i) Denary and decimal are same.
ii) Binary
iii) Hexadecimal (A=10, B=11, C=12, ___ F)
iv) Octal

i) Conversion

* $(24)_{10} = (?)_2$

now;

		Reminder
2	24	0
2	12	0
2	6	0
2	3	1
2	1	1
	0	

$= (11000)_2$

$(\cancel{0.625}\ 405)_{10} = (?)_8$

		Reminder
8	405	5
8	50	2
8	6	6
	0	

$(405)_{10} = (625)_8$

* $(0.625)_{10} = (?)_2$

Fractional Decimal	Operation	Product	Fraction part	Integer
0.625	multiply by 2	1.250	0.250	1
0.250	multiply by 2	0.500	0.500	0
0.500	" "	1.000	0.000	1

$(0.625)_{10} = (101)_2$

Note:
i) Fraction part 0 na aauni samma garna parxa.
ii) Up to down lekhni (↓).

↗ Note same rule ~~for~~ of binary

ii) $(0.625)_{10} = (?)_8$

Fractional Decimal	Operation	Product	Fractional Part	Integer part
0.625	multiple by 8	0.5	0.5	0
0.5	" "	4.0	0.0	4

$\therefore (0.625)_{10} = (0.04)_8$

iii) $(495)_{10} = (?)_{16}$

16	495	15 → F
16	30	14 → E
16	1	1
	0	

$\therefore (495)_{10} = (1EF)_{16}$

*) $(1110) = (?)_{10}$

$= 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$
$= 8+4+2+0$
$= (14)_{10}$

* $(11.001)_2 = (?)_{10}$

~~$1\times2^1 \times 2^0 + 0\times2^1 +$~~
$= 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$
$= 2+1+0+0+1.25$
$= (4.25)_{10}$

$(521)_8 = (?)_{10}$

$= 5\times8^2 + 2\times8^1 + 1\times8^0$
$= 320 + 16 + 1$
$= 337$

$(127.54)_8 = (?)_{10}$

$1\times8^2 + 2\times8^1 + 7\times8^0 + 5\times8^{-1} + 4\times8^{-2}$
$= 64 + 16 + 7 + 0.625 + 0.0625$
$= (87.6875)_{10}$

• $(555)_{16} = (?)_{10}$
- same as octal multiply by 16

$55.005 = (?)_{10}$
- same as octal

* Binary to octal

Steps

i) Separate the given binary number into group of three digits (from right to left, add 0 in the left most side if require.

ii) Replace each group by its decimal equivalent

* $(10111)_2 = (?)_8$

~~010~~ New grouping into 3 bit i.e 010 and 111

010 = 2 / ∴ $(10111)_2 = (27)_8$

111 = 7 / ↳ 4 2 1

0 1 0 = 2

* Short cut

4	2	1
0	1	0

= 2

4	2	1
1	1	1

= 7

⇒ Mathi ko lai add garera nbr nikalni like 7 vaneko 4+2+1

* For B (11)

8	4	2	1
1	0	1	1

= (1011)

For, (1101)

8	4	2	1
1	1	0	1

= 13 = D

~~64 32 16 8 4 2 1~~

Q.n) 1011011 − 1010

(64	32	16	8	4	2	1)	
1	0	1	1	0	1	1	= 91
	−	1	0	0	1	0	= −18
1	0	0	1	0	0	1	73

carry mai jo na lekhni

yesr baddi janxa (like multiply by 2 garni

32 × 2 = 64, 64 × 2 = 128

1	1	0	1	0	0	1	1
− 0	1	1	0	1	1	1	0
0	1	1	0	0	1	0	1

Note double brow ligda (0 cut hunxa ani 1 matra ranxa

```
*        1 0 1 1
       × 1 0 1 1
         1 0 1 1
       1 0 1 1 x
     0 0 0 0 x x
   + 1 0 1 1 x x x
   -----------------
     1 1 1 1 0 0 1
```

Rules:

1 ÷ 1 = 1	1 × 1 = 1
1 ÷ 0 = not defined	1 × 0 = 0
0 ÷ 1 = 0	0 × 0 = 0
0 ÷ 0 = not defined	0 × 1 = 0

www.unibytes.xyz

10. Check answer apply ~~digit m~~ decimal method.

Eg: Divide 101011 by 110

```
110 ) 101011 ( 111
      -110
       1001
      -110
        111
       -110
          1 ← Re
```

∴ Reminder = 1
Quotient = 111

Imp Complement ⇒ Phone ma SS xa.

Binary	Decimal
- 1's	- 9's
- 2's	- 10's

* 9's complement

e.g 9's complement of 3 is : (9-3)=6
" " " 234 is : (999-234) = 765
Jati ~~dhari~~ ota digit hunxa teti ota 9 hunxa ani substract gami.

* 10's complement

~~eg~~ The 10's complement of decimal number can be obtained by adding 1 to the least significant digit of 9's complement of the number

Paile 9's complement nikalni ani 1 add gardini

10's complement of 3 : (9-3 = 6+1 = 7)
" " " 123 : (999-123 = 876+1 = 877)

3) 1's complement

Another method

1's complement of binary number is obtain by subtracting each bit by 1. We can get complement by simply replacing 1 by 0 and 0 by 1.

1's complement of 10010 = 01101.

4) 2's complement

2's complement of a binary number is obtain by 1 to the 1's complement of the number.

For e.g: 2's complement of 110101 is:

first we have to find 1's of 110101 = 001010

For 2's we have to add 1

001010 + 1 = 001011

* Subtraction of decimal number by using 9's complement.
- ~~Dec~~ SS. (Step imp xa hai bujna ko lagni).

Example of substraction of decimal numbers by using 9's complement.

i) Subtract $(123)_{10}$ from $(345)_{10}$.

Sol:

9's complement of 123 = 999 − 123 = 876

Adding it with minuend (345) = 876
+ 345
1221
↳ yo carry aako ho

= ~~221~~
~~+1~~
~~222~~

In the result there exist one carry i.e 1, so adding this carry to the remaining digits:

221
+ 1
222

Hence, $(222)_{10}$ is the required result after substracting $(123)_{10}$ from $(345)_{10}$.

ii) Subtracting $(345)_{10}$ from $(123)_{10}$

9's complement of 345
= (999 − 345) = 654

Adding it with minued (123) = 123
+654
777

In the result, there doesn't exist any carry, so calculating 9's complement of the result & prefix by (-ve) sign.

~~777~~

9's complement of 777 = (999 − 777) = 222

Hence, $(-222)_{10}$ is the required result after subtracting $(345)_{10}$.

* Substraction of decimal number by using 10's complement (S.S)

1) Substract $(123)_{10}$ from $(345)_{10}$

Sol:

10's complement of (123) = (999 − 123)
= 876 + 1
= 877

Adding it with minuend (345) = 877
+ 345
1222
↳ carry aayo

In the result, there exist one carry i.e 1 so ~~remaining~~ removing it to find the result

Hence, $(222)_{10}$ is the required result after substracting $(123)_{10}$ from $(345)_{10}$.

1) Subtract $(345)_{10}$ from $(123)_{10}$

10's complement of (345) = 999
-345
654 + 1
= 655

Adding it with minued (123) = 655 + 123
= 778

In the result, there doesn't exist carry, so calculating the 10's of the result add prefix by negative sign

~~999 - 778 =~~

10's complement of 778 = (999-778) = 221+1 = 222

Hence $(-222)_{10}$ is the required result after subtracting $(345)_{10}$ from $(123)_{10}$

* Subtraction of binary number using 1's complement

i) Substract $(110000)_2$ from $(111000)_2$

1's complement of (110000) = 001111

Adding it with minuend (111000) = 001111
+ 111000
1000111
carry aayo

In the result, there exist one carry ~~that~~ i.e 1, adding it with the remaining digits.

000111
+ 1
001000

Hence $(001000)_2$ is the required result after substracting $(110000)_2$ from $(111000)_2$.

ii) Subtract $(111000)_2$ from $(110000)_2$

1's complement of (111000) = 000111

Adding it with minuend (110000) = 110000
+ 000111
110111

In the result, there doesn't exist any carry. ~~making~~ So calculating 1's complement of the result and prefix by negative sign.

1's complement of 110111 = 001000

Hence $(-001000)_2$ is the required ~~req~~ result after subtracting $(111000)_2$ by $(110000)_2$.

* Subtraction of binary number using 2's complement.

i) Subtract $(110000)_2$ from $(111000)_2$

Sol:

1's complement of 110000 = 00111

2's complement of 110000 = 00111
+1
01000

Adding it with minuend (111000) ⇒ 0 110000
+ 111000
1 001000

In the result, there exist ~~can~~ one carry i.e 1. So, removing the carry.

Hence $(001000)_2$ is the required result after subtracting $(110000)_2$ from $(111000)_2$

ii) Subtract $(111000)_2$ from $(110000)_2$

2's complement of (111000) = 000111
+1
001000

Adding it with minuend (110000) = 110000
+ 001000
111000

10's = $r^n - 1$

9's = $(r^n - 1) - N$ / (~~r~~ $r^n - r^{-m}$) ~~+N~~

In the result, there doesn't exist carry so calculating 2's complement of the result and prefix by negative (-ve) sign.

2's complement of 111000 = 000111
+1
001000

Hence $(-001000)_2$ is the required result after subtracting $(111000)_2$ from $(110000)_2$.

Find 10's complement of 645

N = (number) = 645
r = (base) = 10
n = (total number given) = 3

now,

$r^n - N = 10^3 - N$
$= 1000 - 645$
$= 355$

9's $= (r^n - 1) - N = (10^3 - 1) - 645$
$= (1000 - 1) - 645$
$= 999 - 645$
$= 354$

9's complement of $(547.23)_{10}$

N = 547.23
n = 3
r = 10
m = (total number after decimal) = 2

I's ra 2's nii same xa phone maa hemi.

$= (r^n - r^{-m}) - N$

$= (10^3 - 10^{-2}) - 541.23$

$= 1000 - 0.01 - 541.23$

$= 452.76$

* 10's ~~of~~ complement of $(5663.5565)_{10}$

Sol:

$N = 5663.5565$

$r = 10$

$m = 4$ (pant aghi ko total number)

ndw;

$r^n - N = 10^4 - 5663.5565$

$= 10000 - 5663.5565$

$= 4336.4435$

* Substract $\underset{5}{72532} - \underset{4}{3250}$ using 10's

Sol:

making equal number of 3250 $= \underset{5}{03250}$

10's complement of 0350 =

$$\begin{array}{r} 9999\ 9 \\ -\ 03250 \\ \hline 96749 \\ +\ 1 \\ \hline 96750 \end{array}$$

Adding it with minued =

$$\begin{array}{r} 96750 \\ +\ 72532 \\ \hline 169282 \end{array}$$

In the result ~~carry~~ their exist carry, so removing it. $(69282)_{10}$ ans

another method:

$M = 72532$

$N = 03250$

10's complement of $N = r^n - N$

$= 10^5 - 03250$

$= 100000 - 03250$

$= 96750$

then, add the 10's complement wit M

$$\begin{array}{r} 72532 \\ +\ 96750 \\ \hline 169282 \end{array}$$

Since there exist carry, 10's complement of result is remove.

∴ The final result is 69282.

* Subtract $(101.10)_2 - (100.01)_2$ using 2's complement.

Sol:

At first integer part only.

$A = 101$

$B = 100$

2's complement of B (100) =

$$\begin{array}{r} 011 \\ +\ 1 \\ \hline 100 \end{array}$$

now;

$$\begin{array}{r} 101 \\ +\ 100 \\ \hline \end{array}$$

remove $\boxed{1}001$

For fraction part only

$A = 10,\ B = \cancel{0.}101$

2's complement of $B(0.1) = 10$

$$\begin{array}{r} 10 \\ +1 \\ \hline 11 \end{array}$$

now;

$$\begin{array}{r} 10 \\ +11 \\ \hline \end{array}$$

discard $\boxed{1}01$

$\therefore$ The final result $= (001.01)_2$

2) Substract $1010.\cancel{101} - 101.101$ using both 2's and 1's complement.

Sol:

For 2's complement.

Taking integre part only

$M\ A = 1010$
$N\ B = 0101$ } → equal garako xa.

2's complement of $B = 1010$

$$\begin{array}{r} +1 \\ \hline 1011 \end{array}$$

Adding with M = 1010

$$\begin{array}{r} +1011 \\ \hline 1)0101 \end{array}$$

remove carry 0101

~~$+x$ → (Adding carry)~~
~~0110~~

For fraction part

$M = 1010,\ N = 101$

2's complement of $N = 010$

$$\begin{array}{r} +1 \\ \hline 011 \end{array}$$

Adding with M;

$$\begin{array}{r} 110 \\ +011 \\ \hline \end{array}$$

remove → ~~1)010 0~~
remove → $1)001$

~~remove~~ $\therefore (0101.001)_2$ ans

For 1's complement

Taking integre part

$M = 1010,\ N = 0101$

1's complement of $N = 1010$

Adding with M = 1010

$$\begin{array}{r} +1010 \\ \hline 1)0100 \end{array}$$

$$\begin{array}{r} +1 \\ \hline 0101 \end{array}$$ → adding carry

For fraction part

$M = 110, N = 101$

1's Complement of $N = 010$

Adding with $M = 110$

$$\begin{array}{r} 110 \\ +010 \\ \hline 1000 \\ +1 \\ \hline 0001 \end{array}$$

→ adding carry

∴ The final result is (0101.001) ans

3) Substract 453.35 - 321.17 using both 9's and 10's Complement.

Sol:

For 9's complement

Integer part

$M = 453, N = 321$

9's complement of $N = \cancel{10 - N}\ 999 - 321 = 678$

Adding it with M

$$\begin{array}{r} 453 \\ +678 \\ \hline 1131 \\ +1 \\ \hline 132 \end{array}$$

→ adding carry.

For fraction part

$M = 35 \quad N = 17$

9's complement of $N = 99 - 17 = 82$

Adding with M,

$35 + 82 = 117$

again,

$$\begin{array}{r} 17 \\ +1 \\ \hline 18 \end{array}$$

∴ final result is 132.18

For 10's complement

For integre part

$M = 453, N = 321$

10's complement of $N = r^n - N = 10^3 - 321 = 1000 - 321 = \cancel{322}\ 679$

Adding with M

$$\begin{array}{r} 453 \\ +679 \\ \hline 1132 \end{array}$$

again,

In the result there exist carry, remove it, the result is 132

For fraction

$M = 35 \quad N = 17$

10's complement of $N = (99-17)+1 = 83$

Adding with $M = 35 + 83 = 118$

remove carry

∴ The final result is 132.18 ans

* Signed magnitude of 6 using eight bit

$+6 = 00000110$ / 1's complement (+6) $= 11111001$

$-6 = 10000110$ / 2's complement $= 11111001 + 1$

$= 11111010$

* Binary Code and Decimal (BCD)

BCD maa 3 add garni

Decimal	BCD	Excess-3	
0	0000	0100	0011
1	0001	0101	0100
2	0010	0110	0101
3	0011	0111	0110
4	0100	1000	0111
5	0101	1001	1000
6	0110	1010	1001
7	0111	1011	1010
8	1000		1011
9	1001		1100

Excess of 5 $= 0101 + 0011 = 1000$

i) Which one of the following is hexadecimal equivalent of $(5073.052)_8$?

now,

$(5073.052)_8$ Note '0' maa 3 ota jo '0' hunu

$= 101\,000\,111\,011 . 000\,101\,010_{000}$

$= A3B.150$ ans

* Binary to gray conversion

$(01001)_2 = (?)_{gray}$

$= 0 \oplus 1 = 1$

$= 1 \oplus 0 = 1$ (different 1)

$= 0 \oplus 0 = 0$ (∵ Similary '0')

$= 0 \oplus 1 = 1$

∴ $(01001)_2 = (1101)_{gray}$

* Represent 4673 in BCD

$4673 = (0100\,0110\,0111\,0011)_2$

8 4 2 1

$4 = 0100$

* Convert $(0110)_{BCD}$ into Excess-3

we have to add 3 in each digit (+3)

Step 1: Convert BCD to Decimal

$(0110)_{BCD} = (6)_{10}$

Date
Page

eg 8 ⇒ convert to excess-3 (binary)

$(9)_{10} = (1001)_2$

* Convert $(10010010)_{excess-3}$ to BCD

Sol.

1) Subtract 3 (0011) from each digit

1001	1010
−0011	−0011
0110	0111

$(10011010)_{excess-3} = (01100111)_{BCD}$

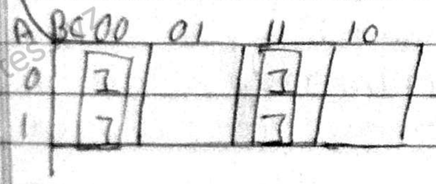
VV Imp/* Implement the following function using PLA

$F_1 = \sum_{(A,B,C)} m(0,3,4,7)$

$F_2 = \sum_{(A,B,C)} m(1,2,5,7)$ ⇒ last page maa truth table xa paile truth banayesw K-map garauni

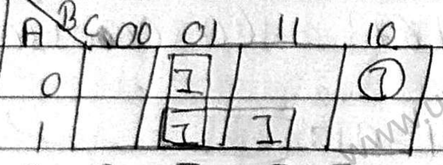
Sol:

K-MAP for F_1

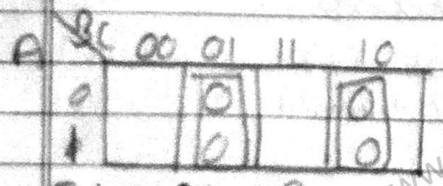


$F_1 = \bar{B}\bar{C} + BC$

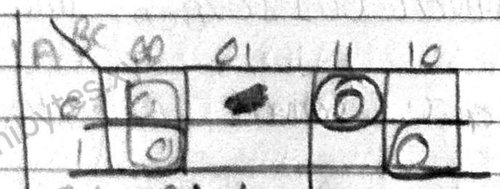
K-map for F_2



$F_2 = AC + \bar{B}C + \bar{A}B\bar{C}$



$F_1' = B'C + BC'$



$F_2' = B'C' + AC' + A'BC$

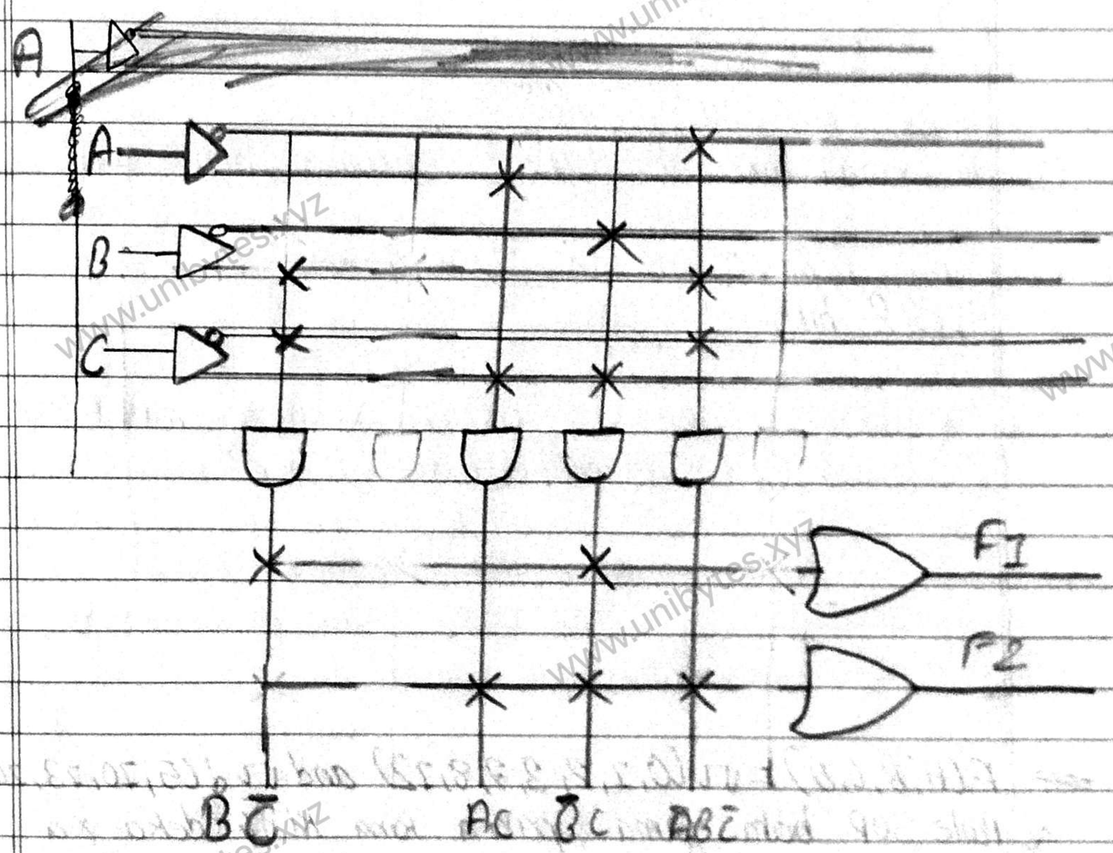
Note: $\bar{B} = 0$, $B = 1$ → Same for all yedi kunni term available xaina yeane — hand

Product Table

F1 maa $\bar{B}\bar{C}$ ra BC available xa so 1 na vako lai ●; Same for F2.

Product	Inputs			Outputs	
	A	B	C	F1	F2
~~$\bar{B}\bar{C}$~~	~~–~~	~~0~~	~~0~~	~~1~~	~~1~~
~~BC~~	~~–~~	~~1~~	~~1~~	~~1~~	~~–~~
AC	1	–	1	–	1
$\bar{B}C$	–	0	1	1	1
$\bar{A}B\bar{C}$	0	1	0	–	1
$B\bar{C}$	–	1	0	1	–

Logic diagram



→ $FC = F2 + F1$ = 4 minterms

$= AC + B'C + A'B\bar{C} + BC'$

1ms – 100 Hz A nibble consist of 4 bits

Date
Page

Aprko copy hani imp xa

$F = AC'D' + A'B'C + AC'D + AB'D$

i) When simplified with Boolean Algebra
$(x+y)(x+z)$ simplifies
$= x + yz$

ii) A logic circuit which is used to change a BCD number into an equivalent decimal number
= BCD to decimal decoder

iii) For JK flip-flop with J = 1, K = 0, the output after clock pulse will be
a) 0 b) 1 c) No change d) Toggle

iv) The term bit means
a) A small unit of data b) Binary digit c) A1 ora0

v) How many bit in 1 K. memory.
= 8,192 bits

vi) The device used to convert a binary number to 7 segment
→ Decoder (7-segment decoder jo xa topic nomemi)

vii) In the state diagram, the transition from a current state to the next state is determined by current state and the input

→ $F(A,B,C,D) = \sum(0,1,2,3,7,8,12)$ and $\sum d(5,10,13,14)$
→ Paile SOP bata gama parxa kina ki $\sum$ deko xa ani paxi SOP na vako thau maa balla POS garni yedi. Em diyo vane paile POS garni ani pos garni

Truth table

Minterm	A	B	C	F1	F2
0	0	0	0	1	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	0
7	1	1	1	1	1

www.unibytes.xyz