Digital Electronics

book = Morris Mano

① Number system → ⓐ Decimal
ⓑ Binary
ⓒ octal
ⓓ hexadecimal

② BCD (Binary coded decimal) [4 bits system]

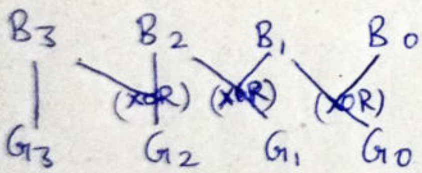
③ Excess 3 code = Decimal (BCD) + 0011

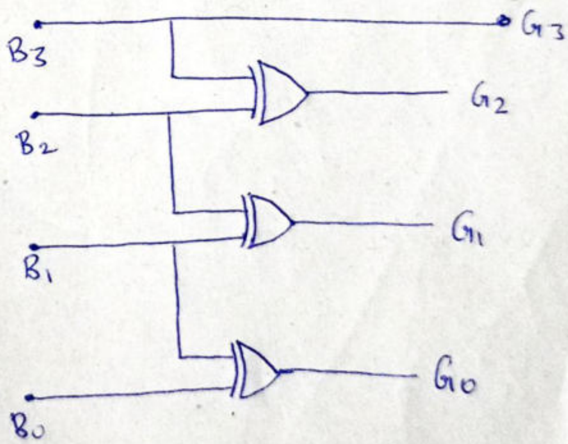
eg) BCD 0000 → excess 3 0011

④ self complimentary code

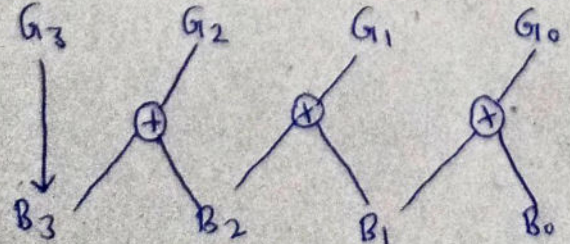
⑤ BCD ↔ Gray Conversions

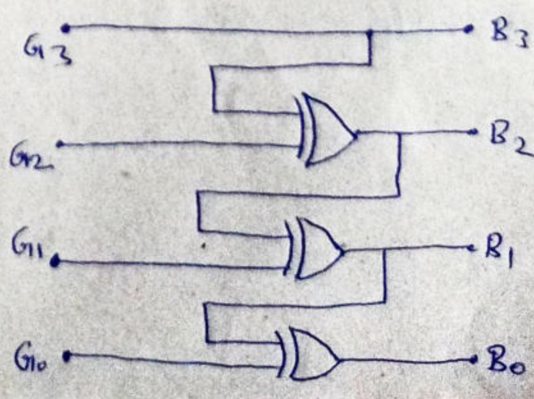
⑥ Binary to Gray code





⑦ Gray Code to Binary Code :-





① boolean algebra
② venn diagram
③ K-map
④ Quin-Mccluskey

Boolean Algebra & Reduction Techniques :

1904 → Huntinton E.V → Postulates (5)

① Associative $A+(B+C)=(A+B)+C$, $A.(B.C)=(A.B).C$

② Distributive $A+(B.C)=(A+B).(A+C)$, $A.(B+C)=(A.B)+(A.C)$

③ commutative $A+B=B+A$, $A.B=B.A$

④ Identity elements $x.1=x$
$x+0=x$

⑤ Inverse law $a+(-a)=0$

1938 - Shannon CE → switching algebra

④ AND operation $A.0=0$, $A.1=A$
$A.A=A$, $A.\bar{A}=0$

⑤ OR operation : $A+A=A$, $A+1=1$
$A+0=A$, $A+\bar{A}=1$

⑥ Involution theorem : $(A')'=\bar{\bar{A}}=A$

⑦ Demorgon's theorem : $\bar{A}+\bar{B}+\bar{C}=\overline{A.B.C}$

⑧ Transposition Theorem : $(A+B)(A+C)=A+BC$

⑨ Consensus Theorem : (used to eliminate redundant term)
$\boxed{\text{3 Variables}}$ + each variable represented 2 times and any one variable should be in complement.

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

(10) Duality Theorem :-

AND $\longleftrightarrow$ OR

$1 \longleftrightarrow 0$

eg) $xyz + y'z = 1$

(11) Compliment Theorem :

$1 \longleftrightarrow 0$

any binary variable is called literal

★ Canonical form :- every ~~literal~~ term has all variables

★ eg) $F(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + AB\bar{C}$

★ standard form : all the terms do not have all literals

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

★ Min term & Max term :-

(Product form) (Sum form)

A.B.C. A+B+C

Max term is Compliment of Min term ($M = \bar{m}$)

★ SOP (Sum of Product) :- each literal represent in min term

eg) $ABC + \bar{A}BC$

★ POS :- Eg) $(A+B+C)\cdot(\bar{A}+\bar{B})$

solve + mention
Theorem +
stepps

~~# Venn diagram~~

Eg) Minimize $F = abc + \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\cdot\bar{c} + \bar{b}\bar{c}$

$\boxed{8C}$

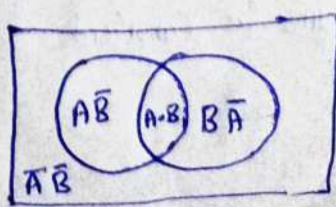
$= abc + \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{b}\bar{c}$ $(a+1=1)$ (distribution)

$=$

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

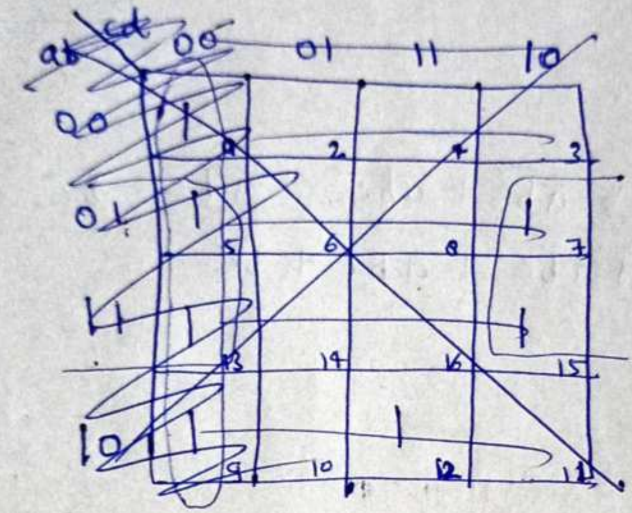
Venn-diagram !

OR = Union
AND = Intersection



K-map :-

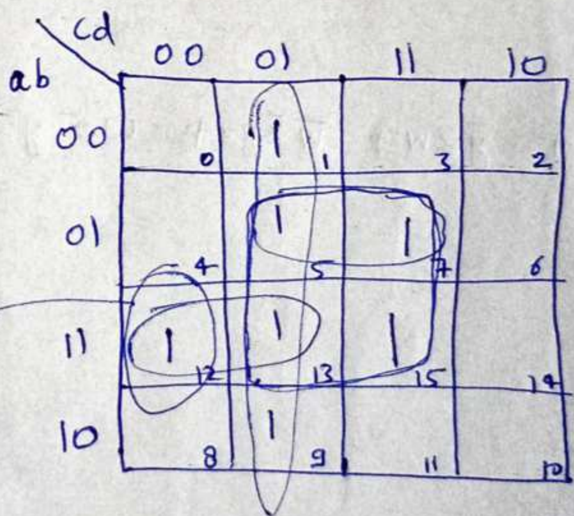
$F(a,b,c,d) = \sum(1,5,7,9,12,13,15)$



~~F(a,b,c,d)~~

~~P(a,b,c,d)~~

essential Prime Implicant



* Implicant = each individual min term in Canonical form

* Prime Implicant ⟶ Obtained by combining maxm possible adjacent cells.

* Essential Prime implicant
  ↳ It is a prime Implicant with one and 2 terms are unique i.e terms cannot combine more than 1 time.

here Implicant = 7
Prime Implicant = 3

Implicant:

$F(a,b,c,d) = \bar{a}\bar{b}\bar{c}d + \bar{a}b\bar{c}d + \bar{a}bcd + ab\bar{c}\bar{d} + ab\bar{c}d + abcd + a\bar{b}\bar{c}d$

Prime Implicant:

$f(a,b,c,d) = \bar{c}d + bd + ab\bar{c}$

essential Prime Implicant: $ab\bar{c}$.

Final result $= f(a,b,c,d) = \bar{c}d + bd + ab\bar{c}$

20/01/23

Minimization Techniques: → Boolean Theorem / Law / Postulates
→ K-map
→ Quin-Mcluksy (Tabulation method)

Limitation of K-map: if no. of variable is higher

Quin-Macluksy Method:-

$$F(A,B,C,D,E,F,G) = \sum m\,(20, 28, 52, 60)$$

Step (i) Group Min-terms

Group	minterm	A	B	C	D	E	F	G	
G_0	20	0	0	1	0	1	0	0	Two 1's
G_1	28	0	0	1	1	1	0	0	Three 1's
	52	0	1	1	0	1	0	0	
G_2	60	0	1	1	1	1	0	0	four 1's

Step (ii) Make Pair of min terms and make 1 bit variation

Group	Paired Minterm	A	B	C	D	E	F	G	
G_0	(20, 28)	0	0	1	_	1	0	0	1-bit variable
	(20, 52)	0	_	1	0	1	0	0	
G_1	(28, 60)	0	_	1	1	1	0	0	1bit variable
	(52, 60)	0	1	1	_	1	0	0	

if any paired min term repeated called redundant

Step (ii) Pair again for 2 bit variation

Group	Paired Min terms	A	B	C	D	E	F	G	
G_0	(20,28,52,60)	0	–	1	–	1	0	0	redundant
	(20,52,28,60)	0	–	1	–	1	0	0	

If redundant come we can neglect any 1 expression.

so minimized term, we get

$$\boxed{F(A,B,C,D,E,F,G) = \overline{A}CE\overline{F}\overline{G}}$$

Digital Logic Circuit

→ Combinational Logic Circuit
(do not have memory)
(output directly depend on I/p)
(eg all ALU operations)

→ Sequential Logic Circuit
(has memory at each step)
(output depend on I/p and previous output)
(eg FF, cnt, Register)