

Teacher:

Qamruzz Nabi

Text Book:

Book 2 (2020-2021) written by Dr. Md. Md. Ali

① Morris Mano

(13 Batch)

(10M 20, 2020) written by Dr. Md. Md. Ali

Digital Logic Design :

• it contains information :-

— Discrete elements

— Discrete digit

0 0 0 0 ————— 0

— letters

1 0 0 0 ————— 1

— alphabet, any meaningful symbols

0 1 0 0 ————— 2

signals \rightarrow (voltage, current)

1 1 0 0 ————— 3

• Digital electronics involves circuits and systems in

which there are only two possible states :

0 1 1 0 ————— 4

two different voltage levels : High & Low

two different current levels : switch on & off

Logic Levels :

voltage : High & Low

Binary Logic

(Notes 21)
Consists of binary variables (A, B, C, x, y, z) & logic

operations (AND, OR, NOT).

— possible values : 0, 1

: combination of variables

Digit

BCD (8921)

digit standard

0 ————— 0 0 0 0

0000

1 ————— 0 0 0 1

0001

2 ————— 0 0 1 0

(thousands, separator) — 2 longita

3 ————— 0 0 1 1

4 ————— 0 1 0 0

5 ————— 0 1 0 1

6 ————— 0 1 1 0

7 ————— 1 0 0 1

8 ————— 1 0 0 0

9 ————— 1 0 0 1

$$2^3 = 8 \text{ (0-7)}$$

$$2^5 = 32$$

$$2^7 = 128$$

$$2^4 = 16 \text{ (0-15)}$$

$$2^6 = 64$$

$$2^8 = 256$$

decimal stored into each cell for memory access

Decimal

Binary

Octal

Hexadecimal

0	0 0 0 0	0	0
1	0 0 0 1	1	1
2	0 0 1 0	2	2
3	0 0 1 1	3	3
4	0 1 0 0	4	4
5	0 1 0 1	5	5
6	0 1 1 0	6	6
7	0 1 1 1	7	7
8	1 0 0 0	10	8
9	1 0 0 1	11	9
10	1 0 1 0	12	A
11	1 0 1 1	13	B
12	1 1 0 0	14	C
13	1 1 0 1	15	D
14	1 1 1 0	16	E
15	1 1 1 1	17	F

* Binary Codes:

- Digital systems represent and manipulate not only binary numbers but also many other discrete elements of information.

6

1

卷之三

1

two bit code : 00, 01, 10, 11

three bit code :

✳ Decimal Codes :

Decimal Digit	BCD (8421)	to 0 1 0 Excess -3	minus 84-2-1	2421
0	0 0 0 0	+3 0 0 1 1	0 0 0 0	0 0 0 0
1	0 0 0 1	+3 0 1 0 0	0 1 1 1	0 0 0 1
2	0 0 1 0	1 1 1 0	0 1 1 0	1 0 0 0 / 0 0 1 0
3	0 1 0 0	0 0 0 1	0 1 1 0	0 0 0 1
4	0 1 0 1	1 0 0 1	0 1 1 0	0 0 0 1
5	0 1 1 0	0 1 0 1	0 1 1 0	0 0 0 1
6	0 1 1 1	0 1 0 1	0 1 1 0	0 0 0 1
7	1 0 0 0	1 1 0 1	1 1 1 0	1 1 1 1
8	1 0 0 1	0 0 1 1	0 1 1 0	0 1 1 1
9	1 0 1 0	1 0 1 1	1 1 1 0	1 1 1 1

DLD Lab

03-02-20



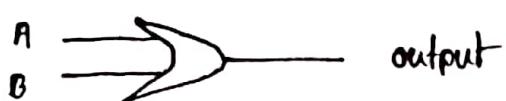
NOT Gate (IC 7404)

2004 2007 2008



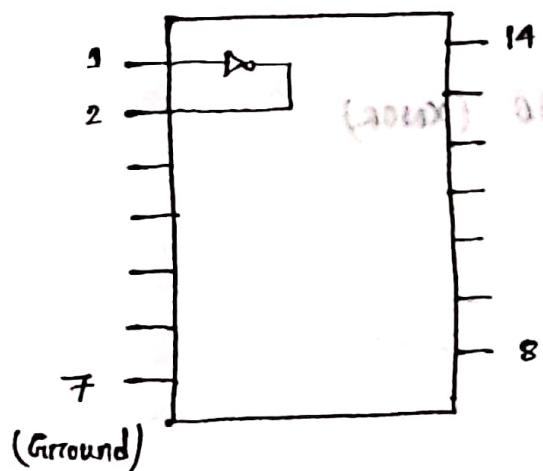
AND Gate (IC 7408)

2004 2007 2008



OR Gate (IC 7432)

2004 2007 2008



14 (VCC)

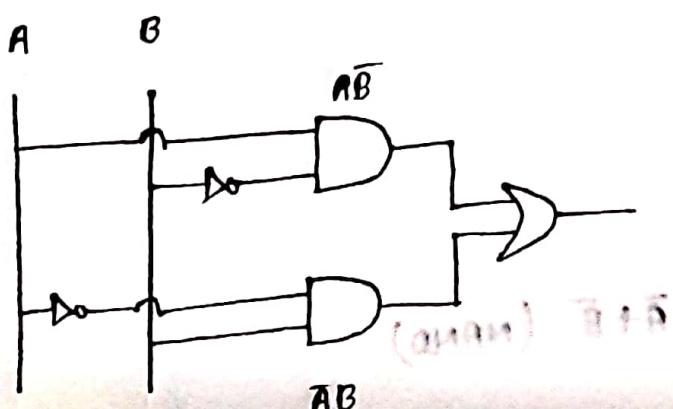
8 (2004) 2007 2008

8

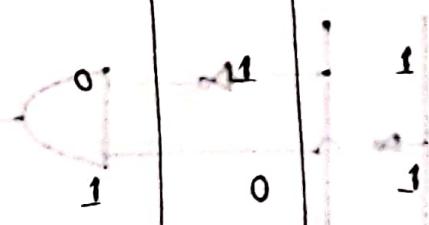
(Ground)

$$\text{XOR: } \overline{A+B} = A\bar{B} + \bar{A}B$$

A	B	output
0	0	0 0
0	1	1 1
1	0	1 0
1	1	0 0



(2004) 2007 2008



2004 2007 2008



XNOR : $\overline{A \oplus B}$

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

IC :

74186

NOR : $\overline{A+B} (= \overline{A} \cdot \overline{B})$

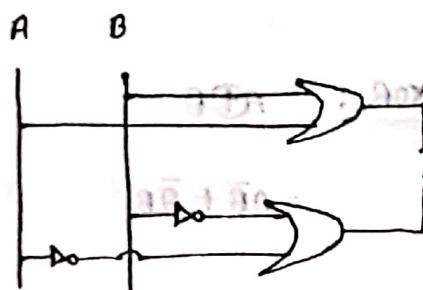
NAND

7400

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

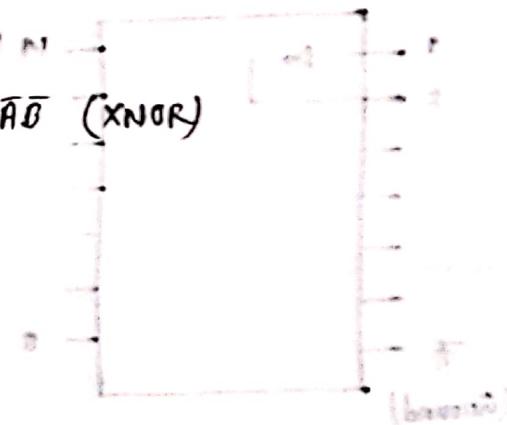
NOR

7402

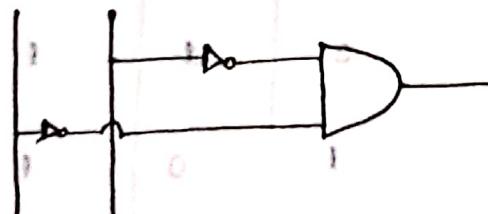


$$AB + \overline{A} \overline{B} \text{ (XNOR)}$$

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

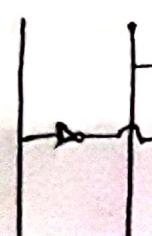


$$\overline{A} \cdot \overline{B} \text{ (NOR)}$$

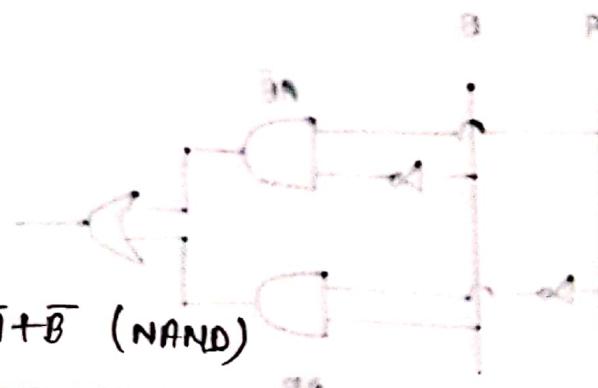


Inputs	A	B	Output
0 0	0	0	1
0 1	0	1	0
1 0	1	0	0
1 1	1	1	0

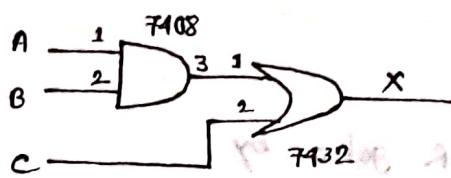
A B



$$\overline{A} + \overline{B} \text{ (NAND)}$$

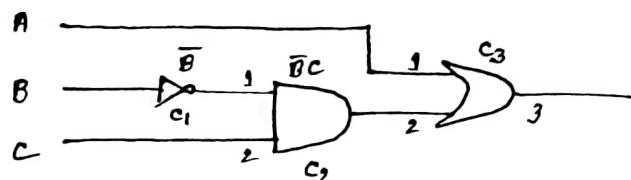


$$X = A \cdot B + C$$



A	B	C	output
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

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



1. $A + B \bar{C}$
: summa formingle
: svitsego

$$G = 7404$$

$$C_1 = 7408$$

$$C_2 = 7432$$

A	B	\bar{B}	C	output
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1

Class Code : x4xel7iz

240A - 2

Experiment - 01

① Use of Basic gates

② Construct NAND, NOR, XOR and XNOR gates by using basic gates.

③ Implement the following function using basic gates

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

Experiment Name :

Objective :

Truth Table :

Truth Table :

Circuit Diagram :

0 0 1 0 0

IC Diagram :

1 1 1 0 0

IC Requirement :

C_1, C_2 :

7408-1

0 1 0 0 1

Conclusion :

$C_3 : 7432-1$

BCD (Binary Coded Decimal) :

Binary BCD to Excess-3

<u>decimal</u>	<u>BCD (8421)</u>	weighted code
5	0101	
7	0111	below logic
10	00010000	

<u>Decimal</u>	<u>BCD</u>	<u>Excess-3</u>
5	0101	1000
+3		
1 2	10010	01000
+3		

BCD Addition :

Add 56 & 92 in BCD code

0101 0110

$$6+2 = 8 \quad (\text{1000})$$

1001 0010

$$5 + 9 = 14 \rightarrow 20$$

$$\begin{array}{r} 1110 \\ - 1000 \\ \hline \end{array}$$

we have to add 6

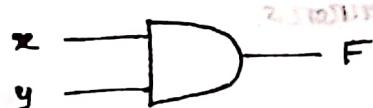
$$\begin{array}{r} 1110 \\ - 0130 \\ \hline 10100 \end{array}$$

④ Consider (+)ve logic

H \rightarrow 1		L \rightarrow 0		H \rightarrow 1		L \rightarrow 0	
x	y	x	y	x	y	x	y
0	0	1	0	1	1	0	0
0	1	0	1	1	0	1	1
1	0	0	0	0	1	1	0
1	1	1	1	0	0	0	1

④ Consider (-)ve logic

H \rightarrow 0		L \rightarrow 1		H \rightarrow 0		L \rightarrow 1	
x	y	x	y	x	y	x	y
1	1	1	0	1	0	0	1
1	0	0	1	0	1	1	0
0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	1



△ \rightarrow polarity indicator

att ni hi vino att zatamne tolli tiviso att =

polarity indicator signifies that negative logic is assumed

for the signal. att ni vino att zatamne tolli

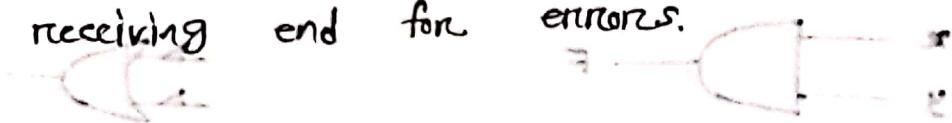
zatamne vino

④ That means positive logic AND gate is a negative logic

OR gate. vino att zatamne tolli vino

* Parity Generation & Checking:

- A parity bit is an extra bit included with a binary message to make the number of 1's either odd or even. The message including the parity bit is transmitted and then checked at the receiving end for errors.



decision making - - -

- the circuit that generates the parity bit in the message at transmitter is called a parity generator. The circuit that checks the parity in the message at receiver is called a parity checker.

parity bit \rightarrow Error detection code

Even Parity:

$$0 \ 0 \ 0 \ 0 - 1/0$$

$$0 \ 0 \ 0 \ 1 - 1$$

$$0 \ 0 \ 1 \ 1 - 0$$

$$0 \ 1 \ 0 \ 0 - 1$$

Parity Generator for 3 bitMessage

A	B	C	
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Parity (even)

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

[if one पाइ, आज्ञानि रहे]
even परिय

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

H.W (Practise): 3 bit एवं 4 bit odd parity

Receiver : 4 bit

Parity checker : 4 bit

A	B	C	D	P (Even Parity checker)			
0	0	0	0	0	1	1	0
0	0	0	1	1	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	1	1	0
0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	1
1	0	1	1	0	0	0	1
1	1	0	0	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	0	0	0	0	1
1	1	1	1	1	1	1	1

✳ Gray Code :

- Gray Code is a non weighted code as no weights are assigned to the bit positions
- to receive data from analog system it is required to convert into digital form before they are applied to a digital system.
- instead of using analog to digital converter it is sometime convenient to use Gray Code to represent the digital data when it is converted from analog data.
- the advantage is only one bit position is changed when going from one number to next.
- so that gray code converter can operate at high speed.

binary — 0 0 1 1 → 3

0 1 0 0 → 4

position of 1st bit is same

3 যেকোনো 4 এ যেতে 1st bit position change হচ্ছে, but
কোনো 4 এ 3 এ 1st bit position change নেই।

Gray code এর ফলে 1st bit position change হচ্ছে।
কোনো 4 এ 3 এ 1st bit position change নেই।

প্রথম bit কি হবে

কোনো 4 এ 3 এ 1st bit position change নেই।

প্রথম bit কি হবে কোনো 4 এ 3 এ 1st bit position change নেই।

কোনো 4 এ 3 এ 1st bit position change নেই।

প্রথম

কোনো 4 এ 3 এ 1st bit position change নেই।

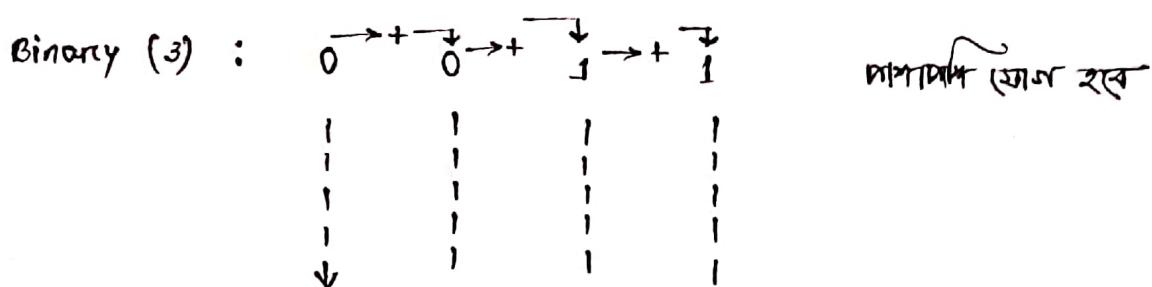
প্রথম bit কি হবে কোনো 4 এ 3 এ 1st bit position change নেই।

প্রথম bit কি হবে কোনো 4 এ 3 এ 1st bit position change নেই।

প্রথম

কোনো 4 এ 3 এ 1st bit position change নেই।

* Binary to Gray Code Conversion:



Gray Code (3) : 0 0 1 0

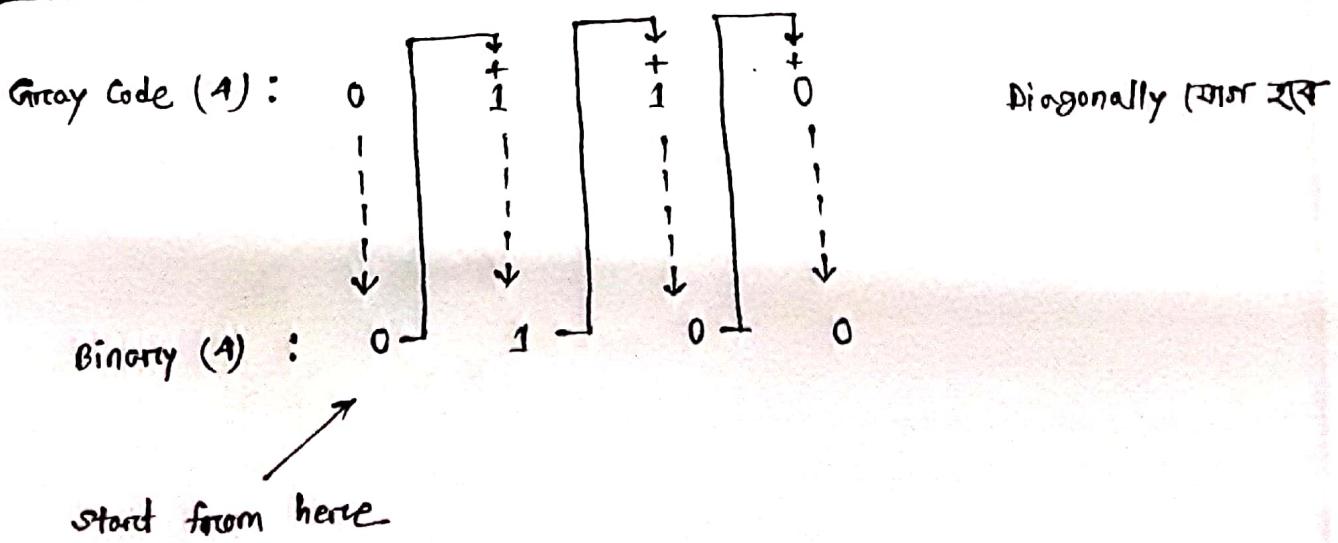
↑
start from here

Binary (4) : 0 1 0 0

Gray Code (4) : 0 1 1 0

3 : 0 0 1 0] Gray Code
 4 : 0 1 1 0]

* Gray Code to Binary Conversion:



④ Boolean Algebra :

operators : +, • (binary)

operators : not (unary) - /'

Laws:

Associative Law:

$$(x * y) * z = x * (y * z)$$

$$(x + y) + z = x + (y + z)$$

Commutative Law:

$$x + y = y + x$$

$$x * y = y * x$$

Distributive Law:

$$i) x + (y + z) = (x * y) + (x * z)$$

$$ii) x + (yz) = (x + y) * (x + z)$$

De Morgan's Law:

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

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

✳ i) $x + \bar{x} = 1$

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

(prove) \rightarrow for x is arbitrary

ii) $x + 1 = 1$

$$x \cdot 1 = x$$

→ Identity element

iii) $x + 0 = x$

$$x \cdot 0 = 0$$



↓ used contradiction

$$(x \cdot 1) + x = x + (x \cdot 1)$$

✳ Duality Principle :

$$(x + y) \cdot z = x \cdot z + y \cdot z$$

- Two part
- obtained from others if the binary operators and the identity element are interchanged.

This property of boolean algebra is called

duality principle. $(x \cdot y) + (x \cdot z) = x \cdot (y + z)$

$$(x \cdot 1) + (x \cdot 0) = (x \cdot 1) + x \cdot 0$$

$$x + 1 = 1$$

complement

$$x \cdot 0 = 0$$

identity element
operator

↓ used duality principle

$$0 \cdot 0 = 0$$

$$\textcircled{1} \quad x + xy = x$$

$$\textcircled{11} \quad (A+B)(A+C) = A+BC$$

$$\begin{aligned} & x + xy \\ &= x \cdot 1 + x \cdot y \\ &= x(1+y) \\ &= x \end{aligned}$$

$$\text{L.H.S.} = (A+B)(A+C)$$

$$\begin{aligned} &= AA + AC + AB + BC \quad [\text{distributive}] \\ &= A + AC + AB + BC \quad [A \cdot A = A] \\ &= A(1+C) + AB + BC \\ &= A + AB + BC \\ &= A(1+B) + BC = A + BC \end{aligned}$$

$$\textcircled{11} \quad xy + \bar{x}z + yz = xy + \bar{x}z$$

$$\text{L.H.S.} = xy + \bar{x}z + yz$$

$$= xy + \bar{x}z + 1 \cdot yz$$

$$= xy + \bar{x}z + yz(x + \bar{x})$$

$$= xy + \bar{x}z + xyz + \bar{x}yz$$

$$= xy + xyz + \bar{x}z + \bar{x}yz$$

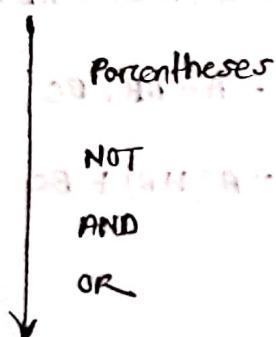
$$= xy(1+z) + \bar{x}z(1+y)$$

④ Complement of a function:

$$(A+B+C)' = A' \cdot B' \cdot C' \quad = (A+x)' = A' \cdot x' = A' \cdot (B+C)' = A' \cdot B' \cdot C'$$

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

operator precedence :



⑤ Representation of boolean Algebra:

i) standard form : \rightarrow sum of products $\rightarrow F = xy + yz$

\rightarrow product of sum $\rightarrow F = (x+y) \cdot (y+z)$

ii) canonical form :

\rightarrow Minterm (standard Product)

\rightarrow Maxterm (standard sum) \rightarrow complement of minterm

			Minterm		Maxterm	
x	y	z	Term	Designation	Term	Designation
0	0	0	$\bar{x}\bar{y}\bar{z}$	m_0	$x+y+z$	M_0
0	0	1	$\bar{x}\bar{y}z$	m_1	$x+y+\bar{z}$	M_1
0	1	0	$\bar{x}y\bar{z}$	m_2	$x+\bar{y}+z$	M_2
0	1	1	$\bar{x}yz$	m_3	$x+\bar{y}+\bar{z}$	M_3
1	0	0	$x\bar{y}\bar{z}$	m_4	$\bar{x}+y+z$	M_4
1	0	1	$x\bar{y}z$	m_5	$\bar{x}+y+\bar{z}$	M_5
1	1	0	$xy\bar{z}$	m_6	$\bar{x}+\bar{y}+z$	M_6
1	1	1	xyz	m_7	$\bar{x}+\bar{y}+\bar{z}$	M_7

$$m_0 + m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7$$

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

④ Obtain Canonical Form:

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

Truth Table:

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

For Minterm:

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

$$= m_1 + m_4 + m_5 + m_6 + m_7$$

$$= \Sigma(1, 4, 5, 6, 7) \rightarrow \text{sum of minterm}$$

Maxterm :

$$\begin{aligned} F' &= m_0 + m_2 + m_3 \\ (F')' &= (m_0 + m_2 + m_3)' \\ &= m_0' \cdot m_2' \cdot m_3' \\ &= M_0 \cdot M_2 \cdot M_3 \\ &= (A+B+C) (A+\bar{B}+C) (A+\bar{B}+\bar{C}) \\ &= \pi(0, 2, 3) \rightarrow \text{product of maxterm} \end{aligned}$$

④ $F = A + \bar{B}C$: Convert to canonical form

$$\begin{aligned} F &= A + \bar{B}C \\ &= A \cdot 1 \cdot 1 + \bar{B}C \cdot 1 \\ &= A (B+\bar{B}) (C+\bar{C}) + \bar{B}C (A+\bar{A}) \\ &= (AB + A\bar{B}) (C + \bar{C}) + \bar{B}C + \bar{A}\bar{B}C \\ &= ABC + ABC + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}C \\ &= m_7 + m_6 + m_5 + m_4 + m_1 \end{aligned}$$

$\therefore F = \Sigma(1, 4, 5, 6, 7) \rightarrow \text{Minterm}$

$$F' = m_0 + m_2 + m_3$$

$$(F')' = (m_0 + m_2 + m_3)'$$

$$= m_0' \cdot m_2' \cdot m_3'$$

$$= M_0 \cdot M_2 \cdot M_3$$

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

$$= \pi(0, 2, 3) \rightarrow \text{Maxterm}$$

* Design a square circuit for 2 bit:

number of terms = $2^2 = 4$

Input		w	x	y	z
A	B	0	0	0	0
0	0	$(A+B)$	$(B+A)$	$(B+A)$	$(A+B)$
0	1	$(A+B)$	$(B+A)$	$(B+A)$	$(A+B)$
1	0	$(A+B)$	$(B+A)$	$(B+A)$	$(A+B)$
1	1	$(A+B)$	$(B+A)$	$(B+A)$	$(A+B)$

Expression :

$$w = AB$$

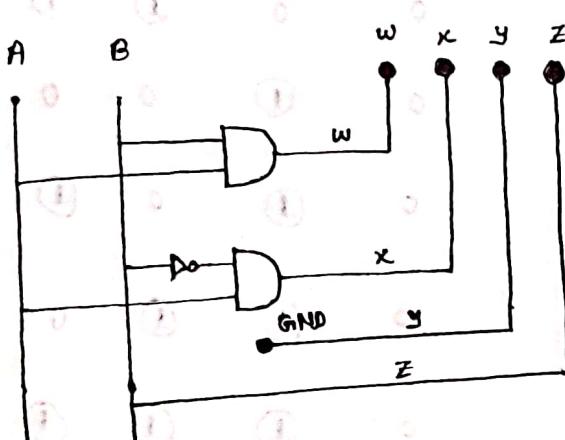
$$x = A\bar{B}$$

$$y = 0$$

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

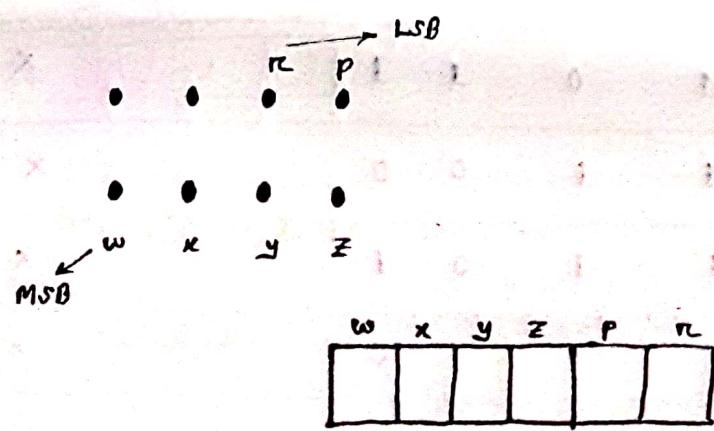
$$= B(\bar{A} + A) = B$$

Circuit Diagram :



Lab \rightarrow 3 bit

$$111 \rightarrow 7^2 = 49 \text{ (6 bit binary)}$$



④ Don't Care Condition:

Design 4 bit binary to BCD converter:

Input (Binary)				Output (BCD)			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	1	0	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

$$w = \Sigma(8, 9)$$

$$x = \Sigma(4, 5, 6, 7)$$

$$y = \Sigma(2, 3, 6, 7)$$

$$z = \Sigma(1, 3, 5, 7, 9)$$

$$d = \Sigma(10-15)$$

16-02-20

④ Gate Level Minimization : Karnaugh Map (K-Map)

3 variable

$\bar{A}BC$

$$F = \Sigma(1, 4, 5, 6, 7)$$

		00	01	11	10	
		$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$	
\bar{A}	0	0	1	3	2	
	1	4	5	7	6	

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

$$F = \Sigma(1, 4, 5, 6, 7)$$

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

1	1	1	1
1	1	1	1

ଅବ୍ୟାକ୍ଷମ

$$F = 1 \text{ (vcc)}$$

$$F = 0 \text{ (GND)}$$

$$Z = \{1, 3, 5, 7, 9\}$$

4 variable K-Map :

	$\bar{c}\bar{d}$	$\bar{c}d$	$c\bar{d}$	cd	
0	$\bar{A}\bar{B}$	0	1	3	2
1	$\bar{A}B$	4	5	7	6
3	AB	12	13	15	14
2	$A\bar{B}$	8	9	11	10

$$F = \{1, 3, 5, 6, 7, 12, 13\}$$

$\bar{e}\bar{o}$	$\bar{c}o$	co	$c\bar{D}$
$\bar{A}\bar{B}$	1	1	33
$\bar{A}B$	1	1	
$A\bar{B}$	x	x	x
$A\bar{B}$	1	x	x

$$Z = \{1, 3, 5, 7, 9\}$$

$$d = 5 (10-15)$$

$$Z = \bar{A}D + \bar{B}\bar{C}D$$

= D

$$F = \pm (1, 3, 5, 6, 7, 12, 13)$$

$$= \bar{A}D + \bar{A}BC + A\bar{B}\bar{C}$$

④ 5 variable & 6 Variable K-Map

$$F = \Sigma (0, 2, 4, 5, 9, 11) \rightarrow \text{sum of product}$$

Product of sum : $F = \prod (1, 3, 6, 7, 8, 10, 12, 15)$

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

$$\text{Product of sum} = A + B$$

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

→ Circuit Design করার

output পরিপন্থ করার

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

$$= 0$$

Experiment - 4

① Detect 4 bit prime number

প্রার্টি প্রাইম নম্বার ক্রেতানে এ নথাব।

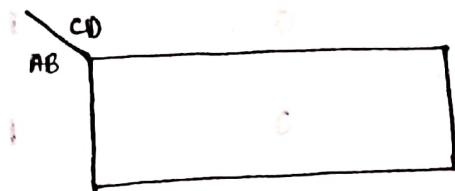
Input				Output (Prime)	
A	B	C	D	0111 not to found	
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0111	0111	
1	0	1	1	0	
1	0	1	0	1	
1	1	0	0	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	

$$F = \sum (2, 3, 5, 7, \dots)$$

↓

sum of product

Kmap :



Minimized Expression

$$F = \boxed{1}$$

Circuit Diagram :

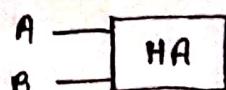
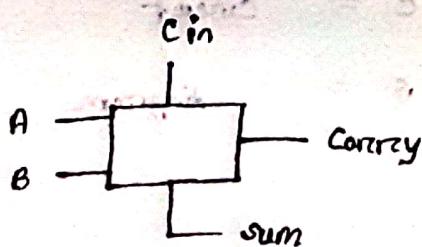
Requirements :

b) Full Adder using basic gates

Objective :

1 bit full adder \Rightarrow IC আছে কিন্তু তাম্বা Basic gate

দিয়ে বাজ করবো,



A, B যোগ করে
Half Adder

Truth Table :

Cin	A	B	Carry	Sum
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

$$C = \sum () \rightarrow \text{Kmap}$$

(Step 4) Now take Kmap (d)

$$\frac{1}{1}$$

$$S = \sum () \rightarrow \text{Kmap}$$

(Step 5) Minimized

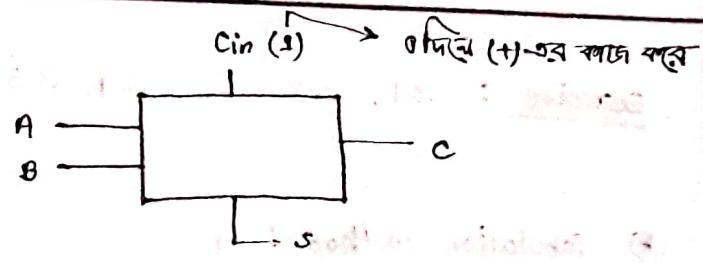


Carry Sum

C = circuit

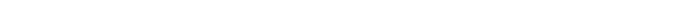
$s =$ diagram

$$\begin{array}{r}
 A - B \\
 \downarrow \\
 \overline{B} + 1 \\
 \downarrow \\
 \overline{A} + \overline{B} + 1 \\
 \hline
 \text{2's complement}
 \end{array}$$



Ex name : Design 1 bit full subtractor using Basic gates.

objective : Identified using, ownership (1)

Truth Table: 

Bin	A	B	Bout	Diff
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$\begin{array}{r}
 & 1 \\
 & 0 \\
 \hline
 & 0 \ 1 \\
 \swarrow & \searrow \\
 \text{Borrow out} & \text{Difference}
 \end{array}$$

20-02-20

Exercise : 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.9, 3.11, 3.13, 3.15

① Tabulation method :

— known as Quine-McCluskey (QM) Method

— Two parts

1) Determine prime implicants

2) Selection of prime implicants

② Simplify the following boolean function using tabulation method

$$F = \Sigma (1, 4, 6, 7, 8, 9, 10, 11, 15)$$

$$J \rightarrow 0 \ 0 \ 0 \ 1$$

$$4 \rightarrow 0 \ 1 \ 0 \ 0$$

$$6 \rightarrow 0 \ 1 \ 1 \ 0$$

$$7 \rightarrow 0 \ 1 \ 1 \ 1$$

$$8 \rightarrow 0 \ 1 \ 0 \ 0$$

$$9 \rightarrow 1 \ 0 \ 0 \ 1$$

$$10 \rightarrow 1 \ 0 \ 1 \ 0$$

Tabulation Method : *chart of solutions* *designed using* *method*

		w	x	y	z			w	x	y	z		w	x	y	z	
✓	1	0	0	0	1		✓	1, 9		0	0	1		8, 9, 10, 11		1 0, --	
✓	4	0	1	0	0		✓	4, 6		0	1	0		8, 10, 9, 11		1 0, --	
✓	8	1	0	0	0		✓	8, 9		1	0	0	-				
✓	6	0	1	1	0		✓	8, 10		1	0	-	0				
✓	9	1	0	0	1		✓	6, 7		0	1	1	-				
✓	10	1	0	1	0		✓	9, 11		1	0	-	1				
✓	7	0	1	1	1		✓	10, 11		1	0	1	-				
✓	11	1	0	1	1			✓	7, 15		1	1	1	1			
✓	15	1	1	1	1			✓	11, 15		1	1	1	1			

Determination of prime implicants :

$$F = \bar{x}\bar{y}z + \bar{w}x\bar{z} + \bar{w}xy + xyz + wyz + w\bar{x}$$

selection of prime implicants / Essential implicants :

	1	4	6	7	8	9	10	11	15
$x'y'z$ 1, 9	⊗					x			
$w'xz'$ 4, 6		⊗	x						
$w'xy$ 6, 7			x	x					
xyz 7, 15				x					x
wyz 11, 15					0			x	x
wx' 8, 9, 10, 11		1			⊗	x	⊗	x	

Essential Implicants, $F = x'y'z + w'xz' + wx' + xyz$
 $\hookrightarrow (7, 15)$

classical string to minimization

Exp 5

Binary to Gray Code

a) Design a 4bit Binary to Gray Converter

Truth Table:

Input (Binary)

(Gray) Code

A	B	C	D
0	0	0	0
0	0	0	1

Output (Gray Code)

w	x	y	z
0	0	0	0
0	0	0	0

$$w = \Sigma () \xrightarrow{\text{KMap}}$$

$$x = \Sigma () \quad " \quad \text{Gray code generated 11 bits}$$

$$y = \Sigma () \quad "$$

$$00000 = 0$$

$$z = \Sigma () \quad " \quad 10000 = 1$$

$$01000 = 2$$

b) Design a 4 bit Gray to Binary Converter

Input (Gray)

Output (Binary)

A	B	C	D	w	x	y	z
0	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	1	1

c) Design a 4 bit 2's complement circuit

Truth Table: दिए गए वाले नंबरों का नियम ले जायेगा

<u>Input (Binary)</u>				<u>1's Complement</u>				<u>2's Complement</u>			
A	B	C	D	W	X	Y	Z	W	X	Y	Z
0	0	0	0	1	1	1	1	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	1	0	0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	1	1	0	0	0
0	0	0	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0

*** Final ना भालूरा

④ QM Method

$$F = \sum (0, 2, 6, 10, 12) \quad d = \sum (1, 3, 5, 7) \quad (\) \oplus = w$$

Step 1: Determine prime implicants () \oplus = x

$$0 - 0000$$

$$1 - 0001$$

$$2 - 0010$$

$$3 - 0011$$

$$5 - 0101$$

$$6 - 0110$$

$$7 - 0111$$

$$10 - 1010$$

$$12 - 1100$$

	W	X	Y	Z	W	X
$\sqrt{6}$	0	0	0	0	$\sqrt{6}, 1$	0
$\sqrt{7}$	0	0	0	1	$\sqrt{6}, 2$	0
$\sqrt{2}$	0	0	1	0	$\sqrt{1}, 3$	0
$\sqrt{3}$	0	0	1	1	$\sqrt{1}, 4$	0
$\sqrt{5}$	0	0	1	1	$\sqrt{2}, 3$	0
$\sqrt{6}$	0	1	0	1	$\sqrt{2}, 4$	0
$\sqrt{10}$	0	1	1	0	$\sqrt{2}, 5$	0
$\sqrt{12}$	1	0	1	0	$(2, 10)$	$[-, 0]$
	1	1	0	0	$\sqrt{3}, 7$	0
	1	0	0	0	$\sqrt{5}, 7$	0
	1	1	1	1	$\sqrt{6}, 7$	0
$\sqrt{7}$	0	1	1	1		

	w	x	y	z
(0, 1, 2, 3)	0	0	-	-
(0, 2, 1, 3)	0	0	-	-
(1, 3, 5, 7)	0	-	-	1
(1, 5, 3, 7)	0	-	-	1
(2, 3, 6, 7)	0	-	1	-
(2, 6, 3, 7)	0	-	1	-

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

1990-1991

1

1922

1

110

3

1

8

312

⇒ Step 2 :

Essential Implicants :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$wx\bar{y}\bar{z}$	12													
$\bar{x}y\bar{z}$	2, 10													
$\bar{w}\bar{x}$	0, 1, 2, 3													
$\bar{w}\bar{z}$	1, 3, 5, 7													
$\bar{w}y$	2, 3, 6, 7													

$$F = wx\bar{y}\bar{z} + \bar{x}y\bar{z} + \bar{w}\bar{x} + \bar{w}y$$

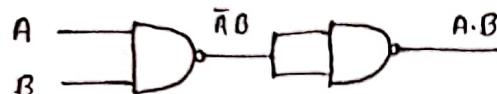
✳ Universal gates : NAND & NOR

Using NAND gate

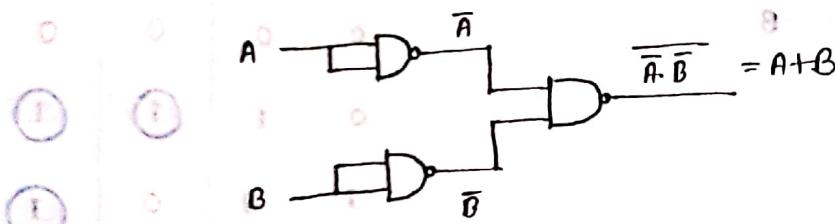
a) Inverterz:



b) AND Gate :

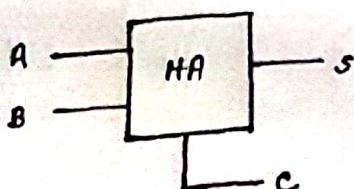


c) GR Gate:



Chapter 4

Half ADDER:



Truth Table :

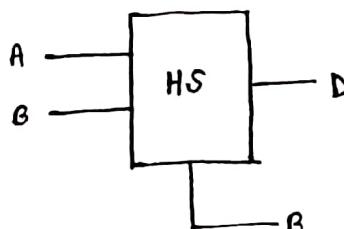
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = A \oplus B$$

$$C = AB$$

$R = \overline{AB} \rightarrow$  $\rightarrow R$

Half Subtractor



$$D = A \oplus B$$

$$B = \overline{AB}$$



Truth Table :

A	B	\overline{AB}	$S = D$	$B = \overline{AB}$
0	0	0	0	0
0	1	1	1	1
1	0	0	0	1
1	1	0	0	0

2 bit Binary ADDER

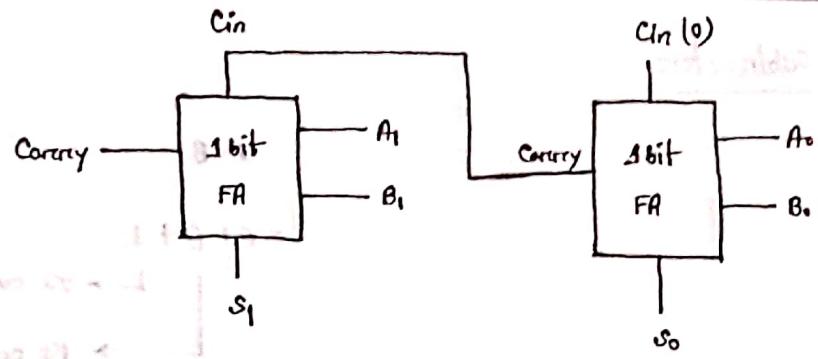
$$A = A_1 \quad A_0$$

$$B = B_1 \quad B_0$$

$$C_1 \quad S_1 \quad S_0$$

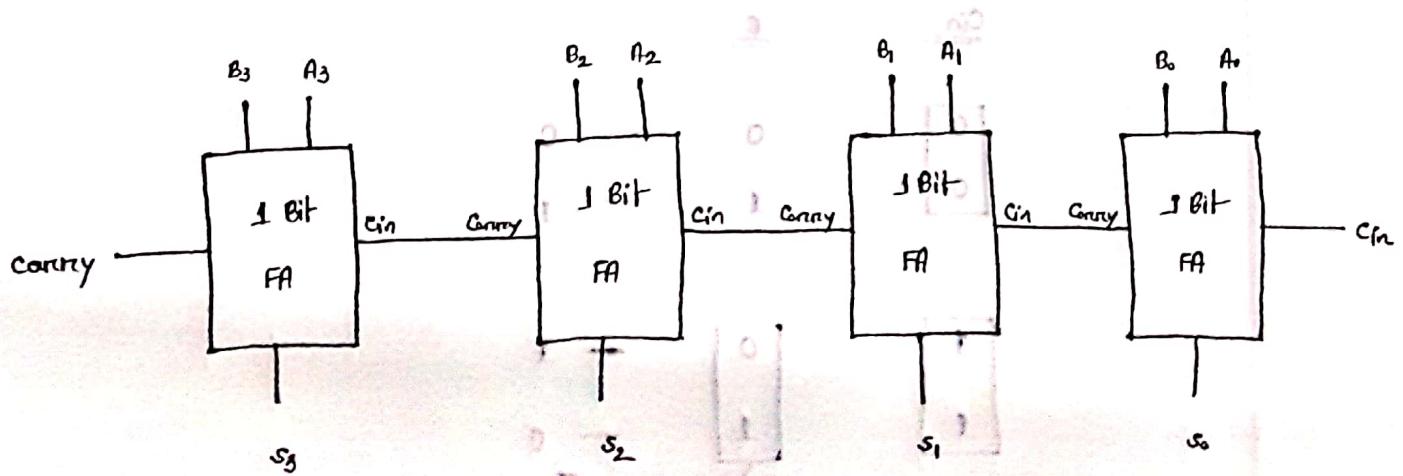


2 bit Binary ADDER



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

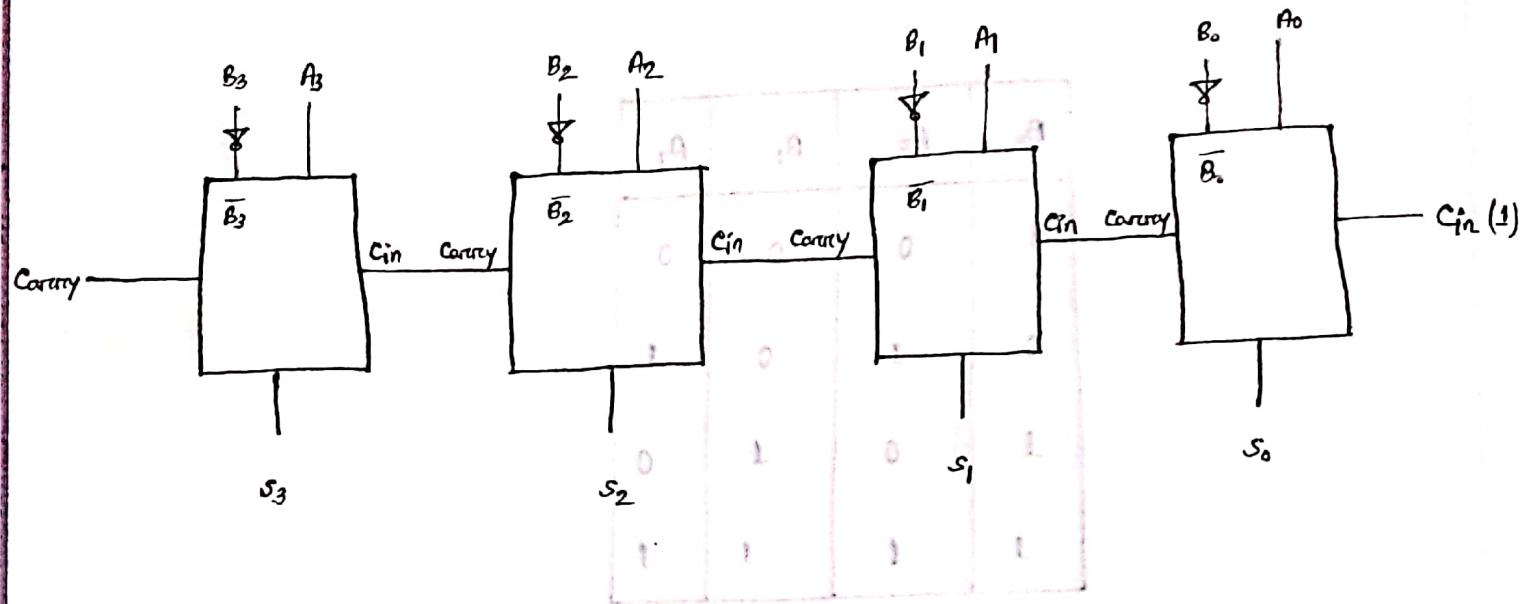
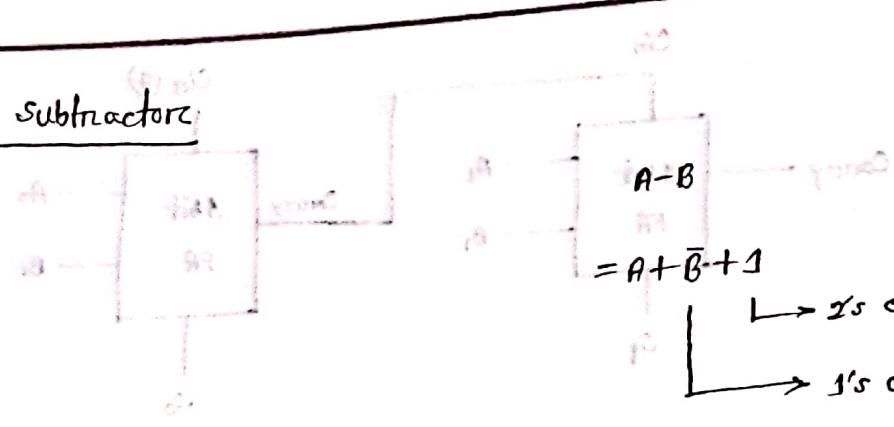
4 bit Binary ADDER :



— Ripple Adder

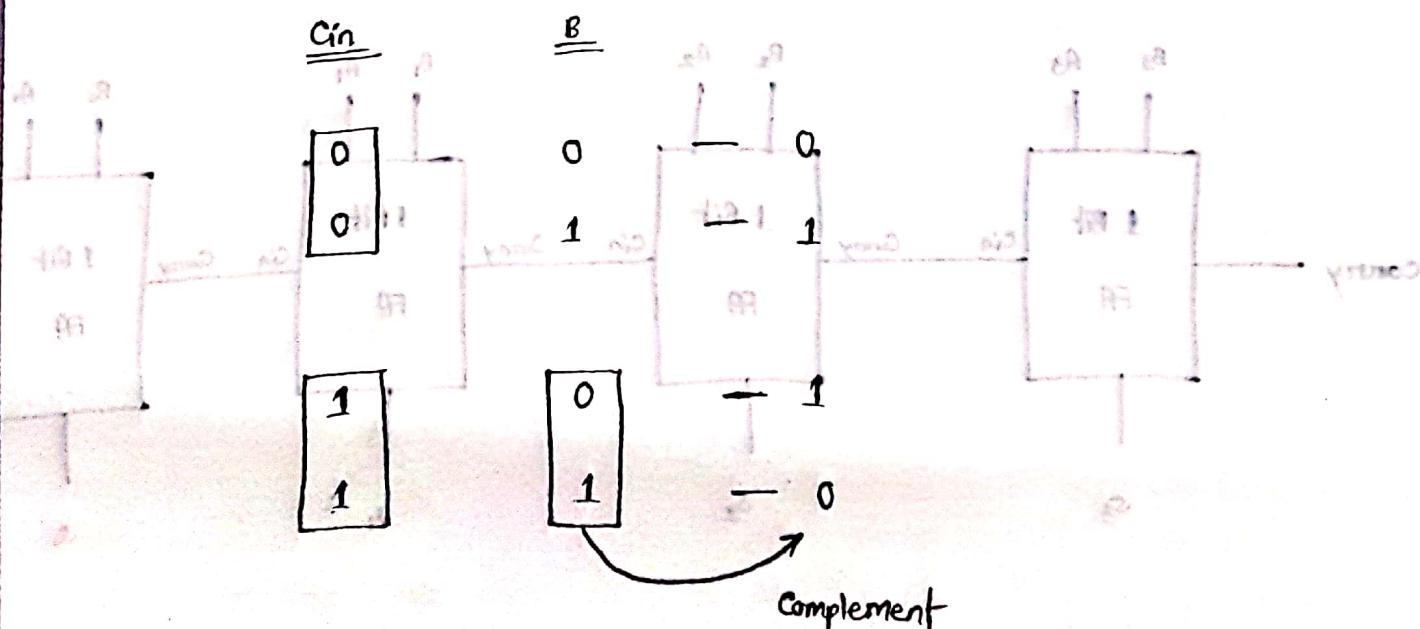
— Slow যাতা যাহা

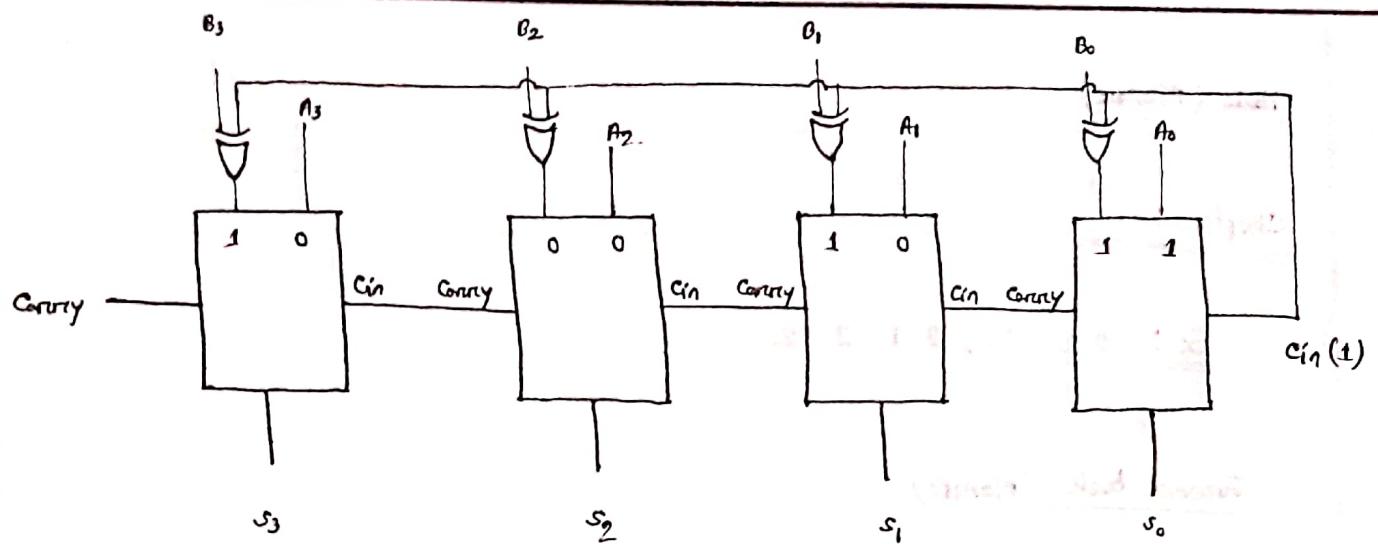
4 bit Binary Subtractor



4 bit Binary Adder, Subtractor

: 230000 यांत्रिक तिक्का





Total \oplus input

$$A = 0 \ 0 \ 0 \ 1$$

~~$$B = 0 \ 1 \ 0 \ 0$$~~

$$\underline{0 \ 1 \ 0 \ 1}$$

$$A = 0 \ 0 \ 0 \ 1$$

~~$$B = 1 \ 0 \ 1 \ 1$$~~

$$+ 1$$

Ans: 1010000000000000

Ans: 1010000000000000

HDL (Manorc)

Chapters : 02

Ex: 2.2, 2.3, 2.4, 2.22

Green book (Manorc)

Ex: 2.5, 2.7, 2.13, 2.14

turn off the plot

Ex: 06

1 0 0 0 → A

1 0 0 0 → A

a) Design a circuit for parity checker, for 3 bit

1 1 1

1 0 1 0

There is a switch S.

if $S=0$ then circuit checks odd parity

if $S=1$ " " " even parity.

Truth Table :

Input				Output	
S	A	B	C	P_1 (odd parity)	P_2 (even parity)
0	0	0	0	0	x
0	0	0	1	1	x
0	0	1	0	1	x
0	0	1	1	0	x
0	1	0	0	1	x
0	1	0	1	0	x
0	1	1	0	0	x
0	1	1	1	1	x

Odd parity checker
($A \oplus C$)

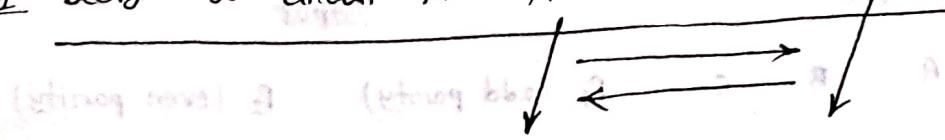
S	A	B	C	P_1 (odd parity)	P_2 (even parity)
1	0	0	0	0	x
1	0	0	1	1	x
1	0	1	0	1	x
1	0	1	1	0	x
1	1	0	0	1	x
1	1	0	1	0	x
1	1	1	0	0	x
1	1	1	1	1	x

Even parity checker
($A \oplus B \oplus C$)

S	A	B	C	$P_1 = \Sigma ()$ $d_1 = (8-15)$	$P_2 = \Sigma ()$ $d_2 = (0-7)$
0	0	0	0	0	0

Code Conversion Final
18/2/2022

bl Design a circuit for xs3 code to 8421 code converter



84-2-1 \longrightarrow 8421

2421 \longrightarrow 84-2-1

Decimal	xs3	Input (xs3)				Output (8421 code)			
		A	B	C	D	w	x	y	z
0	3	0	0	1	1	0	0	0	0
1	4	0	1	0	0	0	0	0	1
2	5	0	1	0	1	0	0	1	1
3	6	0	1	1	0	0	1	1	1
4	7	0	1	1	1	1	1	1	1
5	8	1	0	0	1	0	0	1	1
6	9	1	0	0	1	1	0	1	1
7	10	1	0	1	0	1	1	1	1
8	11	1	0	1	1	1	0	1	1
9	12	1	1	0	0	1	0	0	1

$$w = \Sigma ()$$

$$x = \Sigma ()$$

$$d = \Sigma (0, 1, 2, 13, 14, 15)$$

$$y = \Sigma ()$$

$$z = \Sigma ()$$

Decimal	2421 code	Input (2421)				Output
		A	B	C	D	
0	0	0	0	0	0	
1	1	0	0	0	1	
2	8	1	0	0	0	
3	3	0	0	1	1	
4	10	1	0	1	0	
5	11	1	0	1	1	
6	12	1	1	0	0	
7	13	1	1	0	1	
8	14	1	1	1	0	
9	15	1	1	1	1	

$$d = \{2, 4, 5, 6, 7, 9\}$$

c) Design a 4 bit binary - to gray code converter for $s = 0$

4 bit gray to binary " " " $s = 1$

Conversion

s	A B C D	W X Y Z	
0	0 1 1 1	0 1 1 1	Binary to Gray
0	15	1 1 1 1	
1	0 0 0 0	1 0 0 1	Gray to Binary
1	31	1 0 0 1	